

Ассоциация научно-технических организаций "Уральский профессиональный форум"
Автономная некоммерческая профессиональная образовательная организация
"Современный цифровой колледж при Западно-Уральском институте экономики и права"
(АНПО "СЦК при ЗУИЭП")

УТВЕРЖДАЮ
Директор
/И.И. Лобанова/
«12» августа 2024 г.



**ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ
И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
в форме экзамена
ПМ.02 ПОДГОТОВКА ИНТЕРФЕЙСНОЙ ГРАФИКИ
по профессии
09.01.03 Оператор информационных систем и ресурсов
квалификация «Оператор информационных систем и ресурсов»**

форма обучения: очная

Пермь, 2024

Рекомендовано к утверждению
на заседании Педагогического совета
АНПОО "СЦК при ЗУИЭП",
протокол № 4 от «26» февраля 2024 г.

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ.....	4
3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	6
4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ.....	11
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	70

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (ФОС) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки квалифицированных рабочих, служащих по профессиональному модулю ПМ.02 Подготовка интерфейсной графики.

ФОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

ФОС текущего контроля используется для оперативного и регулярного управления учебной деятельностью студентов.

ФОС промежуточной аттестации студентов по профессиональному модулю предназначен для оценки степени достижения запланированных результатов обучения по завершению изучения междисциплинарных курсов профессионального модуля, экзамена (квалификационного) по завершению изучения профессионального модуля в целом.

ФОС разработан на основании:

- программы подготовки квалифицированных рабочих, служащих по профессии СПО 09.01.03 Оператор информационных систем и ресурсов;
- рабочей программы профессионального модуля ПМ.02 Подготовка интерфейсной графики;
- учебного плана профессии СПО 09.01.03 Оператор информационных систем и ресурсов.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом в рамках освоения профессионального модуля ПМ.02 Подготовка интерфейсной графики является овладение студентами вида профессиональной деятельности «Оформление и компоновка технической документации», в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках

Перечень профессиональных компетенций

Вид профессиональной деятельности	Код	Наименование видов деятельности и профессиональных компетенций
Подготовка интерфейсной графики	ПК 2.1	Создавать визуальный дизайн элементов графического пользовательского интерфейса
	ПК 2.2	Подготавливать графические материалы для включения в графический пользовательский интерфейс
	ВПК 2.3	Применять основы HTML, CSS и JavaScript в профессиональной деятельности

В результате освоения профессионального модуля студент должен:

Иметь практический опыт	разработки графического пользовательского интерфейса в целом или отдельных элементов управления по определенному ранее визуальному стилю; создания раскадровок анимации интерфейсных объектов; рисования пиктограмм, включая разработку их метафор; рисования различных видов интерфейсной графики.
уметь	оптимизировать интерфейсную графику под различные разрешения экрана; Создавать графические документы в программах подготовки растровых изображений; создавать графические документы в программах подготовки векторных изображений; рисовать анимационные последовательности и раскадровку; подбирать графические метафоры, максимально точно соответствующие назначению разрабатываемого элемента управления; работать в границах заданного стиля. читать код веб-страницы встраивать графические изображения в код веб-страницы создавать анимированные кнопки и формы на веб-страницах с использованием графических элементов
знать	правила перспективы, колористики, композиции, светотени и изображения объема; требования целевых операционных систем и платформ к пиктограммам и элементам управления; общие принципы анимации; правила типографского набора текста и верстки. основы HTML, CSS и JavaScript

3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Оценивание уровней сформированности профессиональных и общих компетенций проводится в рамках текущего и промежуточного контроля.

В результате освоения профессионального модуля ПМ.02 Подготовка интерфейсной графики студенты демонстрируют три уровня сформированности профессиональных компетенций: пороговый, базовый и повышенный.

Для каждого конкретного этапа формирования компетенции определены категории «знать», «уметь», «практический опыт», в которые вкладывается следующий смысл:

«приобрести практический опыт» – решать усложненные задачи на основе приобретенных умений и навыков, с их применением в профессиональных деятельности;

«уметь» – решать типичные задачи на основе воспроизведения стандартных алгоритмов решения;

«знать» - воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты.

Промежуточная аттестация:

МДК.02.01 Подготовка интерфейсной графики – экзамен

МДК.02.02 Основы HTML, CSS и JavaScript – дифференцированный зачет

Учебная практика – дифференцированный зачет

Производственная практика – дифференцированный зачет

ПМ.02. Подготовка интерфейсной графики - экзамен по модулю.

Текущий контроль качества обученности студентов осуществляется в следующих формах:

1. проведение экспресс-опросов;
2. фронтальные устные опросы;
3. тестирование по отдельным темам или блокам тем;
4. проверка правильности решения задач по образцу и ситуационных задач;
5. оценка результатов работы на практических занятиях;
6. Защита отчетов по практическим работам
7. Защита отчетов по практическим работам.
8. Экспертное наблюдение за выполнением различных видов работ во время практических занятий, учебной/ производственной практики

КРИТЕРИИ ОЦЕНИВАНИЯ ЗНАНИЙ И УМЕНИЙ ОБУЧАЮЩИХСЯ, СФОРМИРОВАННОСТИ ОБЩИХ И ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ

Критерии оценки устного ответа обучающихся в 5-балльной системе

При оценке устного ответа обучающегося учитывается:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;
- 3) языковое оформление ответа.

Отметка «5»: ответ исчерпывающий, точный, полный и правильный на основании изученного материала; материал изложен в определенной логической последовательности, литературным языком; ответ самостоятельный.

Отметка «4»: ответ полный, обнаруживающий хорошее знание и понимание изученного материала; материал изложен в определенной логической последовательности, последовательно и грамотно, возможны отдельные затруднения в формулировке выводов.

Отметка «3»: ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный ответ, изложенный нелогично, ставится за ответ, в котором в

основном правильно, но схематично или с отклонениями от последовательности изложения раскрыт материал.

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

ОБЩАЯ КЛАССИФИКАЦИЯ ОШИБОК

При оценке знаний, умений, навыков учитываются все ошибки (грубые и негрубые), а также недочёты в работе.

Грубыми считаются ошибки: - незнание определения основных понятий, законов, общепринятых символов обозначений величин; - неумение выделить в ответе главное; обобщить результаты изучения; - неумение применить знания для решения задач, объяснения явления; - неумение подготовить установку или лабораторное оборудование, провести опыт, наблюдение, использовать полученные данные для выводов; - неумение пользоваться первоисточниками, учебником, справочником; - нарушение техники безопасности, небрежное отношение к оборудованию, приборам, материалам.

Негрубыми считаются ошибки: - неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными; - ошибки при снятии показаний с измерительных приборов, не связанные с определением цены деления шкалы; - ошибки, вызванные несоблюдением условий проведения опыта, наблюдения, условий работы прибора, оборудования; - нерациональный метод решения задачи, выполнения части практической работы, недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными); - нерациональные методы работы со справочной литературой; - неумение решать задачи, выполнять задания в общем виде.

Недочётами являются: - нерациональные приёмы выполнения опытов, наблюдений, практических заданий; - арифметические ошибки в вычислениях; - орфографические и пунктуационные ошибки.

КРИТЕРИИ ОЦЕНИВАНИЯ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКОГО ЗАДАНИЯ ОБУЧАЮЩИХСЯ В 5- БАЛЛЬНОЙ СИСТЕМЕ

Отметка «5»: работа выполнена полностью и правильно; сделаны правильные выводы.

Отметка «4»: работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.

Отметка «3»: работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.

Отметка «2»: допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

ОЦЕНКА ТЕСТОВ

Оценка в баллах Степень выполнения задания

Неуд. Выполнено не менее 40 % предложенных заданий

Удов. Выполнено не менее 41-70 % предложенных заданий

Хор. Выполнено не менее 71-95% предложенных заданий

Отл. Выполнено не менее 96-100% предложенных заданий

КРИТЕРИИ ОЦЕНКИ НАПИСАНИЯ СООБЩЕНИЙ, ДОКЛАДОВ

- **оценка «отлично»** выставляется обучающемуся, если выполнены все требования к написанию сообщения (доклада): обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы;

- **оценка «хорошо»** выставляется обучающемуся, если основные требования к сообщению, докладу и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём доклада (сообщения); имеются упущения в оформлении; на дополнительные вопросы даны неполные ответы.

- **оценка «удовлетворительно»** выставляется обучающемуся, если имеются существенные отступления от требований к написанию сообщения (доклада). В частности, тема освещена лишь частично; допущены фактические ошибки в содержании сообщения (доклада) или при ответе на дополнительные вопросы; во время защиты отсутствует вывод;

- **оценка «неудовлетворительно»** выставляется обучающемуся, если тема сообщения (доклада) не раскрыта, обнаруживается существенное непонимание проблемы.

КРИТЕРИИ ОЦЕНИВАНИЯ ПРЕЗЕНТАЦИЙ

Оценка «отлично»: Содержание: Работа полностью завершена, обучающийся демонстрирует глубокое понимание описываемых процессов, даны интересные дискуссионные материалы, грамотно используется лексика, предлагается собственная интерпретация или развитие темы. Дизайн логичен. Все параметры шрифта хорошо подобраны. Текст хорошо читается. Графика подобрана грамотно, соответствует содержанию. Нет орфографических и синтаксических ошибок.

Оценка «хорошо»: Полностью сделаны наиболее важные компоненты работы, обучающийся демонстрирует понимание основных моментов, хотя некоторые детали не уточняются. Некоторые материалы носят дискуссионный характер. Научная лексика используется, но иногда не корректно. Обучающийся в большинстве случаев предлагает собственную интерпретацию или развитие темы. Дизайн презентации выдержан и соответствует содержанию. Параметры шрифта подобраны. Графика соответствует содержанию. Минимальное количество ошибок.

Оценка «удовлетворительно»: В содержании не выделены все важные компоненты. Обучающийся демонстрирует неполное понимание темы. Дискуссионные материалы есть в наличии, но не способствуют раскрытию проблемы. Научная терминология используется не всегда корректно. Дизайн не соответствует полному раскрытию содержания. Параметры шрифта недостаточно хорошо подобраны и могут мешать восприятию. Графика не в полной мере соответствует содержанию. Имеются орфографические и пунктуационные ошибки, мешающие восприятию.

Оценка «неудовлетворительно»: Работа выполнена фрагментарно и с посторонней помощью, обучающийся демонстрирует минимальное понимание темы. Минимум дискуссионных материалов и научных терминов. Интерпретация ограничена или беспочвенна. Дизайн неясен. Элементы дизайна мешают содержанию. Текст трудночитаемый. Графика не соответствует содержанию. Много орфографических и пунктуационных ошибок, делающих материал трудночитаемым

КРИТЕРИИ ОЦЕНИВАНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений студентов;
- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов:
 - творческой инициативы, самостоятельности, ответственности, организованности;
 - формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации;
 - формирования практических умений;
 - развитию исследовательских умений.

Разновидности самостоятельной работы студентов:

1. Чтение основной и дополнительной литературы. Самостоятельное изучение материала по литературным источникам.
2. Работа с библиотечным каталогом, самостоятельный подбор необходимой литературы.
3. Поиск необходимой информации через Интернет.
4. Конспектирование источников.
5. Реферирование источников.
6. Составление рецензий и отзывов на прочитанный материал.
7. Составление обзора публикаций по теме.
8. Ведение дневника (дневник производственной практики)
9. Прослушивание учебных аудиозаписей, просмотр видеоматериала.
10. Подготовка к различным формам промежуточной и итоговой аттестации (к тестированию, диктанту, контрольной работе, зачету, экзамену).
11. Самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тренировочные упражнения, задачи, тесты).
12. Выполнение творческих заданий.
13. Проведение практического задания и составление отчета по нему.
14. Подготовка устного сообщения для выступления на лекционном занятии.
15. Написание реферата. Подготовка к защите (представлению) реферата на лекционном занятии.
16. Подготовка доклада и написание тезисов доклада.
17. Выполнение комплексного задания (проекта) по отдельной дисциплине. Подготовка к его защите на практическом занятии.
18. Подготовка к участию в деловой игре, конкурсе, творческом соревновании.

Оценка «5» ставится когда:

- Студент свободно применяет знания на практике;
- Не допускает ошибок в воспроизведении изученного материала;
- Студент выделяет главные положения в изученном материале и не затрудняется в ответах на видоизмененные вопросы;
- Студент усваивает весь объем программного материала;
- Материал оформлен аккуратно в соответствии с требованиями;

Оценка «4» ставится когда:

- Студент знает весь изученный материал;
- Отвечает без особых затруднений на вопросы преподавателя;
- Студент умеет применять полученные знания на практике;
- В условных ответах не допускает серьезных ошибок, легко устраняет определенные неточности с помощью дополнительных вопросов преподавателя;
- Материал оформлен недостаточно аккуратно и в соответствии с требованиями;

Оценка «3» ставится когда:

- Студент обнаруживает освоение основного материала, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных дополняющих вопросов преподавателя;
- Предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднения при ответах на воспроизводящие вопросы;
- Материал оформлен не аккуратно или не в соответствии с требованиями;

Оценка «2» ставится когда:

- У студента имеются отдельные представления об изучаемом материале, но все, же большая часть не усвоена;
- Материал оформлен не в соответствии с требованиями

4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

4.1. Типовые задания для оценки освоения МДК.02.01 Подготовка интерфейсной графики

ТЕСТ «ФОРМАТИРОВАНИЕ ТЕКСТА»

1. Абзац — это часть документа между двумя ... управляющими символами ... абзаца:
 - а) соседними непечатаемыми, конца +
 - б) соседними непечатаемыми, начала
 - в) соседними напечатанными, начала
2. Высота шрифта, измеряемая от нижнего края самой низкой буквы до верхнего края самой высокой буквы:
 - а) размер кегли
 - б) размер кегля +
 - в) длина кегля
3. Расширение файла, предназначенного для представления в электронном виде полиграфической продукции:
 - а) avi
 - б) jpg
 - в) pdf +
4. К свойствам абзаца относится:
 - а) отступ первой строки +
 - б) размер кегля
 - в) стиль начертания символов
5. К свойствам абзаца не относится:
 - а) отступ первой строки
 - б) оба варианта верны
 - в) цвет символа +
6. Стилизовое форматирование документа применяют при:
 - а) замене символа
 - б) выборе стиля оформления документа +
 - в) работе с абзацем документа
7. Один из основных параметров страницы документа:
 - а) материал бумаги
 - б) шрифт бумаги
 - в) размер бумаги +
8. Один из основных параметров страницы документа:
 - а) ориентация страницы +
 - б) шрифт бумаги
 - в) ориентация полей
9. Один из основных параметров страницы документа:
 - а) шрифт бумаги
 - б) ориентация полей
 - в) размер полей +
10. Форматирование символов — это изменения ... введенных символов:
 - а) данных
 - б) значений свойств +
 - в) размера
11. Расширение файла текстового документа:
 - а) doc +

- б) xls
 - в) exe
12. Не является расширением файла текстового документа:
- а) doc
 - б) mp3 +
 - в) нет верного ответа
13. К свойствам абзаца можно отнести:
- а) выравнивание +
 - б) размер кегля
 - в) цвет символов
14. К свойствам абзаца не относится:
- а) выравнивание
 - б) нет верного ответа
 - в) высота кегля +
15. Прямое форматирование применяют при:
- а) работе с фрагментом текста +
 - б) выборе стиля оформления документа
 - в) установке отступов первой строки во всем документе
16. Прямое форматирование не применяют при:
- а) работе с фрагментом текста
 - б) установке междустрочных интервалов во всем документе +
 - в) нет верного ответа
17. Любой текст, созданный с помощью текстового редактора (процессора), вместе с включенными в него нетекстовыми материалами (например, графикой):
- а) фрагмент
 - б) блокнот
 - в) документ +
18. Этап подготовки документа на компьютере, в ходе которого исправляются обнаруженные ошибки и вносятся необходимые изменения:
- а) подготовка
 - б) редактирование +
 - в) фрагмент
19. Мощная программа для работы с текстами, позволяющая изменять начертание и размер шрифта, включать в документ таблицы, рисунки, схемы:
- а) текстовый редактор
 - б) документ
 - в) текстовый процессор +
20. Произвольная последовательность символов – отдельное слово, строка, абзац, страница и даже весь вводимый текст:
- а) редактирование
 - б) фрагмент +
 - в) документ
21. Программа, предназначенная для создания простых сообщений и текстов:
- а) текстовый редактор +
 - б) текстовый процессор
 - в) фрагмент
22. Какой из форматов текстовых файлов имеет довольно большой информационный объем, зато может быть прочитан многими приложениями:
- а) DOC
 - б) RTF +
 - в) ODT
23. Что размещается на полях документа, кроме номеров страниц:

- а) заметки
 - б) имя документа
 - в) колонтитулы +
24. Какое расстояние задают интервалы перед и после:
- а) между строками
 - б) между абзацами +
 - в) между словами
25. Как ведет себя последняя строка абзаца при выравнивании по ширине:
- а) равномерно растягивается по ширине страницы
 - б) выравнивается по правому краю
 - в) выравнивается по левому краю +
26. Для чего в шрифтах используются засечки:
- а) для лучшего распознавания символов при сканировании
 - б) для облегчения процесса чтения +
 - в) оба варианта не верны
27. Как называются выполненные в едином стиле изображения символов, используемых для письма:
- а) шрифт +
 - б) алфавит
 - в) код
28. Как называется минимальная графическая единица текста:
- а) буква
 - б) точка
 - в) символ +
29. Какие есть два способа форматирования текста:
- а) прямое и стилевое форматирование +
 - б) прямое и косвенное форматирование
 - в) общее и частное форматирование
30. Как называется процесс оформления текста:
- а) декорирование
 - б) форматирование +
 - в) редактирование

ТЕСТ «ОСНОВЫ ВИЗУАЛЬНОЙ ГРАФИКИ И ТИПОГРАФИКИ»

Вариант 1

№	Вопрос	Эталон ответа	Р
1	Выберите правильный вариант ответа: Символ, употребляемый для обозначения элемента маркированного списка называется маркер или ... а) буллит б) тире в) номер	а	1
2	Ответьте, о каком принципе дизайна идет речь? На странице ничто не должно размещаться случайно. Каждый элемент нужно зрительно связывать с другими элементами. Это придает странице аккуратный, утонченный и чистый внешний вид.	Выравнивание	1
3	Установите соответствие между типом шрифта и его изображением	1-б, 2-в, 3-а	3

	<p>1) Современный стиль а) Добрый день!</p> <p>2) Рубленый шрифт б) Добрый день!</p> <p>3) Рукописный шрифт в) Добрый день!</p>		
4	<p>Установите соответствие: Имеют ли строчные буквы в приведенных ниже примерах:</p> <p>А) тонкие горизонтальные засечки Б) засечки с наклоном В) отсутствие засечек Г) толстые горизонтальные засечки</p> <p>1) Спорт! 2) Спорт! 3) Спорт! 4) Спорт!</p>	1-в, 2-б, 3-а, 4-г	4
5	<p>Выберите образцы, на которых шрифты создают Контраст:</p> <p>1) Гадкий Утенок 2) Спорт <i>обозреватель</i> 3) <i>сказка «Красная шапочка»</i> 4)</p>	1, 2, 4	3
6	<p>Укажите можно ли использовать два рукописных шрифта на одной странице?</p>	нет	1
7	<p>Ответьте, верно ли утверждение? Если в одном шрифте буквы отличаются контрастными переходами от толстого штриха к тонкому, то другой должен быть рубленым или брусковым.</p>	верно	1
8	<p>Выберите несколько правильных ответов: Расстановка элементов текста на странице может определять взаимоотношения между ними как:</p> <p>1) выровненные 2) гармонические, 3) конфликтные 4) контрастные. 5) повторяющиеся</p>	2,3,4	3
9	<p>Установите соответствие между изображением и видом цветовой схемы</p> <p>1) а) комплементарная схема</p>	1-а, 2-в, 3-б	3

	2) б) монохромная схема 3) в) прямоугольная схема		
10	Выберите, что из перечисленного ниже относится к шрифтовому контрасту? 1. Форма 2. Насыщенность 3. Цвет 4. Расположение 5. Подчеркивание	1, 2, 3	3
11	Назовите основные цвета современного спектрального круга Освальда, который характеризуется плавностью переходов цветов	Красный, Синий, Зеленый	3
	Итого:		26

Вариант 2

№	Вопрос	Эталон ответа	Р
1	Выберите правильный вариант ответа: белое (пустое) пространство, со всех сторон окруженное элементами страницы (текстом или фотографиями) называется а) Основным б) Замкнутым в) Открытым	б	1
2	Ответьте, о каком принципе дизайна идет речь? Используйте этот принцип, чтобы избежать расположения на странице одинаковых элементов. Если эти элементы: шрифт, цвет, размер, толщина штриха, контур, пробелы и т. д. - не означают одно и то же, их следует оформлять по-разному.	Контраст	1
3	Установите соответствие между типом шрифта и его изображением 1) Современный стиль а) Goodbye! 2) Декоративный шрифт б) Goodbye! 3) Старый стиль в) Goodbye!	1-в, 2-б, 3-а	3
4	Установите соответствие: Имеют ли строчные уквы в приведенных ниже примерах: А) тонкие горизонтальные засечки Б) засечки с наклоном В) отсутствие засечек Г) толстые горизонтальные засечки	1-в, 2-а, 3-б, 4-г	4

	<p>1) Music! 2) Music! 3) Music! 4) Music!</p>		
5	<p>Выберите образцы, на которых шрифты создают Конфликт:</p> <p>1) СКАЗКА «Красная шапочка»</p> <p>2) Гадкий Утенок</p> <p>3)</p> <p>4) Алиса в стране чудес</p>	1, 4	2
6	<p>Укажите верно ли, что если один шрифт – высокий и тонкий, то другой должен быть низким и насыщенным.</p>	да	1
7	<p>Ответьте, верно ли утверждение? Шрифты надо скомбинировать так, чтобы их сочетание было очень привлекательным, пусть даже абсолютно нечитаемым.</p>	неверно	1
8	<p>Выберите несколько правильных ответов: К основным принципам дизайна относятся:</p> <p>1) Повтор 2) Контраст 3) Выравнивание 4) Структура 5) Размер 6) Приближение</p>	1, 2, 3, 6	4
9	<p>Установите соответствие между видом цветовой схемы и изображением</p> <p>1) а) аналоговая схема</p> <p>2) б) монохромная схема</p> <p>3) в) триада</p>	1-в, 2-а, 3-б	3
10	<p>Вставьте пропущенные слова: Классическим кругом из 12 цветов считается круг Иттена. Основу его составляют цвета: красный, ... и ..., второстепенные – ..., фиолетовый и ..., а остальные цвета образуются путем смешивания.</p>	Желтый Синий Оранжевый Зеленый	4
11	<p>Укажите, верно ли утверждение, что нужно подчеркивать значимость одного типографского элемента при помощи жирного шрифта, а другого – при помощи крупного кегля.</p>	неверно	1
	Итого:		25

ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ПО ТЕМЕ «ОСНОВЫ КОМПЬЮТЕРНОЙ ГРАФИКИ»

1. Направления компьютерной графики.
2. Приложения компьютерной графики.
3. Изобразительная компьютерная графика.
4. Обработка и анализ изображений.
5. Анализ сцен.
6. Виртуальная реальность.

ТЕСТ С ОТВЕТАМИ: «КОМПЬЮТЕРНАЯ ГРАФИКА»

1. Одной из основных функций графического редактора является:
 - а) Создание рисунка;
 - б) Ввод рисунка и текста;
 - в) Создание рисунка и манипулирование им+
2. Инструментами в графическом редакторе являются:
 - а) Кривая, скругленный прямоугольник, овал
 - б) Прямая, ластик, многоугольник
 - в) Распылитель, масштаб, выбор цвета+
3. Необходимо установить соответствие:
Текстовый редактор:
 - а) редактирование рисунков
 - б) сочинение стихотворения +
 - в) рисование
4. Необходимо установить соответствие:
Текстовый редактор:
 - а) редактирование изложения +
 - б) редактирование рисунков
 - в) рисование
5. Чтобы изменить размер изображения, можно:
 - а) использовать инструмент Выделение
 - б) вставить фрагмент из буфера обмена
 - в) использовать инструмент Лупа +
6. Чтобы изменить размер изображения, можно:
 - а) вставить фрагмент из буфера обмена
 - б) выделить фрагмент и растянуть рамку выделения +
 - в) использовать инструмент Выделение
7. Закрасить фрагмент средствами графического редактора можно с помощью этого инструмента:
 - а) Ластик
 - б) Выбор цветов
 - в) Заливка +
8. Что нужно использовать, чтобы отобразить симметрично выделенный фрагмент средствами графического редактора:
 - а) инструмент Повернуть +
 - б) один из пунктов меню Файл
 - в) клавишу Delete
9. Как называется устройство, которое чаще всего используют для создания графических файлов:
 - а) видеокамера
 - б) клавиатура

- в) графический планшет +
10. Укажите лишнее в перечне:
- а) линия
 - б) ластик +
 - в) кривая
11. После выделения части текста и выбора команды ВЫРЕЗАТЬ, в графическом редакторе этот фрагмент:
- а) исчезнет с экрана и будет помещен в буфер обмена +
 - б) останется на экране
 - в) останется на экране и не будет помещен в буфер обмена
12. Пиксель является:
- а) основой векторной графики
 - б) основой растровой графики +
 - в) основой трёхмерной графики
13. Графическим объектом НЕ является:
- а) текст письма +
 - б) чертёж
 - в) рисунок
14. Программа для создания и редактирования рисунков:
- а) графический директор
 - б) графический режиссер
 - в) графический редактор +
15. Для ввода изображения в компьютер используются:
- а) сканер +
 - б) монитор
 - в) принтер
16. К какому типу компьютерной графики относится программа Paint:
- а) трёхмерная
 - б) растровая +
 - в) фрактальная
17. Графический редактор Paint находится в группе программ:
- а) утилиты
 - б) Microsoft Office
 - в) стандартные +
18. Чем больше разрешение, тем изображение:
- а) темнее
 - б) качественнее +
 - в) светлее
19. Наименьшим элементом изображения на графическом экране монитора является:
- а) символ
 - б) линия
 - в) пиксель +
20. Необходимо установить соответствие:
- Текстовый редактор:
- а) создание иллюстраций
 - б) набор текста +
 - в) редактирование фотографий
21. Необходимо установить соответствие:
- Текстовый редактор:
- а) редактирование сочинения +
 - б) создание иллюстраций
 - в) редактирование фотографий

22. Необходимо установить соответствие:
Графический редактор:
а) набор текста
б) редактирование фотографий +
в) редактирование сочинения
23. Необходимо установить соответствие:
Графический редактор:
а) набор текста
б) редактирование сочинения
в) создание иллюстраций +
24. Чтобы изменить размер изображения, можно:
а) вставить фрагмент из буфера обмена
б) воспользоваться пунктом меню Изменить размер +
в) использовать инструмент Контур
25. Удалить часть изображения средствами графического редактора можно с помощью:
а) инструмента Кисть
б) одного из пунктов меню Файл
в) выделения и клавиши Delete +
26. Для чего нецелесообразно использовать графический редактор:
а) для обработки сканированных изображений
б) для создания текстового документа +
в) для редактирования фотографий
27. Как называется устройство, которое чаще всего используют для создания графических файлов:
а) цифровой фотоаппарат +
б) видеокамера
в) клавиатура
28. Укажите лишнее в перечне:
а) карандаш
б) кисть
в) линия +
29. После выделения части текста и выбора команды КОПИРОВАТЬ в графическом редакторе этот фрагмент:
а) останется на экране и будет помещен в буфер обмена +
б) останется на экране
в) будет помещен в буфер обмена
30. Выберите устройства являющиеся устройством вывода:
а) сканер
б) принтер +
в) клавиатура
31. Необходимо установить соответствие:
Графический редактор:
а) редактирование рисунков +
б) сочинение стихотворения
в) редактирование изложения
32. Необходимо установить соответствие:
Графический редактор:
а) редактирование изложения
б) сочинение стихотворения
в) рисование

**ВОПРОСЫ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ
«СОЗДАНИЕ ИЗОБРАЖЕНИЯ В СРЕДЕ ГРАФИЧЕСКОГО РЕДАКТОРА
РАСТРОВОГО ТИПА»**

1. Какие виды графики вы знаете?
2. В чем преимущества растровой графики?
3. В чем недостатки растровой графики?
4. Какие графические редакторы растрового типа вы знаете? Для чего они предназначены?
5. Как облегчить себе работу, если изображение включает одинаковые объекты?

ТЕСТ "ОБРАБОТКА РАСТРОВЫХ ИЗОБРАЖЕНИЙ"

Вариант 1




1. Что понимают под растровым изображением?
 - Совокупность пикселей (точек) различного цвета;
 - Графическую сетку;
 - Программу, предназначенную для создания и редактирования изображений.
2. Как называют графическую сетку?
 - Графическим редактором;
 - Растром;
 - Вектором.
3. Что понимают под графическим редактором?
 - Совокупность пикселей (точек) различного цвета;
 - Графическую сетку;
 - Специальную программу, предназначенную для создания и редактирования изображений.
4. Выберите из предложенных программ графический редактор, позволяющий создавать и редактировать растровые изображения:
 - CorelDraw;
 - WindowsMedia;
 - Paint.
5. Какие возможности реализованы в графическом редакторе Paint?
 - Создавать изображения;
 - Редактировать изображения;
 - Создавать и редактировать изображения.
6. Установите взаимно-однозначное соответствие между форматом растровой графики и его описанием.

1 – BMP	А – используется для хранения многоцветных изображений с плавным переходом между цветами.
2 – PNG	Б – Основной формат, поддерживаемый редактором Paint.
3 – JPEG	В – Обеспечивает высокое качество изображений, создавался специально для Интернета.

7. Что понимают под отражением?
 - Зеркальное отображение изображения относительно невидимой оси;
 - Выделенную часть изображения;
 - Изменение изображения.
8. Что понимают под фрагментом изображения?
 - Зеркальное отображение изображения относительно невидимой оси;
 - Выделенную часть изображения;

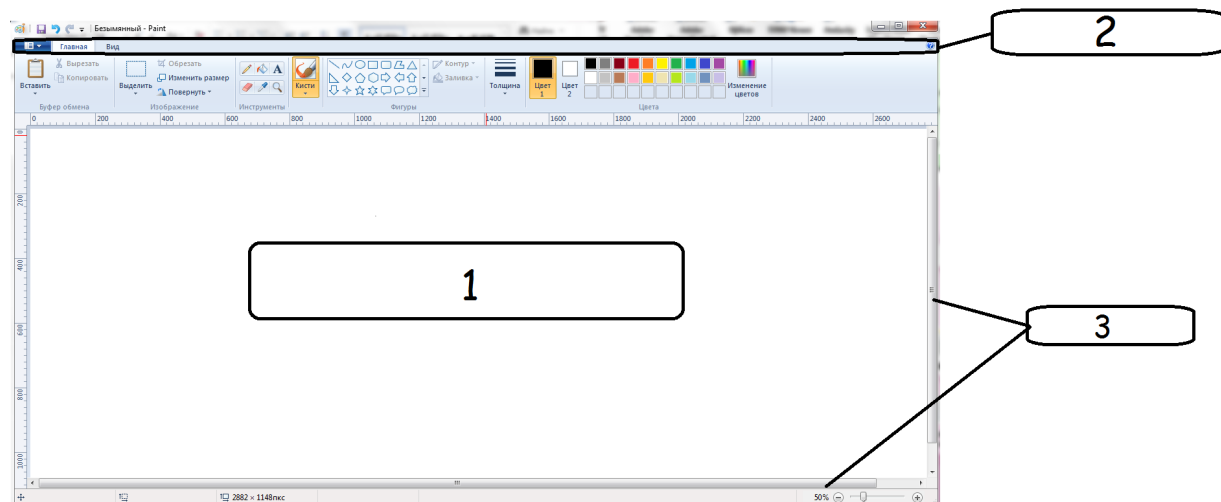
- Изменение изображения.

9. Установите взаимно-однозначное соответствие между инструментом и его названием.

1 – 	А – Кривая.
2 – 	Б – Заливка цветом.
3 – 	В – Многоугольник.

10. Как называются элементы интерфейса графического редактора Paint?

1 – ..., 2 – ..., 3 – ...



ТЕСТ «ОБРАБОТКА РАСТРОВЫХ ИЗОБРАЖЕНИЙ»




Вариант 2

1. Что называют совокупностью пикселей (точек) различного цвета?
 - Растр;
 - Графический редактор;
 - Растровое изображение.
2. Что понимают под растром?
 - Специальную программу;
 - Графическую сетку;
 - Векторную сетку.
3. Что называют специальной программой, предназначенной для создания и редактирования изображений?
 - Растр;
 - Графический редактор;
 - Растровое изображение.
4. Какие изображения позволяет создавать и редактировать графический редактор Paint?
 - Векторные изображения;
 - Растровые изображения;
 - Векторные и растровые изображения.
5. Что относится к редактированию изображений в графическом редакторе Paint?
 - Рисовать детали, поворачивать, масштабировать, наклонять изображение или его части;
 - Вырезать, склеивать и стирать произвольные части изображения, добавлять к изображению текст;
 - Рисовать детали, поворачивать, масштабировать, наклонять изображение или его части, вырезать, склеивать и стирать произвольные части изображения, добавлять к изображению текст.
6. Установите взаимно-однозначное соответствие между форматом растровой графики и его описанием.

1 – GIF	А – используется для хранения многоцветных изображений с плавным переходом между цветами.
2 – TIFF	Б – хранит изображения, которые содержат 256 цветов.
3 – JPEG	В – используется для хранения изображений с большим количеством цветов.

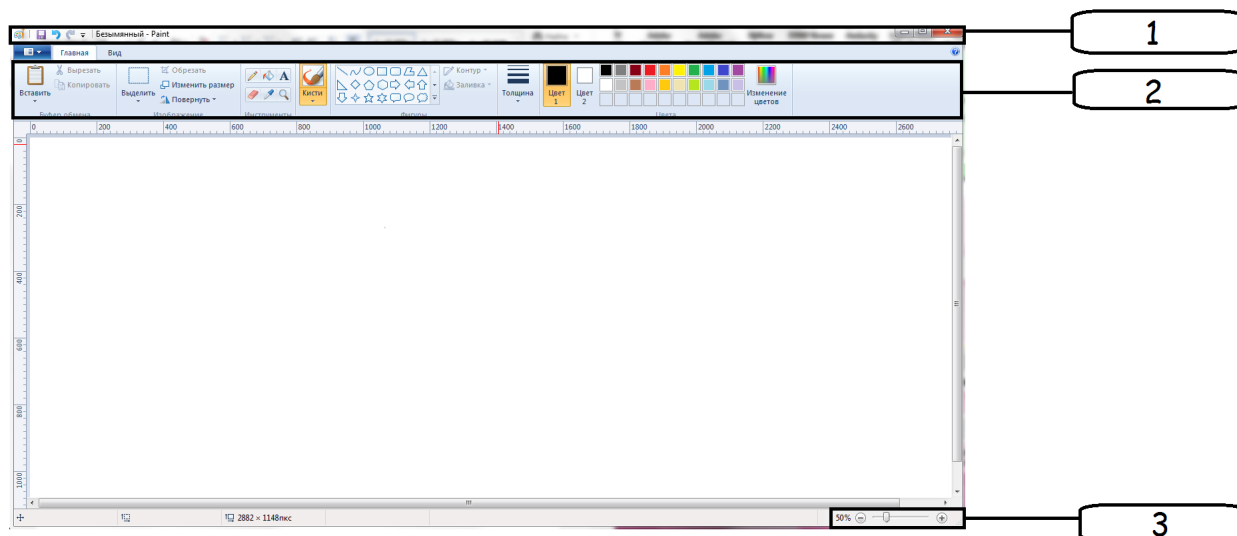
7. Что понимают под редактированием?
 - Зеркальное отображение изображения относительно невидимой оси;
 - Выделенную часть изображения;
 - Изменение изображения.
8. Что понимают под буфером обмена?
 - Специальную область памяти компьютера для временного хранения данных при выполнении операций Вырезать и Копировать;
 - Выделенную часть изображения;
 - Изменение изображения.

9. Установите взаимно-однозначное соответствие между инструментом и его названием.

1 – 	А – Текст.
2 – 	Б – Овал.
3 – 	В – Прямоугольник.

10. Как называются элементы интерфейса графического редактора Paint?

1 – ..., 2 – ..., 3 –



ТЕСТОВОЕ ЗАДАНИЕ

«ГРАФИЧЕСКИЕ РЕДАКТОРЫ ВЕКТОРНОГО ТИПА»

1. Элементарным объектом обработки в векторном графическом изображении является ...

- А) линия;
- Б) палитра цветов;
- В) символ;
- Г) точка изображения.

2. Векторное изображение – это...

- А) полноцветное изображение, обработанное особым образом и хранящееся в памяти компьютера;
- Б) изображение, создаваемое посредством математических вычислений, представляющее собой набор действий по созданию рисунка с помощью различных линий, фигур;
- В) чертежи, выполняемые графическими редакторами для проектных работ.

3. Большой размер файла — это недостаток ...

- А) фрактальной графики;
- Б) растровой графики;
- В) векторной графики.

4. Какие программы предназначены для работы с векторной графикой

- А) Компас3Д;

- Б) Photoshop;
- В) Corel Draw;
- Г) Blender;
- Д) Picasa;
- Е) Gmp.

5. В векторном редакторе можно выполнить следующие операции (выберите 2 варианта ответа):

- А) изменить разрешение изображения;
- Б) изменить размер объекта изображения;
- В) изменить форму объекта изображения;
- Г) изменить яркость (контрастность) изображения.

6. К достоинствам векторной графики можно отнести:

- А) фотографическое качество изображения;
- Б) возможность экспорт/импорт информации в различные графические форматы;
- В) возможность масштабирования изображения без потери качества;
- Г) относительно небольшой размер файлов.

7. Выберите один правильный ответ. Примитивом называются ...

- А) простые объекты;
- Б) точки на рисунке;
- В) простые карандаши;
- Г) картины, нарисованные ребенком.

8. Какие утверждения являются недостатками векторной графики?

- А) большой объем файлов;
- Б) сложности с выводом на печать;
- В) искажение при масштабировании;
- Г) невозможность создать изображение с фотографической точностью.

9. Укажите свойства присущие объекту "линия" в редакторе векторной графики:

- А) стиль, толщина, цвет;
- Б) цвет, длина, заливка;
- В) стиль, заливка, длина;
- Г) толщина, длина, кривизна.

10. Разрешение изображения измеряют в ...

- А) пикселях;
- Б) точках на дюйм (dpi);
- В) мм, см, дюймах.

ЭТАЛОН ОТВЕТОВ НА ТЕСТОВОЕ ЗАДАНИЕ «Графические редакторы векторного типа»

- 1. А
- 2. Б
- 3. Б
- 4. А, В
- 5. Б, В
- 6. В, Г
- 7. А
- 8. А, Г
- 9. А
- 10. Б

САМОСТОЯТЕЛЬНАЯ РАБОТА «ГРАФИЧЕСКИЕ РЕДАКТОРЫ ВЕКТОРНОГО ТИПА»

1. Составить презентацию по теме "Работа с объектами в векторных графических редакторах"
2. Изобразить с помощью данных инструментов животное.

ВОПРОСЫ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ «МЕТОДЫ СЖАТИЯ»

1. В каких случаях работы с изображением происходит потеря качества?
2. Какой из критериев оценки качества сжатого изображения имеет наивысшую чувствительность к "биению" отдельных пикселей?
3. Какой из критериев оценки качества сжатого изображения покажет, что изображение сильно испорчено, если увеличить его яркость хотя бы на 5%?
4. Какой из критериев оценки качества изображения используют на практике чаще всего?
5. Какой стандарт является стандартом де-факто для полноцветных изображений?
6. Какой стандарт является наиболее популярным для хранения изображений?
7. В каком формате хранятся большинство фотографий и полноцветных изображений в сети Интернет?
8. Продолжите фразу: алгоритм JPEG
9. Области какого размера оперирует алгоритм сжатия JPEG?
10. Какая компонента в цветовом пространстве YCrCb отвечает за яркость?
11. Как еще называют цветовое пространство YCrCb?
12. Какие компоненты в цветовом пространстве YCrCb отвечают за цвет?
13. На каком шаге алгоритма сжатия изображения JPRG, производится квантование?
14. На каком шаге алгоритма сжатия изображения JPRG, исходное изображение переводим в новое цифровое пространство?
15. В чем выражаются потери в низких частотах при больших значениях коэффициента gamma в алгоритме JPEG?

ТЕСТ «АНИМАЦИЯ. ОСНОВНЫЕ ПОНЯТИЯ. РЕДАКТОР ДЛЯ СОЗДАНИЯ АНИМАЦИИ»

1. Анимация —
 - процесс изменения размера, положения, цвета или формы объекта с течением времени.
 - процесс перехода объектов с места на место
 - процесс создания формы, цвета и расположения объекта.
2. Кадры —
 - изображения последовательных фаз движения объектов или их частей.
 - последовательная перестановка объектов со временем.
 - изображение постепенных ступеней перемещения объектов или их частей.
3. Анимация основана на свойстве человеческого зрения «...»
 - смотреть
 - помнить
 - следить
 - всматриваться
 - наблюдать

4. Первоначально, при подготовке кадров для анимации, каждый кадр рисовался отдельно и полностью, что отнимало много времени даже у большого коллектива художников. Затем стала использоваться послойная техника рисования объектов и фонов на ...

- шкурах животных
- прозрачных плёнках
- копировальной бумаге
- бересте
- Пакетах

5. Компьютерная анимация —

- создание компьютера с помощью анимации.
- создание анимации с помощью компьютера.
- создание движения компьютера с помощью программ анимации.

6. Работая над созданием компьютерной анимации, художник обычно прорисовывает ... и ... положение движущихся объектов, а все промежуточные состояния рассчитывает и изображает

- верхнее, нижнее, монитор
- начальное, конечное, компьютер
- правое, левое, программа

7. Выделяют два способа создания компьютерной анимации:

- 1.
- 2.

- покадровая анимация
- точечная анимация
- расчетная анимация
- автоматическая анимация
- растровая анимация

8. При создании покадровой анимации прорисовываются все фазы ... объекта.

- движения
- стояния
- лежания
- летания
- изображения
- цикла жизни

9. Анимация движения или формы

предполагает рисование только отдельных

- объектов
- кадров
- анимаций

10. Соотнесите:

1. Инструменты
2. Просмотр
3. Цвета
4. Параметры

Варианты ответов

- инструменты рисования и редактирования;
- способ просмотра;
- цвета обводки и заливки;
- настройка свойств выбранного инструмента.

ВОПРОСЫ И ОТВЕТЫ НА ЭКЗАМЕН МДК. 02.01 ПОДГОТОВКА ИНТЕРФЕЙСНОЙ ГРАФИКИ

1. Определение компьютерной графики.
2. Виды компьютерной графики. Их достоинства и недостатки.
3. Области применения компьютерной графики.
4. Дисплей как техническое средство компьютерной графики.
5. Векторная графика: суть, достоинства, недостатки и область применения.
6. Понятие графического примитива. Наиболее распространенные графические примитивы и операции над ними.
7. Математические основы векторной графики. Кривые Безье. Типы опорных точек.
8. Понятие линии, узла, сегмента, контура векторного изображения.
9. Основные редакторы векторной графики. Форматы файлов векторной графики
10. Растровая графика: суть, достоинства, недостатки и область применения.
11. Понятие пикселя и растра.
12. Разрешение растровой графики, виды разрешения.
13. Масштабирование растровых изображений.
14. Физический размер изображения.
15. Разрешение: типы и единицы измерения.
16. Связь разрешения и физического размера изображения.
17. Связь между параметрами изображения и размерами файла.
18. Формат графического файла. Типы форматов.
19. Векторные, растровые и универсальные форматы.
20. Алгоритмы сжатия графической информации. Их классификация.
21. Кодирование изображения. Понятие глубины цвета.
22. Цветовые палитры, их виды.
23. Понятие цветовой модели. Типы цветовых моделей.
24. Понятие цветового режима.

ЭТАЛОННЫЕ ОТВЕТЫ

1. Определение компьютерной графики.

Компьютерная графика (также машинная графика) – это область деятельности, в которой компьютеры используются как инструмент для синтеза (создания) изображений, так и для обработки визуальной информации, полученной из реального мира. Также компьютерной графикой называют результат такой деятельности.

Первые вычислительные машины не имели отдельных средств для работы с графикой, однако уже использовались для получения и обработки изображений. Программируя память первых электронных машин, построенную на основе матрицы ламп, можно было получать узоры.

В 1961 году программист Стивен Рассел возглавил проект по созданию первой компьютерной игры с графикой. Создание игры («Spacewar!») заняло около 200 человеко-часов. Игра была создана на машине PDP-1.

В 1963 году американский ученый Айвен Сазерленд создал программно-аппаратный комплекс Sketchpad, который позволял рисовать точки, линии и окружности на трубке цифровым пером. Поддерживались базовые действия с примитивами: перемещение, копирование и др. По сути, это был первый векторный редактор, реализованный на компьютере. Также программу можно назвать первым графическим интерфейсом, причем она являлась таковой еще до появления самого термина.

В середине 1960-х годов появились разработки в промышленных приложениях компьютерной графики. Так под руководством Т. Мофетта и Н. Тейлора фирма Itek разработала цифровую электронную чертежную машину.

В 1964 году General Motors представила систему автоматизированного проектирования DAC-1, разработанную совместно с IBM.

В 1968 году группой под руководством Н. Н. Константинова была создана компьютерная математическая модель движения кошки. Машина БЭСМ-4, выполняя написанную программу решения дифференциальных уравнений, рисовала мультфильм «Кошечка», который для своего времени являлся прорывом. Для визуализации использовался алфавитно-цифровой принтер.

Существенный прогресс компьютерная графика испытала с появлением возможности запоминать изображения и выводить их на компьютерном дисплее, электронно-лучевой трубке. Разработки в области компьютерной графики сначала двигались лишь академическим интересом и шли в научных учреждениях. Постепенно компьютерная графика прочно вошла в повседневную жизнь, стало возможным вести коммерчески успешные проекты в этой области. К основным сферам применения компьютерной графики относятся:

- [Графический интерфейс пользователя.](#)
- [Спецэффекты](#), Визуальные эффекты, [цифровая кинематография](#). [Цифровое телевидение](#), [Всемирная паутина](#), [видеоконференции](#)
- [Цифровая фотография](#) и существенно возросшие возможности по обработке фотографий.
- [Цифровая живопись.](#)
- Визуализация научных и деловых данных.
- [Компьютерные игры](#), системы [виртуальной реальности](#).
- [Системы автоматизированного проектирования.](#)
- [Компьютерная томография.](#)
- Компьютерная графика для кино и телевидения.
- [Лазерная графика.](#)

Компьютерная графика является также одной из областей научной деятельности. В области компьютерной графики защищаются диссертации, а также проводятся различные конференции.

2. **Виды компьютерной графики. Их достоинства и недостатки.**

Различают три вида компьютерной графики. Это растровая графика, векторная графика и фрактальная графика. Они отличаются принципами формирования изображения при отображении на экране монитора или при печати на бумаге.

Растровая графика.

Растровый метод – изображение представляется в виде прямоугольной матрицы, каждая ячейка которой представлена цветной точкой.

Растровые изображения состоят из прямоугольных точек – растр. Растровые изображения обеспечивают максимальную реалистичность, поскольку в цифровую форму переводится каждый мельчайший фрагмент оригинала. В цифровом изображении каждая точка растра (пиксель) предоставлена единственным параметром – цветом. Такие изображения сохраняются в файлах гораздо большего объема, чем векторные, поскольку в них запоминается информация о каждом пикселе изображения, т.е. качество растровых изображений зависит от их размера.

Растровую графику применяют при разработке электронных (мультимедийных) и полиграфических изданий.

Достоинства растровой графики:

- аппаратная реализуемость;
- программная независимость (форматы файлов, предназначенные для сохранения точечных изображений, являются стандартными, поэтому не имеют решающего значения, в каком графическом редакторе создано то или иное изображение);
- фотореалистичность изображений.

Недостатки растровой графики:

- значительный объем файлов (определяется произведением площади изображения на разрешение и на глубину цвета (если они приведены к единой размерности));
- принципиальные сложности трансформирования пиксельных изображений;
- эффект пикселизации – связан с невозможностью увеличения изображения для рассмотрения деталей. Поскольку изображение состоит из точек, то увеличение приводит к тому, что точки становятся крупнее. Никаких дополнительных деталей при увеличении растрового изображения рассмотреть не удастся, а увеличение точек раstra визуально искажает иллюстрацию и делает ее грубой;

Векторная графика

Векторный метод – это метод представления изображения в виде совокупности отрезков и дуг и т.д. В данном случае вектор – это набор данных, характеризующих какой-либо объект.

Векторные изображения состоят из контуров. Контур состоит из одного или нескольких смежных сегментов, ограниченных узлами. Сегменты могут иметь прямолинейную или криволинейную форму. Замкнутые контуры могут иметь залив. Заливка может быть сплошная, градиентная, узорная, текстурная. Любые контуры могут иметь обводку.

Контур – понятие математическое и толщины он не имеет. Чтобы контур сделать видимым ему придают обводку – линию заданной толщины и цвета, проведенную строго по контуру.

Векторные изображения строятся вручную, однако они могут быть также получены из растровых изображений с помощью трассировки. Программные средства для работы с векторной графикой предназначены в первую очередь для создания иллюстраций и в меньшей степени для их обработки. Такие средства широко используют в рекламных агентствах, дизайнерских бюро, редакциях и издательствах. Оформительские работы, основанные на применении шрифтов и простейших геометрических элементов, решаются средствами векторной графики много проще.

Достоинства векторной графики:

- полная свобода трансформации (изменение масштаба без потери качества и практически без увеличения размеров исходного файла);
- огромная точность;
- небольшой размер файла по сравнению с растровым изображением;
- прекрасное качество печати;
- отсутствие проблем с экспортом векторного изображения в растровое;
- объектно-ориентированный характер векторной графики (возможность редактирования каждого элемента изображения в отдельности);

Недостатки векторной графики:

- практически невозможно экспортировать из растрового формата в векторный (можно, конечно, трассировать изображение, хотя получить хорошую векторную картинку нелегко);
- невозможно применение обширной библиотеки эффектов, используемых при работе с растровыми изображениями.

Фрактальная графика.

Программные средства для работы с фрактальной графикой предназначены для автоматической генерации изображений путем математических расчетов. Создание фрактальной художественной композиции состоит не в рисовании или оформлении, а в программировании. Фрактальная графика, как и векторная – вычисляемая, но отличается от неё тем, что никакие объекты в памяти компьютера не хранятся. Изображение строится по уравнению (или по системе уравнений), поэтому ничего, кроме формулы, хранить не надо. Изменив коэффициенты в уравнении, можно получить совершенно другую картину. Способность фрактальной графики моделировать образы живой природы

вычислительным путем часто используют для автоматической генерации необычных иллюстраций.

Фрактал – это геометрическая фигура, состоящая из частей, и которая может быть поделена на части, каждая из которых будет представлять уменьшенную копию целого (по крайней мере, приблизительно)

Основное свойство фракталов — самоподобие. Любой микроскопический фрагмент фрактала в том или ином отношении воспроизводит его глобальную структуру. В простейшем случае часть фрактала представляет собой просто уменьшенный целый фрактал.

3. **Области применения компьютерной графики.**

Область применения компьютерной графики не ограничивается одними художественными эффектами. Во всех отраслях науки, техники, медицины, в коммерческой и управленческой деятельности используются построенные с помощью компьютера схемы, графики, диаграммы, предназначенные для наглядного отображения разнообразной информации. Конструкторы, разрабатывая новые модели автомобилей и самолетов, используют трехмерные графические объекты, чтобы представить окончательный вид изделия. Архитекторы создают на экране монитора объемное изображение здания, и это позволяет им увидеть, как оно впишется в ландшафт.

Научная графика. Первые компьютеры использовались лишь для решения научных и производственных задач. Чтобы лучше понять полученные результаты, производили их графическую обработку, строили графики, диаграммы, чертежи рассчитанных конструкций. Первые графики на машине получали в режиме символьной печати. Затем появились специальные устройства - графопостроители (плоттеры) для вычерчивания чертежей и графиков чернильным пером на бумаге. Современная научная компьютерная графика дает возможность проводить вычислительные эксперименты с наглядным представлением их результатов.

Деловая графика - область компьютерной графики, предназначенная для наглядного представления различных показателей работы учреждений. Плановые показатели, отчетная документация, статистические сводки — вот объекты, для которых с помощью деловой графики создаются иллюстративные материалы. Программные средства деловой графики включаются в состав электронных таблиц.

Конструкторская графика используется в работе инженеров-конструкторов, архитекторов, изобретателей новой техники. Этот вид компьютерной графики является обязательным элементом САПР (систем автоматизации проектирования). Средствами конструкторской графики можно получать как плоские изображения (проекции, сечения), так и пространственные трехмерные изображения.

Иллюстративная графика — это произвольное рисование и черчение на экране компьютера. Пакеты иллюстративной графики относятся к прикладному программному обеспечению общего назначения. Простейшие программные средства иллюстративной графики называются графическими редакторами.

Художественная и рекламная графика - ставшая популярной во многом благодаря телевидению. С помощью компьютера создаются рекламные ролики, мультфильмы, компьютерные игры, видеоуроки, видеопрезентации. Графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти. Отличительной особенностью этих графических пакетов является возможность создания реалистических изображений и «движущихся картинок». Получение рисунков трехмерных объектов, их повороты, приближения, удаления, деформации связано с большим объемом вычислений. Передача освещенности объекта в зависимости от положения источника света, от расположения теней, от фактуры поверхности, требует расчетов, учитывающих законы оптики.

Компьютерная анимация — это получение движущихся изображений на экране дисплея. Художник создает на экране рисунки начального и конечного положения

движущихся объектов, все промежуточные состояния рассчитывает и изображает компьютер, выполняя расчеты, опирающиеся на математическое описание данного вида движения. Полученные рисунки, выводимые последовательно на экран с определенной частотой, создают иллюзию движения.

4. Дисплей как техническое средство компьютерной графики.

В XIX веке во Франции возникла техника живописи, которую назвали пуантилизмом: рисунок составлялся из разноцветных точек, наносимых кистью на холст. Подобный принцип используется и в компьютерах. Точки на экране компьютера выстроены в ровные ряды. Совокупность точечных строк образует графическую сетку, или растр.

Одна точка носит название видеопиксель (краткое название — пиксель). Слово «пиксель» происходит от английского picture element — элемент рисунка. Чем гуще сетка пикселей на экране, тем лучше качество изображения. Размер графической сетки обычно представляется в форме произведения числа точек в строке на число строк: $M \times N$.

На современных мониторах используются, например, такие размеры графической сетки:

- 1280 x 1024;
- 1366 x 768;
- 1920 x 1080 и более.

Размер монитора характеризуется длиной диагонали его экрана, выраженной в дюймах (1 дюйм = 2,54 см). Бывают мониторы с диагональю 15, 17, 19 и более дюймов.

Существуют мониторы, работа которых основана на разных физических принципах. Первоначально на ПК использовались только мониторы на основе электронно-лучевой трубки — ЭЛТ-мониторы. На экране такого монитора пиксель образуется люминесцирующим веществом, которое светится под воздействием луча, испускаемого электронной пушкой. Такой луч пробегает по порядку (сканирует) все строки сетки пикселей. При этом он модулируется: на точки, которые должны светиться, падает, а на темных точках прерывается

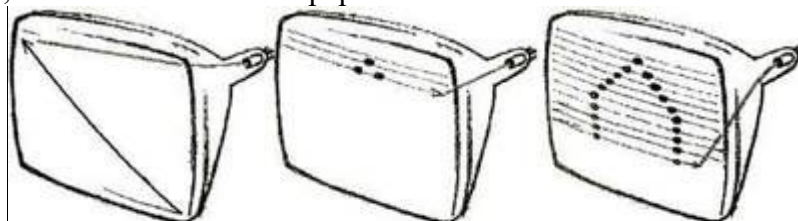


Рис. 4.8. Получение растрового изображения в электронно-лучевой трубке

Поскольку после прекращения воздействия электронного луча на точку экрана ее свечение быстро затухает, постольку сканирование периодически повторяется с высокой частотой (75-85 раз в секунду и более). При такой частоте наше зрение не замечает мерцания изображения.

Первоначально на компьютерах использовались черно-белые мониторы. На черно-белом экране пиксель, на который падает электронный луч, светится белым цветом. Неосвещенный пиксель — черная точка. При изменении интенсивности электронного потока получают промежуточные серые тона (оттенки).

Каждый пиксель на цветном экране — это совокупность трех точек разного цвета: красного, зеленого и синего. Эти точки расположены так близко друг к другу, что нам они кажутся слившимися в одну точку.

Из сочетаний красного, зеленого и синего цветов складывается вся красочная палитра на экране.

Электронная пушка цветного монитора испускает три луча. Каждый луч вызывает свечение точки только одного цвета. Для этого в мониторе используется специальная фокусирующая система.

Все большее распространение получают жидкокристаллические мониторы — ЖК-мониторы. По сравнению с электронно-лучевыми мониторами они значительно меньше по весу, имеют плоскую форму. При работе с ЖК-монитором меньше устают глаза.

На рисунке дана схема системы вывода изображения на экран. Она включает в себя монитор (другое название — дисплей) и видеоадаптер, который через информационную магистраль связан с центральным процессором и оперативной памятью.



Рис. 4.9. Схема системы вывода изображения на экран

Видеоадаптер (другое название — видеокарта) — устройство, управляющее работой графического дисплея. Видеоадаптер состоит из двух частей: видеопамяти и дисплейного процессора.

Видеопамять предназначена для хранения видеоинформации — двоичного кода изображения, выводимого на экран. В видеопамяти содержится информация о состоянии каждого пикселя экрана.

Видеопамять — это электронное энергозависимое запоминающее устройство. На современных компьютерах ее размер составляет от сотен мегабайтов до нескольких гигабайтов.

Дисплейный процессор — вторая составляющая видеоадаптера. Дисплейный процессор читает содержимое видеопамяти и в соответствии с ним управляет работой дисплея.

Таким образом, к видеопамяти имеют доступ два процессора: центральный и дисплейный. Центральный процессор записывает видеоинформацию, а дисплейный периодически читает ее и передает на монитор, на котором эта информация превращается в изображение.

5. Векторная графика: суть, достоинства, недостатки и область применения.

В векторной графике создают изображения при помощи математических формул, которые представляют собой своего рода набор объектов, отрезков. Растровая графика — это изображения из пикселей, совокупностей точек.

Эти два подраздела неразрывно связаны между собой. К примеру, просматривая изображение, вы не увидите на мониторе набор формул. Вам будет показана непосредственно картинка. Векторная графика преобразуется в растровую перед выводом на экран. Этот процесс происходит автоматически посредством работы видеокарты.

В основе векторной графики лежит линия. Формулы задают ее координаты. Таким образом, создается контур изображения. Это понятие подразумевает собой 2D графику.

Свойства картинки (цвет заливки, толщина линий и т.д.) находятся в прерогативе растровой графики.

Векторы имеют ряд преимуществ наряду с другими способами передачи графической информации. Они способны наиболее точно передать изображение и чаще всего используются для создания точных графиков и чертежей, которые не требуют фотореализм.

Достоинства векторной графики:

1. Основным положительным качеством данного способа передачи информации является его компактность. Файл, содержащий описание (формулы), может занимать мало места, независимо от того, какого размера будет сам объект.

2. Благодаря формульному описанию, сами объекты можно неограниченно увеличивать. При этом качество изображения не будет изменяться в худшую сторону.

3. Объекты можно форматировать, изменять их положение, группировать. Качество остается неизменным.

4. Над объектами можно производить математические преобразования, а также пересечения, дополнения.

5. Толщина контурной линии может быть постоянной, независимо от масштаба самого изображения.

Однако, несмотря на большое количество положительных качеств, векторная графика обладает и недостатками:

1. В том случае, если детализация изображения сложная, размер исполнительного файла будет достаточно большим. Исходя из этого, может возникнуть трудность реалистичной передачи графики.

2. Обратный переход из растровой графики происходит достаточно сложно и влияет на качество вектора.

3. Сложности с совместимостью программного обеспечения.

6. **Понятие графического примитива. Наиболее распространенные графические примитивы и операции над ними.**

Различные графические векторные редакторы имеют свои отличительные особенности, могут содержать различный функционал и своеобразный интерфейс. Но, не смотря на различия, в основе любого векторного редактора лежит стандартный набор графических примитивов.

Графические примитивы — это заранее определенные элементы, которые можно поместить в чертеж при помощи одной команды. Каждый графический примитив формируется на основании геометрического описания объекта. Примитивы можно классифицировать как односложные и составные, плоские и объемные.

Основные графические примитивы:

1. Точка — это один из простейших примитивов, который характеризуется тремя пространственными координатами X, Y и Z.

2. Линия — это часть прямой линии, задаваемая двумя крайними точками с нулевой шириной (1 пиксель). Линия является наиболее фундаментальным примитивом для любого чертежа.

3. Полилиния - ломаная линия.

4. Прямоугольник - фигура, для которой указываются координаты начальной и противоположной угловых точек.

5. Дуга - часть окружности, которая геометрически определяется центром, радиусом и двумя центральными углами.

6. Круг (эллипс) - часть плоскости, ограниченная окружностью.

7. Фигура — это часть плоскости, ограниченная четырехугольником (треугольником).

Каждый примитив формируется своей командой, чаще всего совпадают по имени с примитивом. Для некоторых примитивов пользователю предлагается несколько способов построения одного и того же примитива по различным исходным данным, например окружность можно построить по центру и радиусу, по центру и диаметру, по трем точкам на плоскости и т.д. Каждый примитив обладает рядом свойств (например, принадлежность слою, цвет, видимость, тип линии и т.д.).

Примитивы имеют следующие свойства:

- цвет;

- тип линий;

- масштаб типа линий;
- принадлежность слою;
- уровень и высота.

Над примитивами можно выполнять следующие операции:

- создавать;
- удалять;
- устанавливать свойства;
- получать копии;
- перемещать;
- поворачивать;
- отображать зеркально;
- масштабировать;
- штриховать;
- закрашивать и др.

7. Математические основы векторной графики. Кривые Безье. Типы опорных точек.

Если основным элементом растровой графики является пиксель, то в случае векторной графики в роли базового элемента выступает линия. Любой объект состоит из набора линий, соединенных между собой узлами. Фрагмент линии, соединяющий соседние узлы, называется сегментом. Сегмент может быть задан с помощью уравнения прямой или уравнения кривой линии, требующих для своего описания разного количества параметров.

Кривые Безье — это частный вид кривых третьего порядка, требующий для своего описания меньшего количества параметров. В основе построения кривых Безье лежит использование двух касательных, проведенных к крайним точкам отрезка линии. На кривизну (форму) линии влияет угол наклона и длина отрезка касательной, значениями которых можно управлять в интерактивном режиме путем перетаскивания их концевых точек. Таким образом, касательные выполняют функции виртуальных рычагов, позволяющих управлять формой кривой.

В начале 70-х годов профессор Пьер Безье, проектируя на компьютере корпуса автомобилей «Рено», впервые применил для этой цели особый вид кривых, описываемых уравнением третьего порядка, которые впоследствии стали известными под названием кривые Безье.

Эти линии имеют особое значение как для векторной, так и растровой графики. Кроме того, появление кривых Безье вызвало настоящий переворот и в трехмерной графике. В настоящее время кривые Безье присутствуют в любом современном графическом пакете. Также большинство компьютерных шрифтов состоят из кривых Безье. Гибкость в построении и редактировании кривых Безье во многом определяется характеристиками узловых точек.

Узлы (опорные точки). В векторных редакторах (как, впрочем, и в растровых) форму произвольного контура изменяют путем манипуляции узлами. Узлы можно перемещать, изменять их тип, добавлять, удалять.

Таким образом, в основе всех процедур, связанных с редактированием (отчасти и созданием) любого типа контуров, лежит работа с узлами.

Типы узловых точек. Касательная линия всегда является касательной к сегменту кривой в узловой точке. Ее наклон и длина определяют наклон и радиус изгиба соответствующего криволинейного сегмента. Перемещение узловых точек и настройка касательных линий позволяют изменять форму криволинейных сегментов. Вид касательных линий и соответственно методы управления кривизной сегмента в узловой точке определяются типом узловой точки. Различают три типа узловых точек:

- гладкий узел;
- симметричный узел;

- острый узел.

8. **Понятие линии, узла, сегмента, контура векторного изображения.**

Основным объектом векторной графики является линия. В некоторых редакторах ее называют кривой (curve). При этом прямая (line) рассматривается как частный случай кривой. В некоторых программах вместо понятия кривая используется понятие контур (path). По-видимому, понятие контур наиболее адекватно отражает суть, поскольку контур может быть и прямой, и кривой, и фигурой, и ломаной. Далее мы будем придерживаться термина контур.

Каждый векторный контур может иметь две или более опорных точек (узлов). В некоторых редакторах их называют узлами (nodes).

Элемент векторного контура, заключенный между двумя опорными точками (узлами), называется сегментом векторного контура. Если контур имеет более двух опорных точек (узлов), то он состоит из нескольких сегментов. Форму векторного контура изменяют перемещением опорных точек (узлов), изменением их свойств, добавлением новых опорных точек или удалением части опорных точек векторного контура.

Векторный контур может быть открытым или замкнутым. Если последняя опорная точка (узел) векторного контура одновременно является и его первой точкой (простого геометрического совпадения этих точек недостаточно), то векторный контур считается замкнутым. В противном случае он открыт. Свойства замкнутых и открытых векторных контуров различаются.

Векторный контур является элементарным графическим объектом. Из контуров можно создавать новые объекты или их группы. С несколькими контурами можно выполнить операции группирования, комбинирования и объединения. Эти операции образуют, соответственно: группу объектов, составной векторный контур или новый контур. В операции группирования каждый контур группы сохраняет свои опорные точки (узлы) и свойства. В операции комбинирования векторные контуры сохраняют свои опорные точки (узлы), но свойства составного контура становятся новыми. В операции объединения образуются новые опорные точки и изменяются свойства исходных объектов.

Узлом называется точка на плоскости изображения, определяющая положение одного из концов сегмента кривой и его направление в месте выхода из узла. Сегментом называется часть линии, соединяющая два смежных узла. Эти два типа элементарных объектов неразрывно связаны, они не могут существовать в отрыве друг от друга или вне линии. В замкнутой линии число узлов и сегментов одинаково, в незамкнутой – различается на единицу.

Имеющиеся в составе линии узлы могут быть краевыми и промежуточными. Краевым называется узел, смежный только с одним сегментом линии.

Промежуточный узел располагается между двумя смежными сегментами. Кроме того, один из узлов линии является начальным. В незамкнутой линии это один из краевых узлов, в замкнутой – один из промежуточных.

Сегменты линии подразделяются на прямолинейные и криволинейные. Прямолинейным называется сегмент, представляющий собой отрезок прямой линии, криволинейным – сегмент, в любой точке которого радиус кривизны отличен от бесконечности.

Различают три типа узлов:

- точки излома (узел, в котором наклон и длину направляющих рукояток можно изменять независимо друг от друга. К точкам излома условно относят также узлы, в которых соприкасаются два прямолинейных сегмента);
- сглаженные узлы (узел, в котором направляющие рукоятки лежат на общей прямой, проведенной через выделенный узел. По крайней мере, один из сегментов, примыкающих к сглаженному узлу, должен быть криволинейным);

- симметричные узлы (узел, в котором направляющие рукоятки лежат на общей прямой, проведенной через выделенный узел, и имеют одинаковую длину. Симметричные узлы могут располагаться только на стыке двух криволинейных сегментов).

9. Основные редакторы векторной графики. Форматы файлов векторной графики.

Для создания векторных изображений используют такие редакторы, как:

1. Corel Draw.
2. Adobe Illustrator.
3. Adobe Free Hand и др.

Corel Draw. С ними работают профессионалы дизайна и рекламы. Это самый популярный графический редактор среди любителей, начинающих профессионалов и продвинутых художников-графиков. Актуальные версии Corel Draw позволяют выбирать предустановки, где расположение рабочих областей напоминает Photoshop или Illustrator. Благодаря поддержке 64-разрядных ОС обеспечивается высокая производительность программы. Онлайн-сервис Content Center дает возможность обмениваться цветовыми градиентами и векторными узорами с другими пользователями.

Adobe Illustrator. Данное ПО – мировой стандарт для отрасли полиграфии. Именно Illustrator позволяет качественно переносить изображения из компьютера на бумагу или любой другой запечатываемый носитель. Предусмотрен революционный уровень масштабирования – до 64000 %.

Особым удобством отличается интерфейс. Пользователи без труда находят все необходимые инструменты. Кроме того, Adobe Illustrator корректно работает с вектором. Созданные в программе изображения, как правило, без изменений выводят на фотонаборном автомате.

Интерактивные инструменты фигур обеспечивают поддержку динамической коррективы. Символы можно изменять с помощью заливки, обводки и т. п. Также предусмотрены масштабирование, повороты, наклоны, отражения и т. д. При этом связь с родительским символом не нарушается.

Если проект не был сохранен, для его восстановления необходимо просто перезагрузить Illustrator. Программа предлагает варианты диагностики и ликвидации ошибок.

ПО можно устанавливать на мобильных устройствах с такими операционными системами, как iOS, Android и Windows.

Adobe Free Hand. Интерфейс данного ПО более насыщен инструментами и различными возможностями редактирования, чем Illustrator. При этом методы работы в двух программах схожи. После использования Corel Draw привыкнуть к Free Hand проще, чем к Illustrator.

С помощью представленной [программы для векторной графики](#) можно создавать многостраничные сайты. Adobe Free Hand уменьшает затраты времени на разработку и редактирование иллюстраций. Для этого дизайнеры и художники используют шаблоны фонов и страниц, библиотеки символов и т. д.

Форматы векторной графики:

- *AI* (Adobe Illustrator) — векторный формат файлов программы Adobe Illustrator.
- *EPS* (Encapsulated Post Script).
- *CDR* (Corel Draw) — векторный формат файлов программы Corel Draw.
- *CMX* (Corel Presentation Exchange).
- *EMF* (Enhanced Metafile, расширенный метафайл Windows).
- *WMF* (Windows Metafile).
- *SVG* (Scalable Vector Graphics).

EPS — он хорошо известен тем, кто продает свою векторную графику на микростоках. EPS (Encapsulated Post Script) может хранить как растровую, так и векторную графику. EPS — это один из самых универсальных форматов векторной графики, он поддерживается большинством популярных редакторов векторной графики. И именно поэтому формат EPS 10 и EPS 8 принимаю на [микростоках](#) как основной формат векторной графики.

CMX — векторный формат графических программ компании Corel. Его главное предназначение — передача рисунков между разными программами.

EMF и WMF — довольно простые форматы векторной графики, позволяющие хранить основную информацию. EMF может содержать как векторную, так и растровую информацию в одном файле. EMF лучше чем WMF поддерживает кривые, используемые в рисунках Flash, прекрасно воспринимает «составленные» из нескольких программ векторные изображения (к примеру Chem Win и MS Power Point). WMF более старый, чем EMF формат. Он широко применяется в офисном пакете MS Office.

SVG — основан на XML разметке, используется для двухмерной векторной графики, как анимированной, так и неподвижной.

10. Растровая графика: суть, достоинства, недостатки и область применения.

Растровое изображение — это изображение, в основе которого стоит таблица. Ячейки таблицы — это пиксели.

Пиксель — это единица измерения размеров растрового изображения. Один пиксель — это одна клеточка в растровом изображении. Соответственно растровые изображения состоят из пикселей, только они настолько малы, что в нормальном виде их трудно различить. Каждый пиксель растрового изображения имеет следующие свойства: цвет и координаты расположения пикселя. Чем больше количество пикселей и чем меньше их размеры, тем лучше выглядит изображение. Большие объемы данных — это основная проблема при использовании растровых изображений. Для активных работ с большеразмерными иллюстрациями типа журнальной полосы требуются компьютеры с исключительно большими размерами оперативной памяти (128 Мбайт и более). Разумеется, такие компьютеры должны иметь и высокопроизводительные процессоры.

Достоинства растровой графики:

1. Возможность воспроизведения изображений любого уровня сложности. Количество деталей, воспроизводимых на изображении во многом, зависит от количества пикселей.

2. Точная передача цветовых переходов.

3. Наличие множества программ для отображения и редактирования растровой графики. Абсолютное большинство программ поддерживают одинаковые форматы файлов растровой графики. Растровое представление, пожалуй, самый «старый» способ хранения цифровых изображений.

Недостатки растровой графики:

1. Большой размер файла. Фактически для каждого пикселя приходится хранить информацию о его координатах и цвете.

2. Невозможность масштабирования (в частности, увеличения) изображения без потери качества.

Растровая графика применяется:

- для хранения и обработки полутоновых изображений (сканированные или изначально созданные на компьютере картины, фотографии);
- в веб-дизайне. Применяемые на веб-страницах изображения, как правило не велики, а вывод их на экран осуществляется самим веб-обозревателем без применения дополнительных программ.

11. Понятие пикселя и растра.

Растр — это матрица ячеек, называемых пикселями. Любой пиксел имеет свой цвет. Совокупность пикселей различного цвета образует изображение. В зависимости от расположения пикселей в пространстве различают квадратный, прямоугольный, гексагональный и другие типы растров. Для описания расположения пикселей используют разнообразные системы координат. Общим для всех таких систем является то, что координаты пикселей образуют дискретный ряд значений. Часто используется система целых координат — номеров пикселей. Начало координат, как правило, располагается в левом верхнем углу.

Размер растра, как правило, измеряется количеством пикселей по горизонтали и вертикали. Разрешающая способность характеризует расстояние между соседними пикселями — шаг дискретной сетки растра. Разрешающую способность измеряют количеством пикселей на единицу длины.

Наиболее популярной единицей измерения является **dpi** (dots per inch) — количество пикселей в одном дюйме длины (2.54 см). Шаг не тождественен размерам пикселей: размер пикселей может равняться шагу, а может принимать другие значения. Форма пикселей растра определяется особенностями устройства графического вывода. Например, пиксели могут иметь форму прямоугольника или квадрата, а также иметь круглую форму и по размерам могут не равняться шагу растра.

Другой важный параметр - количество цветов (глубина цвета). Отметим, что глаз человека способен различать порядка 350000 цветов. Изображения можно классифицировать следующим образом:

1. Двухцветные (бинарные) (1 бит на пиксел). Характерный и наиболее распространенный пример – черно-белые изображения.
2. Полутоновые. Градации серого или другого цвета. Например, 256 градаций (1 байт на пиксел).
3. Цветные изображения. Глубина цвета 16 бит на пиксел (65536 цветов) называется High Color; 24 бита на пиксел (16.7 млн. цветов) — True Color. В компьютерных графических системах используют и большую глубину цвета — 32, 48 и более битов на пиксел.

Существуют различные методы улучшения качества растровых изображений, которые основываются на субъективном восприятии разрешающей способности и количества цветов. Как правило, улучшение одной характеристики происходит за счет ухудшения другой. Также в растровых системах при невысокой разрешающей способности существует проблема ступенчатого эффекта (aliasing): при большом шаге сетки растра пиксели линий образуют нечто похожее на ступени лестницы.

12. Разрешение растровой графики, виды разрешения.

Разрешение — [величина, определяющая количество точек \(элементов растрового изображения\)](#) на единицу площади (или единицу длины). Термин обычно применяется к изображениям в цифровой форме, хотя его можно применить, например, для описания уровня грануляции фотоплёнки, фотобумаги или иного физического носителя. Более высокое разрешение (больше [элементов](#)) типично обеспечивает более точные представления оригинала. Другой важной характеристикой изображения является [разрядность цветовой палитры](#).

Как правило, разрешение в разных направлениях одинаково, что даёт пиксель квадратной формы. Но это не обязательно — например, горизонтальное разрешение может отличаться от вертикального, при этом элемент изображения (пиксель) будет не квадратным, а прямоугольным. Более того, возможна не квадратная решётка элементов изображения, а например [шестигранная](#) (гексагональная) или вовсе не регулярная ([стохастическая](#)), что не мешает говорить о максимальном количестве точек или управляемых элементов изображения на единицу длины или площади.

Ошибочно под разрешением понимают размеры фотографии, экрана монитора или изображения в [пикселях](#). Размеры растровых изображений выражают в виде

количества [пикселей](#) по горизонтали и вертикали, например: 1600×1200. В данном случае это означает, что ширина изображения составляет 1600, а высота — 1200 точек (такое изображение состоит из 1 920 000 точек, то есть примерно 2 [мегапикселя](#)). Количество точек по горизонтали и вертикали может быть разным для разных изображений. Изображения, как правило, хранятся в виде, максимально пригодном для отображения экранами мониторов — они хранят цвет пикселей в виде требуемой яркости свечения излучающих элементов экрана ([RGB](#)), и рассчитаны на то, что пиксели изображения будут отображаться пикселями экрана один к одному. Это обеспечивает простоту вывода изображения на экран.

При выводе изображения на поверхность экрана или бумаги, оно занимает прямоугольник определённого размера. Для оптимального размещения изображения на экране необходимо согласовывать количество точек в изображении, пропорции сторон изображения с соответствующими параметрами устройства отображения. Если пиксели изображения выводятся пикселями устройства вывода один к одному, размер будет определяться только разрешением устройства вывода. Соответственно, чем выше разрешение экрана, тем больше точек отображается на той же площади и тем менее зернистой и более качественной будет ваша [картинка](#). При большом количестве точек, размещённом на маленькой площади, глаз не замечает мозаичности рисунка.

Справедливо и обратное: малое разрешение позволит глазу заметить растр изображения («ступеньки»). Высокое разрешение изображения при малом размере плоскости отображающего устройства не позволит вывести на него всё изображение, либо при выводе изображение будет «подгоняться», например для каждого отображаемого пикселя будут усредняться цвета попадающей в него части исходного изображения. При необходимости крупно отобразить изображение небольшого размера на устройстве с высоким разрешением приходится вычислять цвета промежуточных пикселей. Изменение фактического количества пикселей изображения называется [передискретизация](#), и для неё существуют целый ряд алгоритмов разной сложности.

При выводе на бумагу такие изображения преобразуются под физические возможности принтера - проводится цветоделение, масштабирование, и растеризация для вывода изображения красками фиксированного цвета и яркости, доступными принтеру.

Большинство форматов графических файлов позволяют хранить данные о желаемом масштабе при выводе на печать, то есть о желаемом разрешении в [dpi](#) (эта величина говорит о количестве точек на единицу длины: например 300 dpi означает 300 точек на один [дюйм](#)). Это исключительно справочная величина. Как правило, для получения распечатка фотографии, который предназначен для рассматривания с расстояния порядка 40 — 45 сантиметров, достаточно разрешения 300 dpi. Исходя из этого можно рассчитать, какого размера отпечаток можно получить из имеющегося изображения или какого размера изображение надо получить, чтоб затем сделать отпечаток нужного размера.

Для обозначения разрешающей способности различных процессов преобразования изображений (сканирование, печать, растеризация и т. п.) используют следующие термины:

1. [dpi](#) ([англ. dots per inch](#)) — количество точек на дюйм.
2. [ppi](#) ([англ. pixels per inch](#)) — количество пикселей на дюйм.
3. [lpi](#) ([англ. lines per inch](#)) — количество линий на дюйм, разрешающая способность графических планшетов (дигитайзеров).
4. [spi](#) ([англ. samples per inch](#)) — количество семплов на дюйм; плотность дискретизации ([англ. sampling density](#)), в том числе разрешение [сканеров изображений](#).

По историческим причинам величины стараются приводить к [dpi](#), хотя с практической точки зрения [ppi](#) более однозначно характеризует для потребителя процессы печати или сканирования. Измерение в [lpi](#) широко [используется в полиграфии](#). Измерение в [spi](#) используется для описания внутренних процессов устройств или алгоритмов.

Для векторных изображений, в силу принципа построения изображения, понятие разрешения неприменимо.

13. Масштабирование растровых изображений.

В компьютерной графике и цифровой обработке изображений, **масштабирование** описывает изменение размера цифрового изображения.

При масштабировании растровой графики изменяется разрешение их изображения. Это означает, что новое изображение с большим или меньшим количеством точек изображения (пикселей) создается из данной растровой графики. Когда количество пикселей увеличивается (апскейлинг), это обычно связано с видимой потерей качества. С точки зрения цифровой обработки сигналов, масштабирование растровой графики является примером преобразования частоты дискретизации, преобразования дискретного сигнала из одной частоты дискретизации (здесь - локальной частоты дискретизации) в другую.

Масштабирование изображений используется, среди прочего, в веб-браузерах, программах обработки изображений, средствах просмотра изображений и файлов, программных лупах, с цифровым масштабированием, увеличением секций и созданием изображений предварительного просмотра, а также при выводе изображений на экраны или принтеры.

Методы масштабирования растровой графики.

Масштабирование с фильтром реконструкции. Программы редактирования изображений обычно предлагают несколько методов масштабирования. Наиболее часто поддерживаемые методы - повторение пикселей, билинейная и бикубическая интерполяция - масштабируют изображение с помощью фильтра восстановления. При масштабировании указанная сетка изображения должна быть перенесена в выходную сетку разных размеров. Таким образом, масштабирование можно четко представить, поместив пиксельную сетку выходного изображения, подлежащего вычислению, поверх пиксельной сетки входного изображения. Каждому пикселю выходного изображения назначается значение цвета, которое вычисляется из пикселей входного изображения, которые находятся поблизости. Используемый фильтр реконструкции определяет, какие пиксели входного изображения используются для вычисления и как их значения цвета взвешиваются.

Масштабирование с использованием радиально-симметричного восстанавливающего фильтра. Значение цвета выходного изображения вычисляется как сумма значений цвета входного изображения, взвешенная фильтром восстановления. При масштабировании фильтр двумерной реконструкции помещается на каждый пиксель выходного изображения. Значение цвета вычисляется как сумма значений цвета пикселей входного изображения, которые перекрываются несущей фильтра восстановления, взвешенных по значению фильтра восстановления в этих пикселях. Обычно фильтры реконструкции уменьшаются по мере удаления от центра. В результате значения цвета, расположенные близко к выходному пикселю, имеют больший вес, а те, которые находятся дальше - меньший. Размер фильтра восстановления зависит от раstra входного изображения и, в случае уменьшения, от раstra выходного изображения.

Повторение пикселей. При повторении пикселей, также называемом ближайшим соседом, значение цвета ближайшего пикселя входного изображения присваивается каждому пикселю выходного изображения. Уменьшение изображений с помощью этого метода может привести к серьезному искажению, которое проявляется в виде артефактов изображения. При увеличении с помощью повторения пикселей получается блочное, «пиксельное» представление. В случае увеличения повторение пикселей соответствует реконструкции с помощью прямоугольного фильтра размером 1×1 . Такой фильтр перекрывает только один пиксель входного изображения, а именно ближайший.

Билинейная интерполяция. При масштабировании с использованием билинейной интерполяции значение цвета выходного изображения вычисляется из четырех

ближайших значений цвета входного изображения. При билинейной интерполяции значение цвета пикселя в выходном изображении интерполируется из четырех соседних значений цвета во входном изображении. Этот фильтр является разделяемым и может быть рассчитан как серия интерполяций с помощью одномерного фильтра реконструкции. Сначала вычисляется интерполированное значение цвета для каждой из двух задействованных линий изображения, а затем выполняется интерполяция между этими двумя вертикальными точками.

Бикубическая интерполяция. При бикубической интерполяции значение цвета выходного изображения интерполируется из соседних значений цвета входного изображения с использованием кубических сплайнов. Есть несколько общих кубических сплицев с разными свойствами; поэтому термин «бикубическая интерполяция» неоднозначен.

Процесс супер разрешения. При масштабировании с использованием так называемых методов сверхвысокого разрешения (SR) используется информация из соседних изображений одного изображения из последовательности. Более высокое качество достигается за счет больших вычислительных затрат. Такие методы особенно актуальны для медицины, астрофотографии и судебно-медицинской экспертизы записей с камер наблюдения.

Искажение изображения в зависимости от содержимого. Зависящее от содержимого искажение изображения — это процесс, при котором пропорции изображений изменяются при сохранении соответствующего содержимого изображения.

Масштабирование пиксельной графики. Для увеличения пиксельных изображений с резкими краями были разработаны специальные алгоритмы, которые обеспечивают лучшие результаты, чем методы, описанные выше для этих типов изображений. Некоторые из этих алгоритмов могут масштабировать изображения в два, три или четыре раза по сравнению с их исходным размером в реальном времени.

Физический размер изображения.

Размеры изображения:

1. Физический размер изображения.
2. Логический размер изображения.
3. Разрешение.

Чтобы понимать принципы работы с растровой графикой, важно разобраться в размерах изображения – не менее важно, чем знать теорию цвета. Точно так же, как неполноцветное изображение ограничивает нас в выразительных средствах и иногда не позволяет полностью передать все нюансы рисунка, малый размер изображения не позволит нам передать мелкие детали рисунка или помешает использовать его в большом размере.

Осложняет дело и то, что у растрового изображения есть два разных размера. С одной стороны, на экране монитора или на листе бумаги изображение можно измерить линейкой и получить размер в сантиметрах. С другой стороны, существует размер в пикселях – он определяет количество пикселей в изображении.

Размер в пикселях принято называть физическим размером изображения. Это твердая, неизменная величина, которая достаточно точно описывает изображение, такой размер является абсолютным.

Размер в сантиметрах, миллиметрах и других линейных единицах измерения принято называть логическим размером, поскольку он описывает только сиюминутное состояние изображения – тот размер, с которым его сейчас вывели на экран монитора или на печать, и это размер относительный.

Оба размера связаны между собой третьей величиной, которая именуется разрешением.

Физический размер изображения – это количество пикселей в изображении по ширине и по высоте. Таким образом, мы получаем размеры изображения в пикселях. Чем

больше пикселей в изображении и чем больше его физический размер, тем выше может быть качество изображения. При большом количестве пикселей мы можем сохранить более мелкие детали изображения, которые были бы не видны при меньшем количестве. Поскольку пиксел является наименьшей деталью изображения, детали размером меньше 1 пикселя не могут быть сохранены в изображении.

Нужно особо подчеркнуть, что хотя размер изображения в пикселях можно изменить (используя графический редактор), но улучшить его качество и «проявить» недостающие детали невозможно. Поскольку вся информация об изображении записана в пикселях, новой информации при увеличении их количества взяться просто неоткуда. Таким образом, очень важно, чтобы уже на стадии создания или оцифровки изображение содержало достаточное количество пикселей.

Вычислить необходимое и достаточное количество пикселей в изображении раз и навсегда – невозможно. Строго говоря, это зависит от того, как изображение будет использоваться, – для большого плаката нужно большое изображение, а для картинки в углу веб-сайта размер изображения должен быть небольшим.

Однако уменьшить изображение намного легче, чем увеличить его: в этом случае нужно всего лишь избавиться от лишней информации, а не «придумать» несуществующую и каким-то образом восстановить мелкие детали рисунка. Поэтому, если мы не знаем заранее будущих размеров изображения или не можем их вычислить достаточно точно, следует делать запас в большую сторону: оцифровывать или создавать изображение с максимальным возможным физическим размером, с тем чтобы потом уменьшить его в случае необходимости.

Логический размер изображения, измеряемый в сантиметрах, миллиметрах или других единицах длины, является относительным. Изображение может быть легко увеличено или уменьшено на мониторе, выведено на печать с большим или меньшим размером.

Неприятным фактом является то, что при увеличении или уменьшении логического размера изображения пиксели также увеличиваются и уменьшаются. Увеличивая логический размер изображения, мы нарушаем иллюзию «цельности» картинка, поскольку пиксели становятся заметны.

Как правило, незначительное увеличение изображения проходит практически бесследно. Однако при увеличении в два и более раза пиксели становятся хорошо заметны, что визуально снижает качество изображения. Становится заметен недостаток деталей в изображении, нарушается иллюзия цельности.

Как и в случае с изменением физического размера, уменьшение логического размера – значительно меньшая проблема, хотя изображение все равно пострадает: мелкие детали станут неразличимы.

Величина под названием **«разрешение»** является «связующим звеном» между логическим и физическим размером. Она определяет соотношение между ними и позволяет судить о качестве изображения.

Измеряя разрешение, мы фактически измеряем размер одного пикселя. Поскольку размер этот очень мал (если, конечно, мы пытаемся добиться высокого качества и иллюзии цельного изображения), то не принято измерять его в долях миллиметра. Разрешение измеряют в количестве пикселей на дюйм или на сантиметр, то есть – сколько пикселей нужно выстроить в ряд, чтобы они составили один линейный дюйм (или сантиметр). Обозначается разрешение сокращением ppi (pixel per inch, то есть «пикселей на дюйм») или – для разрешения печатного устройства – dpi (dots per inch, «точек на дюйм»).

15. **Разрешение: типы и единицы измерения.**

Для изображений большое значение имеет понятие **разрешение**, которое выражается количеством точек на единицу длины. Различают разные типы разрешения:

- монитора;

- изображения;
- печати.

Разрешение монитора указывается двумя значениями и определяет количество точек по горизонтали и вертикали. Элементарную точку на экране принято называть пикселем. Таким образом, значение 800 x 600 означает, что видимая область монитора имеет 800 пикселей по горизонтали и 600 — по вертикали. **Пиксель** — это относительная единица измерения, ее величина зависит от установленного экранного разрешения и размера монитора. Например, для 15-дюймового монитора популярным разрешением является 800 x 600 пикселей. Картинка с такими же размерами (800 x 600) будет занимать всю область экрана. Увеличив разрешение до 1024 x 768, вы, тем самым, уменьшите размеры изображения на экране.

Разрешение монитора имеет важное значение для web-дизайнеров, ведь от этого параметра зависит и размер помещаемых на web-страницу изображений. При неудачном учете этого значения рисунок может не поместиться целиком на web-страницу, и тогда появится горизонтальная полоса прокрутки, которая создает неудобства посетителям сайта и может испортить дизайн. Таким образом, для сайта, который ориентирован на разрешение монитора пользователей 800 x 600, ширину изображений следует ограничить размером 700—770 пикселей. Данная величина получается с учетом ширины полосы прокрутки браузера и толщины вертикальных рамок окна.

Количество пикселей, показываемых на единицу длины изображения, характеризует величину **разрешения экранного изображения** и называется пиксель на дюйм (pixel per inch — *ppi*). Если изображение предназначено только для вывода на экран монитора, а не для печати, то графические программы, как правило, устанавливают разрешение 72 *ppi*.

Для любого печатающего устройства характерна величина, выражающая, сколько точек на дюйм он может воспроизвести. Эта величина называется *dpi* (dot per inch — количество точек на дюйм). Для примера, большинство лазерных принтеров поддерживает разрешение 600 *dpi* и выше. Размер точки определяется характеристиками принтера и свойствами изображения.

Поскольку графические редакторы в силу физических различий между пикселем экрана и печатной точкой не в состоянии передать величину точки, то плотность точек заменяется количеством пикселей на экране. Так, для изображения, которое на печати будет иметь размер один на один дюйм, при разрешении 72 *dpi* количество пикселей будет равно 5184 (72 x 72). Увеличение разрешения до 300 *dpi* увеличит и количество пикселей до 90 тыс. (300 x 300). Более высокое разрешение, как правило, передает и большее количество деталей в изображении.

Иногда путают разрешение изображения и печати в связи с тем, что в некоторых графических программах используется только один термин — *dpi*. Поэтому следует учитывать, что если изображение предназначено для вывода на экран монитора, на web-страницу, например, то разрешение в данном случае должно пониматься как **ppi** (пиксель на дюйм).

16. **Связь разрешения и физического размера изображения.**

Любое графическое изображение может быть показано как на экране, так и распечатано на принтере. По этой причине графические редакторы поддерживают два взаимосвязанных размера изображения — один измеряется в пикселях и предназначен для вывода на экран, а другой — измеряется в сантиметрах, миллиметрах, дюймах и других типографских единицах и показывает, какой ширины и высоты будет напечатано изображение.

Изменение разрешения изображения никак не сказывается на ширине и высоте напечатанного изображения, но существенно влияет на качество печати, хотя увидеть конечный результат на экране невозможно. Вместо этого будет изменяться количество пикселей по горизонтали и вертикали, чтобы сохранить плотность точек на прежнем

уровне. Увеличивая разрешение в два раза, вы тем самым увеличиваете количество пикселей в 4 раза, линейные размеры изображения удваиваются по ширине и высоте.

При резком уменьшении разрешения исчезают мелкие детали, и возникает паразитный узор, называемый **муаром**.

Механизм возникновения муара состоит во взаимодействии двух сеток, разрешение которых близко друг другу. Периодическая структура изображения (минимальные периодические линии оригинала) лежит в граничной зоне (близка разрешению) дискретизации. Муар – это проблема, сопровождающая процесс растеризации. Он может появиться и при операции изменения размеров изображения.

17. **Связь между параметрами изображения и размерами файла.**

Размер файла с пиксельным изображением зависит от разрешения пиксельного изображения и от цветовой модели. Он имеет прямо пропорциональную зависимость от произведения числа пикселей по горизонтали и вертикали на глубину цвета использованной цветовой модели.

Размеры пиксельных изображений, предназначенных для вывода на печать, могут достигать величин сотен и тысяч мегабайт. При работе с такими изображениями неизбежно замедление даже на самых мощных компьютерах. Поэтому размеры пиксельного изображения в пикселах и его разрешение должны точно соответствовать назначению этого изображения.

Средствами растровой графики принято иллюстрировать работы, требующие высокой точности в передаче цветов и полутонов. Однако размеры файлов растровых иллюстраций стремительно растут с увеличением разрешения. Фотоснимок, предназначенный для домашнего просмотра (стандартный размер 10x15 см, оцифрованный с разрешением 200-300 dpi, цветное разрешение 24 бита), занимает в формате TIFF с включенным режимом сжатия около 4 Мбайт. Оцифрованный с высоким разрешением слайд занимает 45-50 Мбайт. Цветоделенное цветное изображение формата A4 занимает 120-150 Мбайт.

18. **Формат графического файла. Типы форматов.**

Знание файловых форматов и их возможностей является одним из ключевых факторов в допечатной подготовке изданий, в подготовке изображений для web-страниц и в компьютерной графике вообще. Все форматы имеют какие-то характерные особенности и возможности, делающие их незаменимыми в работе. Формат файла определяется по его расширению. Поэтому в большинстве случаев обозначение формата и расширение совпадают.

Существует несколько различных типов графических форматов, каждый из которых сохраняет данные определенным способом. В настоящее время наиболее широко используются растровый, векторный и метафайловый форматы. Существуют, однако, и другие типы форматов — форматы сцены, анимации, мультимедиа, гибридные, гипертекстовые, гипермедиа, объемные, язык моделирования виртуальной реальности, аудиоформаты, форматы шрифтов, язык описания страницы.

Растровые форматы используются для хранения растровых данных. Файлы этого типа особенно хорошо подходят для хранения реальных изображений, например фотографий и видеоизображений. Растровые файлы, по сути дела, содержат точную пиксельную карту изображения. Программа визуализации реконструирует это изображение на отображающей поверхности устройства вывода.

TIF. При сохранении иллюстрации в этом формате не используется ни один из видов компрессии (сжатия). В этом формате получают максимально возможную степень качества и соответствия, сохраненной в файле копии изображения. Это единственный формат, используемый в профессиональном дизайне для хранения изображений высокого качества. Качественные TIF-изображения могут занимать несколько сотен мегабайт. TIF-формат является лучшим выбором при передаче изображений и растровой графики в векторные программы и издательские системы.

.JPG. Этот формат используется для сжатия изображения в десятки раз. Формат позволяет использовать различные степени сжатия, делая тем самым выбор либо в сторону увеличения качества, либо в сторону уменьшения файла. В профессиональной полиграфии этот формат не используется из-за существенных потерь качества изображения. Для просмотра изображения на экране монитора или для распечатки на принтере качества JPG-формата достаточно. В формате JPG используется метод сжатия jpeg. Этим методом лучше сжимаются растровые изображения фотографического качества и плохо сжимаются логотипы или схемы. В этом формате хорошо и с меньшими потерями сжимаются большие изображения с высоким разрешением 200-300 ppi и плохо сжимаются с низким разрешением 72-150 ppi. Нежелательно сохранять изображения в JPG-формате, где важны все тонкости цветопередачи, так как во время сжатия происходит отбрасывание некоторой цветовой информации. В этом формате следует сохранять только конечный вариант работы, потому что любое пересохранение приводит к новым потерям данных и превращениям изображения в кашу.

.GIF. Этот формат имеет метод сжатия, который обозначается LZW. Этот формат имеет ограниченную палитру цветов. Основное ограничение GIF состоит в том, что цветное изображение может иметь не больше 256 цветов, поэтому цвета в этом формате становятся грубыми, а само изображение зернистым. Не используется в полиграфии и не рекомендуется для изображений, предназначенных для монитора или принтера. В GIF-формате пиксели изображения записываются через строку. По этой технологии, получив только часть файла уже можно увидеть изображение целиком, но с низким качеством. В случае с контрастностью изображения с четкими границами между цветами или в случае с однотонным изображением при использовании этого формата большая степень сжатия, чем JPG, причем качество не изменяется. В GIF можно оставить один-два цвета прозрачными, и они станут невидимыми в программах-браузерах просматриваемых web-страниц. Прозрачность обеспечивается за счет дополнительного альфа-канала в изображении, которое сохраняется вместе с файлом. Кроме того, этот файловый формат может содержать не одну, а несколько растровых картинок, которые Интернет-браузеры могут подгружать одну за другой с указанной в файле частотой. С помощью нескольких картинок создается иллюзия движения, называемая GIF-анимацией. GIF-формат используется для создания web-страниц: баннеров (рекламных заставок), элементов фона.

.PNG. Это формат, разработанный относительно недавно, предназначенный для того, чтобы заменить GIF-формат. В нем используется метод сжатия без потерь качества, который обозначается deflate. Сжатые индексированные файлы (с небольшим количеством цветов) имеют меньший размер по сравнению с аналогичными GIF-файлами. Глубина цвета в файлах может быть любой до 48 бит. В отличие от GIF-формата PNG поддерживает не только прозрачность, но и полупрозрачность. В файловом формате PNG записана информация о гаммах коррекции. Гамма представляет собой некоторое число, характеризующее зависимость яркости свечения экрана монитора от напряжения на электродах. Это число считывается из файла, позволяющего ввести поправку яркости при отображении. Требуется оно для того, чтобы картинки, созданные в ОС Macintosh выглядели одинаково в других ОС. Эта особенность позволяет добиться одинакового отображения информации независимо от аппаратуры пользователя.

.EPS. Это самый удобный и универсальный способ хранения графических данных. Предназначен для передачи векторных и растровых изображений в издательские системы. Создается всеми программами, работающими с графикой. Этот формат используется только тогда, когда печать осуществляется на устройстве, поддерживающем язык PostScript. В формате EPS сохраняются данные в буфере обмена у всех графических программ фирмы Adobe. Вместе с EPS-файлами можно сохранять эскизы изображений. Эскиз – это копия с низким разрешением, которая сохраняется вместе с файлом EPS и позволяет увидеть, что находится внутри изображения. Открыть EPS-файл для

редактирования могут только программы фирмы Adobe – Photoshop, Illustrator. Остальные графические программы могут открывать только в режиме просмотра.

.PDF. Это независимый от графических программ формат для создания электронной документации, презентаций, а также для передачи графики через сети. PDF-файла создаются путем конвертирования из PostScript-файла или функцией экспорта. Программы Photoshop, Illustrator могут создавать только одностраничный файл PDF. Все данные в формате PDF могут сжиматься. Причем к разного типа информации применяются разные типы сжатия. Файл PDF может быть оптимизирован – из него удаляются повторяющиеся элементы, устанавливается постраничный порядок загрузки страниц с приоритетом сначала для текста, потом для графики. Формат PDF используется для передачи по сетям в компактном виде графики и текста. Особенностью многостраничных файлов является то, что они могут содержать элементы, обеспечивающих поиск и просмотр электронных документов, а также могут содержать гипертекстовые ссылки и электронное оглавление. Наиболее удобным средством для работы с PDF-файлами является программа Acrobat. Причем есть 2 варианта этой программы: Acrobat Professional (для создания многостраничных файлов) и Acrobat Reader (для просмотра PDF-файлов).

.PSD. Это внутренний формат программы Photoshop. Стал поддерживаться все большим количеством графических программ. Этот формат позволяет записывать изображение с многими слоями и дополнительными альфа-каналами, а также с каналами простых цветов и контурами, и другой специфической информацией.

.BMP. Растровый формат, который является родным графическим форматом Windows. Поддерживается всеми редакторами. В этом формате хранятся небольшие растровые изображения, предназначенные для использования в системе Windows. Это формат невысокого качества и с низкой степенью сжатия. Его не рекомендуется использовать не для web-дизайна, не для передачи.

.PCX. Этот формат является самым известным. Практически любая программа, работающая с графикой, поддерживает этот формат. Формат PCX поддерживает метод сжатия, который обозначается RLE. Этот формат используется для штрихованных изображений и для изображений с небольшой глубиной цвета.

Файлы **векторного формата** особенно полезны для хранения линейных элементов (линий и многоугольников), а также элементов, которые можно разложить на простые геометрические объекты (например, текст). Векторные файлы содержат не пиксельные значения, а математические описания элементов изображений. По математическим описаниям графических форм (линий, кривых, сплайнов) программа визуализации строит изображение.

Векторные файлы структурно более просты, чем большинство растровых файлов, и обычно организованы в виде потоков данных.

.WMF. Это векторный формат, который используется графическими программами ОС Windows. Этот формат служит для передачи векторных изображений через буфер обмена в среде Windows. Этот формат принимается практически всеми программами, работающими с векторной графикой. Использовать этот формат для растровых изображений нельзя. Недостатки: искажение цвета и несохранение ряда параметров, которые устанавливаются для изображений в графических программах.

.AI. Внутренний формат программы Illustrator. Может открываться программой Photoshop и, кроме того, этот формат поддерживают все программы, связанные с векторной графикой. Этот формат является лучшим средством при передаче векторных изображений из одной программы в другую. Растровые графические элементы при передаче через AI-формат в большинстве случаев теряются.

.CDR. Это внутренний формат программы Corel Draw. Этот формат имеет большую популярность, как и сам пакет программ. Многие программы могут импортировать векторные файлы в форматы Corel Draw. В формате CDR содержатся и

растровые графические объекты. В этом формате применяется компрессия, причем для векторных и растровых файлов применяется разная компрессия.

Метафайлы могут хранить и растровые, и векторные данные. Простейшие метафайлы напоминают файлы векторного формата; они содержат язык или синтаксис для определения элементов векторных данных, но могут включать и растровое представление изображения. Метафайлы часто используются для транспортировки растровых и векторных данных между аппаратными платформами, а также для перемещения изображений между программными платформами.

19. **Векторные, растровые и универсальные форматы.**

«Смотри №19»

20. **Алгоритмы сжатия графической информации. Их классификация.**

Цель сжатия данных - обеспечить компактное представление данных, вырабатываемых источником, для их более экономного сохранения и передачи по каналам связи.

Сжатие информации является одним из способов ее кодирования. Вообще коды делятся на три большие группы - коды сжатия (эффективные коды), помехоустойчивые коды и криптографические коды. Коды, предназначенные для сжатия информации, делятся, в свою очередь, на коды без потерь и коды с потерями. Кодирование без потерь подразумевает абсолютно точное восстановление данных после декодирования и может применяться для сжатия любой информации. Кодирование с потерями имеет обычно гораздо более высокую степень сжатия, чем кодирование без потерь, но допускает некоторые отклонения декодированных данных от исходных.

Все методы сжатия информации можно условно разделить на два больших непересекающихся класса: сжатие с **потерей** информации и сжатие **без потери** информации.

Сжатие без потери информации.

Эти методы сжатия нас интересуют в первую очередь, поскольку именно их применяют при передаче текстовых документов и программ, при выдаче выполненной работы заказчику или при создании резервных копий информации, хранящейся на компьютере.

Методы сжатия этого класса не могут допустить утрату информации, поэтому они основаны только на устранении ее избыточности, а информация имеет избыточность почти всегда (правда, если до этого кто-то ее уже не уплотнил). Если бы избыточности не было, нечего было бы и сжимать.

Вот простой пример. В русском языке 33 буквы, десять цифр и еще примерно полтора десятка знаков препинания и прочих специальных символов. Для текста, который записан только прописными русскими буквами вполне хватило бы шестидесяти разных значений. Тем не менее, каждый символ обычно кодируется байтом, который содержит 8 битов и может выражать 256 различных кодов. Это первое основание для избыточности.

Программы, выполняющие сжатие информации, могут вводить свою кодировку (разную для разных файлов) и приписывать к сжатому файлу некую таблицу (словарь), из которой распаковывающая программа узнает, как в данном файле закодированы те или иные символы или их группы. Алгоритмы, основанные на перекодировании информации, называются **алгоритмами Хафмана**.

Наличие повторяющихся фрагментов — третье основание для избыточности. В текстах это встречается редко, но в таблицах и в графике повторение кодов — обычное явление. Так, например, если число 0 повторяется двадцать раз подряд, то нет смысла ставить двадцать нулевых байтов. Вместо них ставят один ноль и коэффициент 20.

Такие алгоритмы, основанные на выявлении повторов, называются **методами RLE (Run Length Encoding)**.

Большими повторяющимися последовательностями одинаковых байтов особенно отличаются графические иллюстрации, но не фотографические (там много шумов и

соседние точки существенно различаются по параметрам), а такие, которые художники рисуют «гладким» цветом, как в мультипликационных фильмах.

Сжатие с потерей информации.

Сжатие с потерей информации означает, что после распаковки уплотненного архива мы получим документ, который несколько отличается от того, который был в самом начале. Понятно, что чем больше степень сжатия, тем больше величина потери и наоборот. Разумеется, такие алгоритмы неприменимы для текстовых документов, таблиц баз данных и особенно для программ. Незначительные искажения в простом неформатированном тексте еще как-то можно пережить, но искажение хотя бы одного бита в программе сделает ее абсолютно неработоспособной.

В то же время, существуют материалы, в которых стоит пожертвовать несколькими процентами информации, чтобы получить сжатие в десятки раз. К ним относятся фотографические иллюстрации, видеоматериалы и музыкальные композиции. Потеря информации при сжатии и последующей распаковке в таких материалах воспринимается как появление некоторого дополнительного «шума». Но поскольку при создании этих материалов определенный «шум» все равно присутствует, его небольшое увеличение не всегда выглядит критичным, а выигрыш в размерах файлов дает огромный (в 10-15 раз на музыке, в 20-30 раз на фото- и видеоматериалах).

К алгоритмам сжатия графических файлов с потерей информации относится такой известный алгоритм как JPEG. Алгоритм JPEG используется при сжатии фотоизображений. Графические файлы, сжатые этим методом, имеют расширение JPG.

Алгоритмы сжатия с потерей информации применяют только для потребительских задач. Это значит, например, что если фотографии передаются для просмотра, то подобные алгоритмы применять можно. Если же они передаются для дальнейшей обработки, например для редактирования, то никакая потеря информации в исходном материале недопустима.

Величиной допустимой потери при сжатии обычно можно управлять. Это позволяет экспериментировать и добиваться оптимального соотношения размер/качество. На фотографических иллюстрациях, предназначенных для воспроизведения на экране, потеря 5% информации обычно не критична, а в некоторых случаях можно допустить и 20-25%.

Алгоритмы сжатия без потери информации.

RLE

Групповое кодирование - Run Length Encoding (RLE) - один из самых старых и самых простых алгоритмов архивации. К положительным сторонам алгоритма, можно отнести то, что он не требует дополнительной памяти при работе, и быстро выполняется. Алгоритм применяется в форматах PCX, TIFF, BMP. Интересная особенность группового кодирования в PCX заключается в том, что степень архивации для некоторых изображений может быть существенно повышена всего лишь за счет изменения порядка цветов в палитре изображения.

LZW

LZW-код (Lempel-Ziv & Welch) является на сегодняшний день одним из самых распространенных кодов сжатия без потерь. Именно с помощью LZW-кода осуществляется сжатие в таких графических форматах, как TIFF и GIF, с помощью модификаций LZW осуществляют свои функции очень многие универсальные архиваторы. Работа алгоритма основана на поиске во входном файле повторяющихся последовательностей символов, которые кодируются комбинациями длиной от 8 до 12 бит. Таким образом, наибольшую эффективность данный алгоритм имеет на текстовых файлах и на графических файлах, в которых имеются большие одноцветные участки или повторяющиеся последовательности пикселей.

Отсутствие потерь информации при LZW-кодировании обусловило широкое распространение основанного на нем формата TIFF. Этот формат не накладывает каких-

либо ограничений на размер и глубину цвета изображения и широко распространен, например, в полиграфии. Другой основанный на LZW формат - GIF - более примитивен - он позволяет хранить изображения с глубиной цвета не более 8 бит/пиксел.

Алгоритмы сжатия с потерей информации.

JPEG

Алгоритм JPEG был разработан группой фирм под названием Joint Photographic Experts Group. Целью проекта являлось создание высокоэффективного стандарта сжатия как черно-белых, так и цветных изображений, эта цель и была достигнута разработчиками. В настоящее время JPEG находит широчайшее применение там, где требуется высокая степень сжатия - например, в Internet.

В отличие от LZW-алгоритма JPEG-кодирование является кодированием с потерями. Сам алгоритм кодирования базируется на очень сложной математике, но в общих чертах его можно описать так: изображение разбивается на квадраты 8*8 пикселов, а затем каждый квадрат преобразуется в последовательную цепочку из 64 пикселов. Далее каждая такая цепочка подвергается так называемому DCT-преобразованию, являющемуся одной из разновидностей дискретного преобразования Фурье. Оно заключается в том, что входную последовательность пикселов можно представить в виде суммы синусоидальных и косинусоидальных составляющих с кратными частотами (так называемых гармоник). В этом случае нам необходимо знать лишь амплитуды этих составляющих для того, чтобы восстановить входную последовательность с достаточной степенью точности. Чем большее количество гармонических составляющих нам известно, тем меньше будет расхождение между оригиналом и сжатым изображением.

Большинство JPEG-кодеров позволяют регулировать степень сжатия. Достигается это очень простым путем: чем выше степень сжатия установлена, тем меньшим количеством гармоник будет представлен каждый 64-пиксельный блок.

Безусловно, сильной стороной данного вида кодирования является большой коэффициент сжатия при сохранении исходной цветовой глубины. Именно это свойство обусловило его широкое применение в Internet, где уменьшение размера файлов имеет первостепенное значение, в мультимедийных энциклопедиях, где требуется хранение возможно большего количества графики в ограниченном объеме.

Отрицательным свойством этого формата является неустранимое никакими средствами, внутренне ему присущее ухудшение качества изображения. Именно этот печальный факт не позволяет применять его в полиграфии, где качество ставится во главу угла.

Однако формат JPEG не является пределом совершенства в стремлении уменьшить размер конечного файла. В последнее время ведутся интенсивные исследования в области так называемого вейвлет-преобразования (или всплеск-преобразования). Основанные на сложнейших математических принципах вейвлет-кодеры позволяют получить большее сжатие, чем JPEG, при меньших потерях информации. Несмотря на сложность математики вейвлет-преобразования, в программной реализации оно проще, чем JPEG. Хотя алгоритмы вейвлет-сжатия пока находятся в начальной стадии развития, им уготовано большое будущее.

Фрактальное сжатие.

Фрактальное сжатие изображений — это алгоритм сжатия изображений с потерями, основанный на применении систем итерируемых функций (IFS, как правило являющимися аффинными преобразованиями) к изображениям.

Данный алгоритм известен тем, что в некоторых случаях позволяет получить очень высокие коэффициенты сжатия для реальных фотографий природных объектов, что недоступно для других алгоритмов сжатия изображений в принципе. Из-за сложной ситуации с патентованием широкого распространения алгоритм не получил.

Основа метода фрактального кодирования — это обнаружение самоподобных участков в изображении. В соответствии с данным методом изображение разбивается на

множество неперекрывающихся ранговых подизображений (range subimages) и определяется множество перекрывающихся доменных подизображений (domain subimages). Для каждого рангового блока алгоритм кодирования находит наиболее подходящий доменный блок и аффинное преобразование, которое переводит этот доменный блок в данный ранговый блок. Структура изображения отображается в систему ранговых блоков, доменных блоков и преобразований.

Для фрактального алгоритма компрессии, как и для других алгоритмов сжатия с потерями, очень важны механизмы, с помощью которых можно будет регулировать степень сжатия и степень потерь. К настоящему времени разработан достаточно большой набор таких методов.

21. Кодирование изображения. Понятие глубины цвета.

В разных отраслях науки, культуры и техники разработаны специальные формы для записи информации.

Код — это группа обозначений, которую можно использовать для отображения информации. Процесс преобразования сообщения в комбинацию символов в соответствии с кодом называется **кодированием**.

Существует три основных способа кодирования информации:

1. Числовой способ — с помощью чисел.
2. Символьный способ — информация кодируется с помощью символов того же алфавита, что и исходный текст.
3. Графический способ — информация кодируется с помощью рисунков или значков.

Почти все созданные и обработанные изображения, хранящиеся в компьютере, можно поделить на две группы:

- растровая графика;
- векторная графика.

Код пикселя содержит информацию о его цвете. Например, два цвета (чёрный — 0, белый — 1) выглядят так в виде двоичного кода.

```

1111111111111111000000000111111111111111
111111111100001111111111000011111111111
111111110011111111111111111001111111111
111111001111111111111111111110011111111
111100111111111111111111111111001111111
110011111111111111111111111111100111111
1011111111111111111111111111111111101
  
```

Для кодирования не цветных изображений обычно используют 256 оттенков серого, начиная от белого, заканчивая чёрным. Для кодирования всех цветов надо 8 битов (1 байт).

Для кодирования цветных изображений обычно используют три цвета: красный, зелёный и синий. Цветной тон получается при смешивании этих трёх цветов.

Красный	Зеленый	Синий	Название	Цвет
0	0	0	Черный	
0	1	0	Зеленый	
0	0	1	Синий	
1	0	0	Красный	
0	1	1	Бирюзовый	
1	1	0	Желтый	
1	0	1	Малиновый	
1	1	1	Белый	

Размер изображения можно посчитать, умножив его ширину на длину в пикселях. Например, изображение размером 200x100 пикселей, занимает 60000 байт.

Глубина цвета — термин компьютерной графики, означающий количество бит (объём памяти), используемое для хранения и представления цвета при кодировании, либо одного пикселя растровой графики или видеоизображения, либо для каждого цвета составляющего один пиксель. Для стандартов потребительского видео битовая глубина определяет количество бит, используемых для каждого цветового компонента.

Монохромные изображения кодируются с помощью одномерной шкалы яркости. Обычно это набор из чёрного и белого цвета и промежуточных оттенков серого, но могут использоваться и другие комбинации: например, монохромные мониторы часто используют зелёный или оранжевый цвет свечения вместо белого. Изображение кодируется с помощью дискретного набора цветов, каждый из которых описан с помощью палитры независимо друг от друга. С увеличением количества бит в представлении цвета количество отображаемых цветов стало непрактично большим для цветовых палитр. При большой глубине цвета на практике кодируют яркости красной, зелёной и синей составляющих — такое кодирование называют RGB-моделью.

8-битный «реальный» цвет.

Сильно ограниченная, однако «реальная» цветовая схема, в которой по три бита (по восемь возможных значений) для красной (R) и зелёной (G) составляющих, и два оставшихся бита на пиксель для кодирования синей (B) составляющей (четыре возможных значения), позволяют представить 256 ($8 \times 8 \times 4$) различных цвета. Нормальный человеческий глаз менее чувствителен к синей составляющей, чем к красной и зелёной, поэтому синяя составляющая представляется одним битом меньше. Такая схема использовалась в MSX2-серии компьютеров в 1990-х годах.

12-битный «реальный» цвет кодируется 4 битами (по 16 возможных значений) для каждой из R, G и B-составляющих, что позволяет представить 4096 ($16 \times 16 \times 16$) различных цветов. Такая глубина цвета иногда используется в простых устройствах с цветными дисплеями (например, в мобильных телефонах).

HighColor

HighColor или HiColor разработан для представления оттенков «реальной жизни», то есть наиболее удобно воспринимаемый человеческим глазом. Такой цвет кодируется 15 или 16 битами:

- 15-битный цвет использует 5 бит для представления красной составляющей, 5 для зелёной и 5 для синей, то есть $2^5 = 32$ возможных значения каждого цвета, которые дают 32 768 ($32 \times 32 \times 32$) возможных цвета.

- 16-битный цвет использует 5 бит для представления красной составляющей, 5 для синей, но 6 бит ($2^6 = 64$ возможных значения) для представления зелёной, так как человеческий глаз более чувствителен к зелёной составляющей. Таким образом получаются 65 536 ($32 \times 64 \times 32$) цветов. 16-битный цвет упоминается как «тысячи цветов» (*thousands of colors*) в системах Macintosh.

- Большинство современных ЖК-дисплеев отображают 18-битный цвет.

TrueColor

TrueColor приближен к цветам «реального мира», предоставляя 16,7 млн различных цветов. Такой цвет наиболее приятен для восприятия человеческим глазом различных фотографий, для обработки изображений.

Deep Color

В конце 1990-х годов некоторые high-end графические системы, например SGI, начали использовать более 8 бит на канал - например, 12 или 16 бит. Программы профессионального редактирования изображений стали сохранять по 16 бит на канал,

предоставляя «защиту» от накапливания ошибок округления, погрешностей при вычислении в условиях ограниченной разрядной сетки чисел.

Для дальнейшего расширения динамического диапазона изображений были созданы различные модели. Например *High Dynamic Range Imaging (HDR)*, использует числа с плавающей запятой и позволяет наиболее точно описывать в изображениях интенсивный свет и глубокие тени в одном и том же цветовом пространстве. Различные модели описывают такие диапазоны, применяя более 32 бит на канал.

22. Цветовые палитры, их виды.

Цветовая палитра (палитра цвета) — фиксированный набор (диапазон) цветов и оттенков, имеющий физическую или цифровую реализацию в том или ином виде (например, атлас цветов, системная цветовая палитра, Московская цветовая палитра).

Существуют три основные палитры цветов[^]

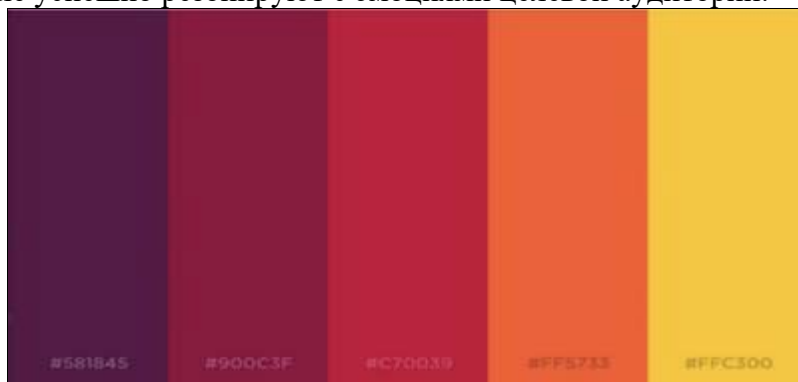
1. Самая известная и популярная — RAL. Впервые стандарт RAL был представлен в 1927 году Немецким Институтом Гарантий Качества и Сертификации (Райхс-Аусшус фюр Лифербедингунген — RAL) по просьбе производителей лакокрасочной продукции. Институт установил стандарт на цветовое пространство, разделив его на диапазоны и обозначив каждый цвет однозначным цифровым индексом. Номера четырёхзначные, (№ XXXX) где 1xxx — жёлтые (27 шт), 2xxx — оранжевые (12 шт), 3xxx — красные (22 шт), 4xxx — фиолетовые (10 шт), 5xxx — синие (23 шт), 6xxx — зеленые (32 шт), 7xxx — серые (37 шт), 8xxx — коричневые (19 шт), 9xxx — светлые и тёмные (12шт). Для определения цвета по системе RAL издаются вееры, каталоги и программное обеспечение. Всего содержится более двух тысяч оттенков по RAL.

2. Более современная и быстрорастущая NCS (англ. Natural Color System, естественная система цвета). Эта цветовая модель была предложена Скандинавским институтом цвета (Skandinaviska Färginstitutet AB), Стокгольм, Швеция в 1979 году. Она основана на системе противоположных цветов и нашла широкое применение в промышленности для описания цвета продукции. При описании цвета по NCS используются шесть простых цветов: белый, чёрный, красный, жёлтый, зелёный и голубой (то есть таких, которые нельзя описать сочетанием двух других, кроме зелёного, его можно получить сочетанием жёлтого и синего). Все остальные цвета представлены сочетанием основных (например, оранжевый — одновременно красноватый и желтоватый). Это облегчает интуитивное понимание цвета из его кодированной записи, в то время как в таких системах как RGB мысленная визуализация цвета по трём цифрам довольно сложна. В описании цвета учитывается близость к черному — темнота цвета, чистота цвета (насыщенность) и процентное соотношение между двумя основными цветами. Полная запись цвета может также включать кодовую букву, обозначающую версию стандарта NCS. Для определения цвета по системе NCS издаются каталоги и программное обеспечение. Последняя редакция цветового веера содержит 1950 цветов.

3. Pantone. Используется в основном в полиграфии. Разработанная американской фирмой Pantone Inc в середине XX века. Использует цифровую идентификацию цветов изображения для полиграфии печати как смесевыми, так и триадными красками. Эталонные пронумерованные цвета напечатаны в специальной книге, страницы которой веерообразно раскладываются. Существует множество каталогов образцов цветов Pantone, каждый из которых рассчитан на определённые условия печати. Например, для печати на мелованной, немелованной бумаге, каталог для металлизированных красок (золотая, серебряная) и т. д. Производитель настаивает на том, что «веера» необходимо ежегодно заменять, так как за это время процесс выцветания и истирания изображения делает цвета неточными.

Смешивание цветов в поиске нужных комбинаций является интересным и увлекательным занятием, особенно для дизайнеров. С помощью цветовой палитры они

добиваются нужных красочных решений при проектировании UI или логотипов. В умелых руках таким образом создаются визуально привлекательные фирменные стили, которые успешно резонируют с эмоциями целевой аудитории.



Дизайнер оперирует богатым арсеналом цветовых схем, при использовании которых очень важно добиваться гармонии. В этом как раз и помогают цветовые палитры, позволяя получать согласованный и эстетически привлекательный дизайн. Придуманы они были не одно тысячелетие назад, но в современной цифровой эпохе получили еще большую популярность. Цветовые палитры представляются в форме HEX-кодов, с помощью которых мы сообщаем компьютеру о том, какой цвет нам нужен.

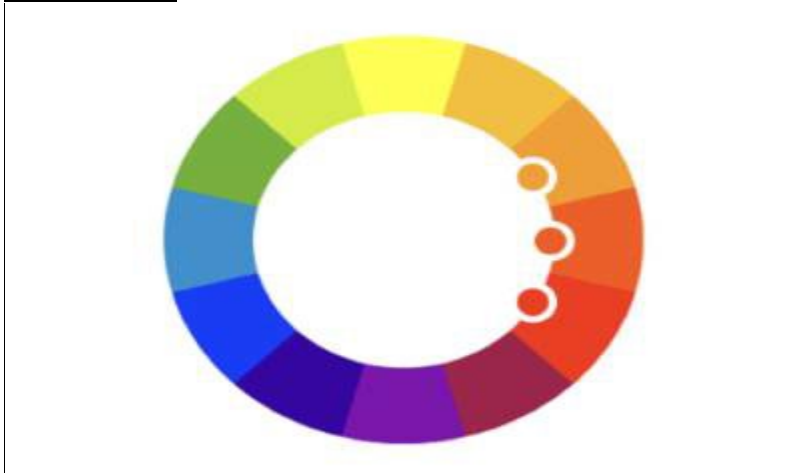
Виды цветовых палитр

Монохроматическая

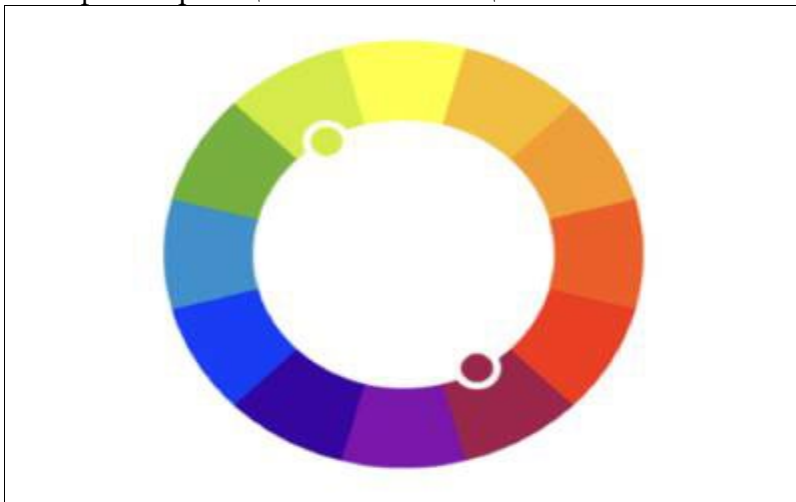


Подразумевает использование одного цвета, его тональностей и теней. Монохроматическая цветовая гармония всегда оказывается предпочтительным выбором. За счет уникального контраста оттенков она делает дизайн более привлекательным.

Аналоговая



Аналоговые палитры основываются на цветах, располагающихся рядом на цветовом круге. Такие палитры используются в случаях, когда нужно создать дизайн без контрастов. В оформлении сайтов это означает соблюдение гармонии цвета за счет заполнения фона страницы аналоговыми цветами.



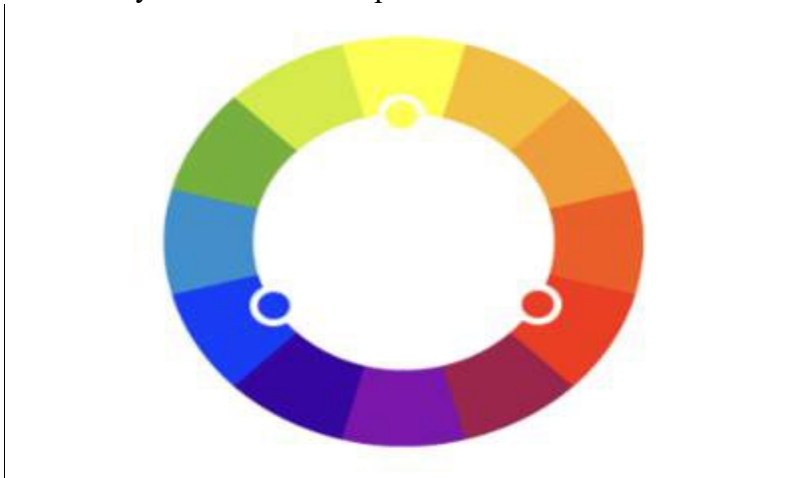
Комплементарная

Комплементарная палитра строится на основе двух противоположных цветов круга и используется для создания эффекта контраста.



Сплит-

В основе сплит-комплементарной схемы лежит та же контрастная пара цветов, только в этом случае один из них разбивается на два соседних.



Триадическая

Триадическая цветовая схема строится из трех равноудаленных друг от друга цветов круга. С помощью такой палитры удастся одновременно добиться и контраста, и гармонии.



Прямоугольная

Прямоугольная палитра включает две пары комплементарных цветов. Наиболее опытные дизайнеры используют ее для создания завораживающих изображений.

В современном цифровом мире существует широкий спектр инструментов для создания цветowych палитр. Можно выбирать цвета из предлагаемого этими инструментами круга и автоматически генерировать HEX-коды для дальнейшего использования в проектах. Существуют также инструменты, генерирующие цветowe палитры на основе загружаемых изображений. Такой способ создания нужных палитр очень эффективен и существенно сокращает время разработки дизайна.

К известным инструментам работы с палитрами относятся:

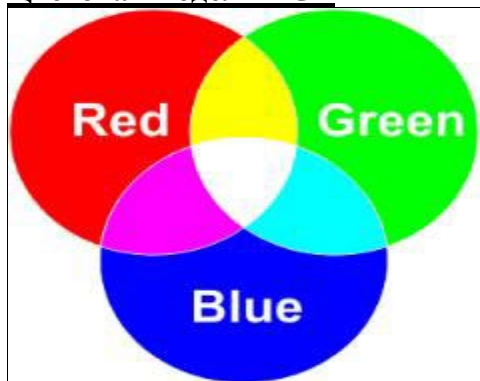
- [Adobe Color](#);
- [Colors](#);
- [Canva Color Wheel](#);
- [Paletton](#);
- Color Hunt;

23. Понятие цветовой модели. Типы цветowych моделей.

Существует немало цветowych моделей, наиболее часто используемые можно разделить на три группы:

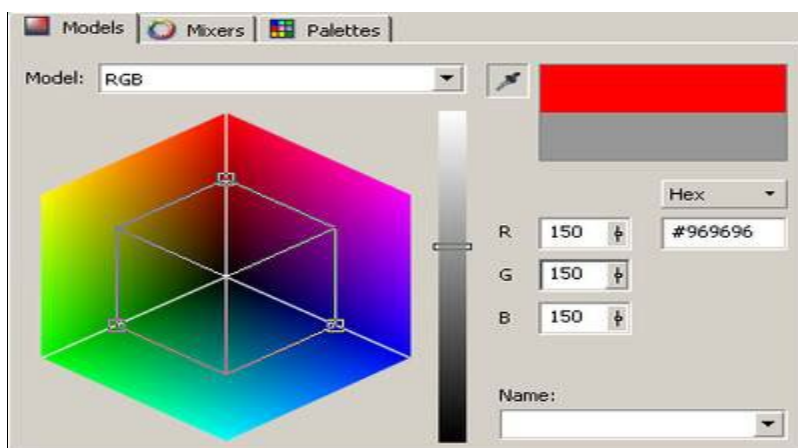
- **аппаратно-зависимые** — цветowe модели данной группы описывают цвет применительно к конкретному, цветowоспроизводящему устройству (например монитору), — *RGB, CMYK*
- **аппаратно-независимые** — эта группа цветowych моделей для того, чтобы дать однозначную информацию о цвете — *XYZ, Lab*
- **психологические** — эти модели основываются на особенностях восприятия человека — *HSB, HSV, HSL*

Цветовая модель RGB



Данная цветовая модель описывает цвет источника света (сюда можно отнести, например экран монитора или телевизора). Из огромного множества цветов, в качестве

основных (первичных) было выделено три цвета: красный (Red), зеленый (Green), синий (Blue). Первые буквы названий основных цветов образовали название цветовой модели RGB.



Когда смешиваются два основных цвета, получившийся цвет осветляется: красный и зеленый дают желтый, зеленый и синий дают голубой, из синего и красного получится пурпурный. Если смешать все три основных цвета, образуется белый. Такие цвета называются аддитивными.

Эту модель можно представить в виде трехмерной системы координат, где каждая отражает значение одного из основных цветов в диапазоне от нуля до максимума. Получился куб, внутри которого находятся все цвета, образующие цветовое пространство RGB.

Важные точки и линии модели RGB

1. Начало координат: в этой точке значения всех основных цветов равны нулю, излучение отсутствует, т. е. это — точка черного цвета.
2. В ближайшей к зрителю точке все составляющие имеют максимальное значение, это значит максимальное свечение — точка белого цвета.
3. На линии, соединяющей эти точки (по диагонали куба), расположены оттенки серого цвета: от черного к белому. Этот диапазон иначе называют серой шкалой (Grayscale).
4. Три вершины куба дают чистые исходные цвета, остальные три отражают двойные смешения исходных цветов.

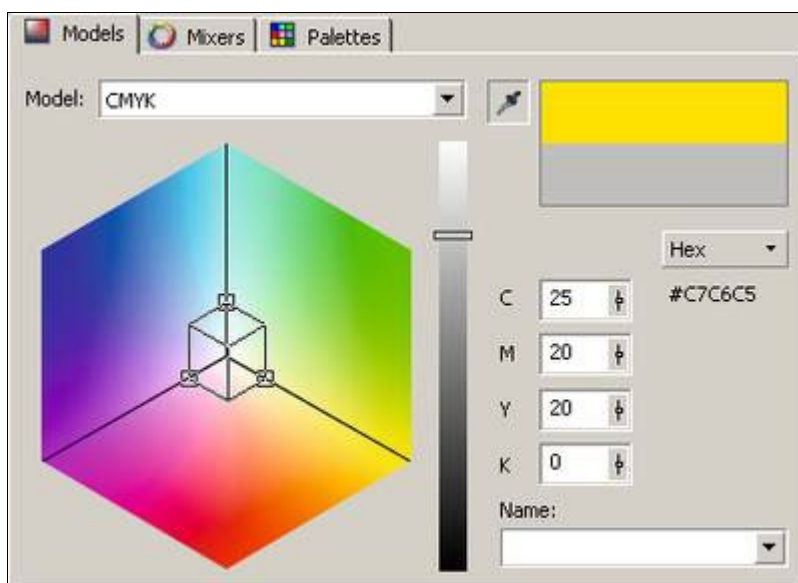
Плюс этой модели состоит в том, что она описывает все 16 миллионов цветов, а минус в том, что при печати часть (самые яркие и насыщенные) этих цветов потеряется. Так как RGB аппаратно-зависимая модель, то одна и та же картинка на разных мониторах может отличаться по цвету, например потому что экраны этих мониторов сделаны по разным технологиям или мониторы по разному настроены.

Цветовая модель CMYK



Если предыдущая модель описывает светящиеся цвета, то CMYK наоборот, для описания цветов отраженных. Еще они называются субтрактивными («вычитательными»),

потому что они остаются после вычитания основных аддитивных. Так как цветов для вычитания у нас три, то и основных субтрактивных цветов тоже будет три: голубой (Cyan), пурпурный (Magenta), желтый (Yellow).



Три основных цвета модели CMYK, называют полиграфической триадой. Печатая этими красками, происходит поглощение красной, зеленой и синей составляющих. В изображении CMYK каждый пиксель имеет значение процентного содержания триадных красок.

Когда смешиваем две субтрактивных краски, то результирующий цвет затемняется, а если смешать три, то должен получиться черный цвет. При нулевом значении всех красок получаем белый цвет. А когда значения всех составляющих равны — получаем серый цвет.

На деле получается, что если смешать три краски при максимальных значениях, вместо глубокого черного цвета у нас получится скорее грязный темно-коричневый. Это происходит потому, что полиграфические краски не идеальны и не могут отразить весь цветовой диапазон.

Что бы компенсировать эту проблему к этой триаде добавили четвертую краску черного цвета, она и добавила последнюю букву в названии цветовой модели **С** — Cyan (Голубой), **М** — Magenta (Пурпурный), **Y** — Yellow (Желтый), **К** — black (Черный).

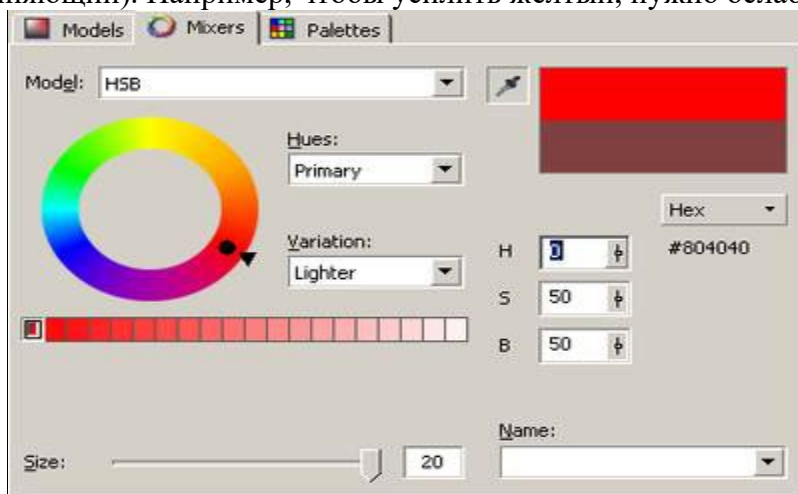
Как и RGB, CMYK тоже модель аппаратно-зависимая. Зависит конечный результат от краски, от типа бумаги, от печатной машины, от особенностей технологии печати. Поэтому одно и то же изображение в разных типографиях может быть напечатанным по-разному.

Цветовая модель HSB



Если вышеописанные модели соединить в одну, то результат можно изобразить в виде цветового круга, где основные цвета моделей RGB и CMY расположены в следующей зависимости: каждый цвет находится напротив комплементарного цвета, его дополняющего и между цветами, с помощью которых он образован.

Чтобы усилить какой-то цвет, нужно ослабить цвет, находящийся напротив (дополняющий). Например, чтобы усилить желтый, нужно ослабить синий.



Для описания цвета в данной модели есть три параметра: **Hue** (оттенок) — показывает положение цвета на цветовом круге и обозначается величиной угла от 0 до 360 градусов, **Saturation** (насыщенность) — определяет чистоту цвета (уменьшение насыщенности похоже на добавление белого цвета в исходный цвет), **Brightness** (яркость) — показывает освещенность или затененность цвета (уменьшение яркости похоже на добавление черной краски). Первые буквы в названии этих параметров и дали название цветовой модели.

Модель HSB хорошо согласуется с человеческим восприятием: цветовой тон — длина волны света, насыщенность — интенсивность волны, а яркость — количество света. Минусом модели HSB является необходимость конвертировать ее в **RGB** для отображения на экране монитора или в **CMYK** для печати.

Цветовая модель Lab



Эту модель создала

Международная комиссия по освещению для того, чтобы уйти от недостатков предыдущих моделей. Было необходимо создать аппаратно-независимую модель для определения цвета независимую от параметров устройства. В модели Lab цвет представлен тремя параметрами:

- **L** — светлота
- **a** — хроматический компонент в диапазоне от зеленого до красного
- **b** — хроматический компонент в диапазоне от синего до желтого

При переводе цвета из какой-нибудь модели в Lab, все цвета сохраняются, так как пространство Lab самое большое. Поэтому данное пространство используют как посредника при конвертации цвета из одной модели в другую.

Цветовая модель Grayscale

Самое простое и понятное пространство используется для отображения черно-белого изображения. Цвет в данной модели описывается всего одним параметром. Значение параметра может быть в градациях (от 0 до 256) или в процентах (от 0% до 100%). Минимальное значение соответствует белому цвету, а максимальное — черному.

Индексные цвета

Вряд ли допечатнику придется работать с индексными цветами, но знать, что это такое, не помешает.

Когда-то давно, на заре компьютерных технологий, компьютеры могли отображать на экране не больше 256 цветов одновременно, а до этого 64 и 16 цветов. Исходя из таких условий был придуман индексный способ кодирования цвета. Каждый цвет, содержащийся в изображении, получил порядковый номер, с помощью этого номера описывался цвет всех пикселей, имеющих соответствующий цвет. Но у разных изображений наборы цветов разные и по этому пришлось в каждой картинке хранить свой набор цветов (набор цветов назвали — цветовая таблица).

Современные компьютеры (даже самые простые) способны отображать на экране 16,8 млн цветов, поэтому нет особой необходимости в использовании индексных цветов. Но с развитием интернета эта модель вновь используется. Все потому, что такой файл может иметь гораздо меньший размер.

24. Понятие цветового режима.



Различные цветовые режимы:

1. Режим RGB (миллионы цветов)
2. Режим CMYK (цвета четырехцветной печати)
3. Режим индексированных цветов (256 цветов)
4. Режим градаций серого (256 оттенков серого)
5. Битовый режим (2 цвета)

Цветовой режим, или режим изображения, определяет, как объединяются цвета на основе количества каналов в цветовой модели. Разные цветовые режимы дают различные уровни детализации цвета и размер файла. Например, используйте цветовой режим CMYK для изображений в полноцветной печатной брошюре, а цветовой режим RGB для изображений, предназначенных для Интернета или электронной почты, чтобы уменьшить размер файла, сохраняя достоверность цветов.

Цветовой режим RGB

Режим RGB в Photoshop использует модель RGB, назначая каждому пикселу значение интенсивности. В изображениях с 8 битами на канал значения интенсивности находятся в диапазоне от 0 (черный) до 255 (белый) для каждого из RGB-компонентов цвета (красный, зеленый, синий). Например, ярко-красный цвет имеет значение R=246, G=20 и B=50. Если значения всех трех компонентов одинаковы, получается затемнение

нейтрально-серого цвета. Если значения всех компонентов равны 255, то получается чистый белый, а если 0, то чистый черный.

Чтобы воспроизвести цвета на экране, в изображениях RGB используются три цвета, или канала. В изображениях, содержащих 8 бит на канал, каждый пиксел содержит 24 бита (3 канала по 8 бит) цветовой информации. В 24-битных изображениях три канала позволяют воспроизводить до 16,7 миллиона цветов на пиксел. В 48-битных (16 бит на канал) и 96-битных (32 бита на канал) изображениях каждый пиксел может воспроизводить еще больше цветов.

Помимо того, что модель RGB является режимом по умолчанию для новых изображений, создаваемых в Photoshop, она еще используется для отображения цветов компьютерными мониторами. Это означает, что при работе в цветовых режимах, отличных от RGB (например, в CMYK), Photoshop конвертирует изображение в RGB для отображения на экране.

Несмотря на то, что RGB является стандартной цветовой моделью, точный диапазон отображаемых цветов может быть разным в зависимости от приложения и устройства вывода.

Режим CMYK

В режиме CMYK пикселу для каждой из триадных красок присваивается значение в процентах. Самым светлым цветам (цветам подсветки) назначается меньшее значение, а более темным (цветам тени) — большее. Например, ярко-красный цвет может состоять из 2 % голубого, 93 % пурпурного, 90 % желтого и 0 % черного. Если в изображениях CMYK все четыре компонента равны 0 %, то получается чистый белый цвет.

Режим CMYK предназначен для подготовки изображения к печати с использованием триадных цветов. В результате преобразования RGB-изображения в CMYK получается *цветоделение*. Если исходное изображение было RGB, его лучше всего отредактировать в режиме RGB и только в самом конце редактирования преобразовать в CMYK. В режиме CMYK можно также работать непосредственно с изображениями CMYK, полученными со сканера или импортированными из профессиональных систем.

Несмотря на то, что CMYK — это стандартная цветовая модель, точный диапазон воспроизводимых цветов может различаться в зависимости от печатной машины и условий печати.

Цветовой режим Lab

Цветовая модель Lab Международной светотехнической комиссии основана на восприятии цвета человеческим глазом. В режиме Lab числовые значения описывают все цвета, которые видит человек с нормальным зрением. Поскольку значения Lab описывают, как выглядит цвет, а не сколько конкретной краски требуется устройству (например, монитору, настольному принтеру или цифровой камере) для воспроизведения цветов, модель Lab считается *аппаратно-независимой* цветовой моделью. Системы управления цветом используют Lab в качестве справочника цветов, чтобы получать предсказуемые результаты при преобразовании цвета из одного цветового пространства в другое.

Изображения Lab можно сохранять в следующих форматах: Photoshop, Photoshop EPS, Large Document Format (PSB), Photoshop PDF, Photoshop Raw, TIFF, Photoshop DCS 1.0 и Photoshop DCS 2.0. 48-битные (16 бит на канал) изображения Lab можно сохранять в форматах Photoshop, Large Document Format (PSB), Photoshop PDF, Photoshop Raw и TIFF.

Режим градаций серого

В режиме градаций серого в изображениях используются различные оттенки серого цвета. В 8-битных изображениях допускается до 256 оттенков серого. Каждый пиксел изображения в градациях серого содержит значение яркости в диапазоне от 0 (черный) до 255 (белый). В 16- и 32-битных изображениях количество оттенков серого значительно больше.

Значения оттенков серого также могут быть выражены в процентах суммарного покрытия черной краской (значение 0 % эквивалентно белому, а 100 % — черному).

Битовый режим

Битовый режим представляет каждый пиксел изображения одним из двух значений (черный или белый). Изображения в этом режиме называются битовыми (1-битными), поскольку на каждый пиксел приходится ровно один бит.

Режим «Дуплекс»

В режиме «Дуплекс» создаются монотонные, дуплексные (двухцветные), триотонные (трехцветные) и тетратонные (четырёхцветные) изображения в градациях серого с использованием от одной до четырех заказных красок.

Режим «Индексированные цвета»

Режим «Индексированные цвета» выдает 8-битные изображения, содержащие не более 256 цветов. При преобразовании в режим индексированных цветов Photoshop строит *таблицу цветов изображения (CLUT)*, в которой хранятся и индексируются цвета, используемые в изображении. Если цвет исходного изображения отсутствует в этой таблице, программа выбирает ближайший из имеющихся цветов или выполняет *дизеринг* для имитации недостающего цвета.

Хотя палитра цветов этого режима ограничена, он позволяет уменьшить размер файла изображения, при этом сохраняя качество изображения, необходимое для мультимедийных презентаций, веб-страниц и т. п. Возможности редактирования в этом режиме ограничены. Если необходимо большое редактирование, следует временно перейти в режим RGB.

Многоканальный режим

Изображения в многоканальном режиме содержат 256 уровней серого для каждого из каналов и могут пригодиться при специализированной печати.

При преобразовании изображений в многоканальный режим могут оказаться полезны следующие сведения:

- Слои не поддерживаются, и поэтому выполняется их сведение.
- Цветовые каналы исходного изображения становятся каналами смесевых цветов.
- При преобразовании изображения CMYK в многоканальный режим создаются голубой, пурпурный, желтый и черный каналы смесевых цветов.
- При преобразовании изображения RGB в многоканальный режим создаются голубой, пурпурный и желтый каналы смесевых цветов.
- Удаление канала из изображения RGB, CMYK или Lab автоматически преобразовывает это изображение в многоканальный режим путем сведения слоев.

4.4. ТИПОВЫЕ ЗАДАНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК.02.02 ОСНОВЫ HTML, CSS И JAVASCRIPT

ТЕСТ «HTML»

1. *Насте требуется выровнять содержимое ячейки по центру, при этом сохранить валидность по XHTML 1.1. Она предложила такой вариант: <td align="center">Содержимое</td>. Выберите верное утверждение:*

Этот вариант нерабочий, но валидный.

Этот вариант нерабочий и невалидный.

Этот вариант рабочий, но невалидный.

Этот вариант рабочий и валидный.

Правильный ответ: Этот вариант рабочий и валидный.

Примечание: Стандарт XHTML 1.1 позволяет использовать атрибут align в теге <td>. Браузер этот вариант так же корректно обрабатывает.

2. Какой тег является дочерним для <!DOCTYPE>:

Никакой.

<body>

<head>

<html>

Правильный ответ: Никакой.

Примечание: У <!DOCTYPE> нет никаких дочерних элементов, поскольку он находится на одном уровне с тегом <html>.

3. Какая ошибка в следующем коде: <i>Страница 1</i>

Не указан обязательный атрибут alt у тега <a>.

Внутри тега <a> не может быть тег и/или тег <i>.

Не указан обязательный атрибут title у тега <a>.

Не закрыт тег .

Правильный ответ: Не закрыт тег .

Примечание: Любой парный тег (а тег парный) должен иметь закрывающий.

4. Какой вариант написан с ошибкой?

<ol type="N">

<ol type="A">

<ol type="I">

<ol type="i">

Правильный ответ: <ol type="N">

Примечание: Значение N у атрибута type тега не существует.

5. Петру надо сделать нумерованный список, какой из тегов ему надо использовать:

<list>

<tr>

Правильный ответ:

Примечание: Нумерованный список создаётся с помощью тега (с англ. Order List).

6. Какой из представленных ниже HTML-кодов НЕ является валидным XHTML 1.1:

<u>Подчёркнутый текст</u>

<p style="font-size: 1000%;">Текст</p>

<p>Текст</p>

<i>Курсив</i>

Правильный ответ: <u>Подчёркнутый текст</u>

Примечание: В стандарте XHTML 1.1 нет тега <u>, поскольку он признан "путающимся" со ссылками.

7. Какой тег существует:

<pr>
<color>
<quote>
<code>

Правильный ответ: <code>

Примечание: В HTML существует тег <code>, и он предназначен для вывода программного кода, допустим, на языке C++.

8. Какой из вариантов содержит ошибку:

Ссылка
Ссылка
Ссылка
Ссылка

Правильный ответ: Ссылка

Примечание: Любое значение (либо его часть) атрибута должно находиться внутри кавычек.

9. Мне хочется, чтобы при наведении мыши на изображение, появлялась всплывающая подсказка с текстом "Подсказка". Какой валидный код XHTML 1.1 для этого используется?

Правильный ответ:

Примечание: Атрибут title используется для задания всплывающей подсказки, а атрибут alt является обязательным в стандарте XHTML 1.1.

10. Какой из представленных вариантов является валидным по XHTML 1.1:

Правильный ответ:

Примечание: Согласно стандарту XHTML 1.1 в каждом теге должен быть атрибут alt. Также каждый одиночный тег должен закрываться так: />.

CSS

11. Какое свойство используется для задания отступов у блока?

position
margin
direction
padding

Правильный ответ: margin

Примечание: Свойство margin используется для задания отступов, а padding используется для задания полей.

12. Какой CSS-код написан правильно?

```
div {border: 1px solid #ccc;}  
<div> {border: 1px solid #ccc;}  
<div> {border: 1px solid #hhh;}  
div {border: 1px solid #hhh;}
```

Правильный ответ: `div {border: 1px solid #ccc;}`

Примечание: Селектора `<div>` не существует, точно так же, как и не существует цвета `#hhh`.

13. Настя обнаружила, что цвет ссылок меняется, когда они являются посещёнными. Какой CSS-код необходимо задать, чтобы цвет посещённых и непосещённых ссылок был одним и тем же:

```
a:link {color: yellow;}  
a:link, a:active {color: yellow;}  
a:link, a:visited {color: yellow;}  
a:active, a:visited {color: yellow;}
```

Правильный ответ: `a:link, a:visited {color: yellow;}`

Примечание: Псевдоэлемент `link` отвечает за непосещённые ссылки, а `visited` за посещённые ссылки. Поэтому чтобы цвет был одинаковым у посещённых и непосещённых ссылок, необходимо задать свойство `color` сразу для двух псевдоэлементов.

14. Есть такой HTML-код: `<p>Синий не синий</p>`. Какой CSS-код внутри тега `` сделает синий цвет:

```
span {color: blue;}  
.blue {color: blue;}  
p span {color: blue;}
```

Все представленные варианты подойдут

Правильный ответ: Все представленные варианты подойдут

Примечание: Все варианты рабочие, у них различна только область действия, но до требуемого элемента доберутся все представленные селекторы.

15. Есть такой CSS-код: `body {font-size: 14pt;} p {font-size: 2em;}`. Какой размер текста будет в теге `<p>`:

16pt
12pt
7pt
28pt

Правильный ответ: 28pt

Примечание: `em` - относительная величина, которая позволяет задавать кратные значения размера от базового. Значение 2 удваивает размер базового: $14pt * 2 = 28pt$.

16. Алексей написал такой HTML-код: `<div id="myid">Содержимое</div>`. Как задать стиль для тега `<div>`?

```
div#myid {margin: 1px;}  
div[myid] {margin: 1px;}
```



```
.myid {margin: 1px;}  
div.myid {margin: 1px;}
```

Правильный ответ: `div#myid {margin: 1px;}`

Примечание: Поскольку у элемента `div` есть атрибут `id`, то здесь подходит селектор ID: `div#myid`.

17. Какой вариант задания цвета НЕ сработает?

```
color: #aaa;  
color: #000;  
color: #aaaaaa;  
color: #hhh;
```

Правильный ответ: `color: #hhh;`

Примечание: Цвета `#hhh` не существует, поскольку он задаётся через 16-ти ричную систему счисления. А её максимальная цифра - F.

18. Есть такой CSS-код во внешнем файле: `p {color: blue;}`. На странице написан такой HTML-код: `<p style="color: red;">текст</p>`. Какого цвета будет "текст"?

В браузере IE8 синего, а в других красного.
Красного.
Чёрного.
Синего.

Правильный ответ: Красного.

Примечание: Цвет будет красным, поскольку стиль через атрибут `style` более приоритетный, чем из внешнего файла.

19. В каком из представленных ниже вариантов содержится явная ошибка:

```
p span {font-size: 150%;}  
p {font-size: 150%;}  
p text {font-size: 150%;}  
p span#text {font-size: 150%;}
```

Правильный ответ: `p text {font-size: 150%;}`

Примечание: Тега `<text>` не существует, как следствие, селектора "p text" существовать так же не может.

20. Какое свойство используется для задания полей у блока?

```
position  
padding  
direction  
margin
```

Правильный ответ: `padding`

Примечание: Свойство `padding` используется для задания полей, а `margin` используется для задания отступов.

ТЕСТ «JavaScript»

1. Какой из вариантов объявления функции правильный: `var func = function() {}`; или `function func() {}`

Оба варианта неправильные.

`function func() {}`

`var func = function() {}`;

Оба варианта правильные.

Правильный ответ: Оба варианта правильные.

Примечание: Оба варианта объявления функций допустимы и являются правильными.

2. Какая функция вызывает окно с текстовым полем, в которое пользователь может ввести строку?

`prompt()`

`prompt()`

`alerts()`

`alert()`

Правильный ответ: `prompt()`

Примечание: Функция `prompt()` служит для вывода сообщения пользователю и текстового поля, в которое пользователь может ввести строку.

3. Есть такой код: `<script type="text/javascript">`

```
var a = prompt("Введите число");
```

```
var b = a + 1;
```

```
alert(b);
```

```
</script>
```

При выполнении скрипта появилось окно, в которое пользователь ввёл 10. Что будет написано в следующем появившемся окне?

2

101

11

1

Правильный ответ: 101

Примечание: Функция `prompt()` всегда возвращает строку. Для того, чтобы использовать эту строку как число, нужно использовать функцию `Number()`. В данном же примере к строке "10" прибавляется число. По правилам JavaScript число преобразуется в строку и происходит соединение строк. Потому и получается "101".

4. Виталий написал следующий код: `<script type="text/javascript">`

```
var a = 5;
```

```
var A = 10;
```

```
alert(a);
```

```
</script>
```

 Что будет написано в появившемся окне?

5

Ошибка из-за попытки объявить две переменных с одинаковыми именами.

Ошибка, поскольку название переменной не может содержать заглавных букв.

10

Правильный ответ: 5

Примечание: В JavaScript регистр имён переменных имеет значение. То есть переменная `a` - это далеко не переменная `A`, поэтому их значения различны.

5. Пётр написал код: `<script type="text/javascript">`

```
var a = "5";  
var b = 2;  
var c = a + b;  
alert(c);
```

`</script>` Что будет написано в появившемся окне?

52

7

NaN

Ошибка выполнения скрипта.

Правильный ответ: 52

Примечание: При сложении числа и строки число преобразуется в строку и происходит сложение строк.

6. Какая разница между функциями `setInterval()` и `setTimeout()`?

Никакой разницы нет.

В функции `setInterval()` можно задавать любой интервал для выполнения кода, тогда как в `setTimeout()` задержка всегда равна 1000 мс.

Функция `setInterval()` вызывает функцию через заданный промежуток только 1 раз, а функция `setTimeout()` вызывает заданную функцию постоянно через заданный промежуток времени.

Функция `setInterval()` вызывает заданную функцию постоянно через заданный промежуток времени, а функция `setTimeout()` вызывает функцию через заданный промежуток только 1 раз.

Правильный ответ: Функция `setInterval()` вызывает заданную функцию постоянно через заданный промежуток времени, а функция `setTimeout()` вызывает функцию через заданный промежуток только 1 раз.

Примечание: Функция `setInterval()` позволяет выполнять функцию (либо код) постоянно через заданный промежуток времени. Тогда как функция `setTimeout()` вызывает заданный код (или функцию) через определённое время, заданное в параметре, но только 1 раз.

7. Есть код: `<script type="text/javascript">`

```
var a = 1;  
function func() {  
  alert(a);  
}
```

```
func();
```

`</script>` Что будет написано в появившемся окне?

Ничего не будет, поскольку возникнет ошибка "переменная не определена".

undefined

1

0

Правильный ответ: 1

Примечание: Все переменные, которые объявляются за пределами функций являются глобальными, следовательно, их значение может быть использовано и внутри функций.

8. Какое значение вернёт функция, если в качестве параметра будет передано число

```
5:<script type="text/javascript">  
function func(a) {  
    var b = a + a;  
    return b + a;  
}  
</script>
```

- 5
- 15
- 20
- 10

Правильный ответ: 15

Примечание: Вначале идёт присвоение к переменной **b** значения $5 + 5$, то есть 10. Далее перед возвращением значения идёт ещё одна операция сложения, после которой возвращается число 15.

9. Алексей написал код: <script type="text/javascript">

```
var a = 1;  
var b = ++a + 1;  
alert(b);  
</script> Что будет написано в появившемся окне?
```

- 1
- Ничего не будет, поскольку инкремент пишется всегда так: $a++$.
- 2
- 3

Правильный ответ: 3

Примечание: Ввиду того, что инкремент идёт перед именем переменной, то прежде чем прибавить 1 и записать данное значение в переменную **b**. Переменная **a** увеличивается на 1, потом прибавляется ещё 1 и записывается в переменную **b**. Здесь ключевую роль играют приоритеты операций.

10. Какая разница между равенством и эквивалентностью?

Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.

Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.

Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.

Никакой разницы нет.

Правильный ответ: Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.

Примечание: Знак эквивалентности используется для сравнения и значений операндов, и их типов. Тогда как знак равенства проверяет лишь значения операндов.

4.5. Типовые задания для оценки освоения учебной и производственной практик

Дифференцированные зачеты по учебной и производственной практике выставляется на основании результатов выполненных в ходе практик работ и защиты отчетов.

Итоги учебной и производственной практики оформляются в соответствии с «Методическими указаниями по организации практической подготовки обучающихся в АНО ВО «МЭО Университет»».

Итоговая оценка за дифференцированный зачет рассчитывается по формуле:

(Средний балл по результатам выполненных работ)*0,75 + (Защита отчета)*0,25
с применением правил математического округления

4.6. ТИПОВЫЕ ЗАДАНИЯ ЭКЗАМЕНА ПО МОДУЛЮ

Обязательные документы: - Сводная ведомость оценивания экзамена (квалификационного) по профессиональному модулю ПМ.02 Подготовка интерфейсной графики;

- аттестационный лист по учебной практике, дневник обучающегося;
- аттестационный лист по производственной практике, дневник обучающегося;
- характеристика профессиональной деятельности обучающегося во время производственной практики.

Дополнительные материалы:

- Доклады участников научно-практических конференций;
- результаты участия во внеурочной научно-исследовательской деятельности;
- Грамоты за спортивные и общественные достижения;
- Дипломы и свидетельства за участие в олимпиадах и конкурсах профессионального мастерства по профессии «Оператор информационных систем и ресурсов»;
- портфолио в электронном виде (сообщения, рефераты, доклады, отчеты по практическим занятиям, видеоматериалы, фотоматериалы, презентации профессиональной направленности, выполненные обучающимися во время самостоятельной работы);
- свидетельства, подтверждающие участие в коллективных творческих мероприятиях (ведущий тематического вечера, член жюри, участник слета, участник турпохода, и т. д.).

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Профессиональных компетенций:		
Раздел 1. Подготовка интерфейсной графики		
ПК 2.1 Создавать визуальный дизайн элементов графического пользовательского интерфейса	<p>Оценка «отлично» - элементы графического пользовательского интерфейса полностью соответствуют заданным характеристикам, соблюдены и пояснены все этапы работы.</p> <p>Оценка «хорошо» - элементы графического пользовательского интерфейса полностью соответствуют заданным характеристикам, соблюдены и пояснены основные этапы работы.</p> <p>Оценка «удовлетворительно» - элементы графического пользовательского интерфейса соответствуют заданным характеристикам, соблюдены основные этапы работы.</p>	<p>Экзамен (по модулю)</p> <p>Экзамен и дифференцированный зачет по междисциплинарным курсам</p> <p>Оценка выполнения практических работ в ходе практических занятий</p> <p>Защита отчетов по практике</p> <p>Интерпретация результатов</p>

		наблюдений за деятельностью обучающегося в процессе практики
ПК 2.2 Подготавливать графические материалы для включения в графический пользовательский интерфейс	<p>Оценка «отлично» - разработан пользовательский интерфейс по определенному ранее визуальному стилю, подобраны технические параметры интерфейсной графики для заданного стиля и требований к графическому пользовательскому интерфейсу в соответствии с заданием, интерфейсная графика оптимизирована под различные разрешения экрана. Пояснены все этапы работы.</p> <p>Оценка «хорошо» - разработан пользовательский интерфейс по определенному ранее визуальному стилю, подобраны технические параметры интерфейсной графики для заданного стиля и требований к графическому пользовательскому интерфейсу в соответствии с заданием, интерфейсная графика оптимизирована под различные разрешения экрана допущены небольшие погрешности, пояснены основные этапы работы.</p> <p>Оценка «удовлетворительно» - разработан пользовательский интерфейс по определенному ранее визуальному стилю, интерфейсная графика оптимизирована под различные разрешения экрана.</p>	<p>Экзамен (по модулю)</p> <p>Экзамен и дифференцированный зачет по междисциплинарным курсам</p> <p>Оценка выполнения практических работ в ходе практических занятий</p> <p>Защита отчетов по практике</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
Раздел 2. Основы HTML, CSS и JavaScript		
ВПК 2.3 Применять основы HTML, CSS и JavaScript в профессиональной деятельности	<p>Оценка «отлично» - - разработан пользовательский интерфейс по определенному ранее визуальному стилю, подобраны технические параметры интерфейсной графики для заданного стиля и требований к графическому пользовательскому интерфейсу в соответствии с заданием, интерфейсная графика оптимизирована под различные разрешения экрана, пояснены все этапы работы.</p> <p>Оценка «хорошо» - - разработан пользовательский интерфейс по определенному ранее визуальному стилю, подобраны технические параметры интерфейсной графики для заданного стиля и требований к графическому пользовательскому</p>	<p>Экзамен (по модулю)</p> <p>Экзамен и дифференцированный зачет по междисциплинарным курсам</p> <p>Оценка выполнения практических работ в ходе практических занятий</p> <p>Защита отчетов по практике</p>

	<p>интерфейсу в соответствии с заданием, интерфейсная графика оптимизирована под различные разрешения экрана, выполнены разметка и форматирование, пояснены основные этапы работы.</p> <p>Оценка «удовлетворительно» - разработан пользовательский интерфейс по определенному ранее визуальному стилю, подобраны технические параметры интерфейсной графики для заданного стиля и требований к графическому пользовательскому интерфейсу в соответствии с заданием, интерфейсная графика оптимизирована под различные разрешения экрана, допущены небольшие погрешности</p>	Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики
Общих компетенций:		
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	<p>– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач;</p> <p>- адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач</p>	Экспертное наблюдение за выполнением заданий в ходе практической подготовки
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по профессии для решения профессиональных задач	
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	<p>- эффективно взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик;</p> <p>- обоснованность анализа работы членов команды (подчиненных)</p>	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	- демонстрация грамотной устной и письменной речи, ясность формулирования и изложения мыслей	
ОК 09. Пользоваться профессиональной документацией на	- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.	

государственном и иностранном языках.		
---------------------------------------	--	--

Требования к структуре оформлению и защите портфолио:

1. Портфолио оформляется обучающимся в течение всего периода освоения профессионального модуля, в том числе в период учебной и производственной практики.
2. Оформление в соответствии с эталоном (титульный лист, паспорт портфолио);
3. Защита портфолио в виде компьютерной презентации, выполненной в среде Power Point, сопровождаемой докладом. Карта формирования общих компетенций.

Критерии оценки портфолио

№	Показатель оценки результата	Документ портфолио	Оценка сформированности компетенции (да- 1балл \ нет - 0 баллов)
ОК 01	– обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач	дневник (учебной) производственной практики; аттестационные листы	
ОК 02	- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернетресурсы, периодические издания по специальности для решения профессиональных задач - эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту	дневник (учебной) производственной практики; аттестационные листы портфолио в электронном виде (сообщения, рефераты, доклады, отчеты по практическим занятиям, видео материалы, фотоматериалы, презентации профессиональной направленности, выполненные обучающимися во время самостоятельной работы)	
ОК 03	- демонстрация ответственности за принятые решения; - обоснованность самоанализа и коррекция результатов собственной работы	дневник (учебной) производственной практики; аттестационные листы	

ОК 04	<p>- эффективно взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик;</p> <p>- обоснованность анализа работы членов команды (подчиненных)</p>	<p>портфолио в электронном виде (сообщения, рефераты, доклады, отчеты по практическим занятиям, видео материалы, фотоматериалы, презентации профессиональной направленности, выполненные обучающимися во время самостоятельной работы)</p>	
ОК 09	<p>- эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке</p>	<p>портфолио в электронном виде (сообщения, рефераты, доклады, отчеты по практическим занятиям, видео материалы, фотоматериалы, презентации профессиональной направленности, выполненные обучающимися во время самостоятельной работы)</p>	

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Время на подготовку – 30 минут, время на ответ – 15 минут

Вопросы к теоретической части экзамена

1. Технические средства ввода графической информации.
2. Технические средства получения твердой копии графической информации.
3. Дисплей как техническое средство компьютерной графики.
4. Векторная и растровая графика: суть, отличия, области применения.
5. Мировые координаты, нормированные координаты, координаты устройства, функция кадрирования.
6. Понятие графического примитива. Наиболее распространенные графические примитивы и операции над ними.
7. Основные отличия текстового и графического режима видеоадаптера.

8. Чем отличаются с точки зрения машинной графики видеоадаптеры EGA, VGA, SVGA, MGA.
9. Особенности представления цвета в видеоадаптерах EGA и VGA.
10. Как программно осуществляется управление принтером.
11. Основные отличия в подходах MS DOS и WINDOWS при разработке графических приложений.
12. Основные этапы преобразования и модели, используемые при переходе от изображений реального мира к компьютерным.
13. Основные этапы растр-векторного преобразования графических объектов.
14. Понятие аффинных преобразований и их прикладное значение для задач компьютерной графики.
15. Элементарные аффинные преобразования на плоскости, составляющие базис операций машинной графики.
16. Понятие и прикладное значение однородных координат.
17. Элементарные аффинные преобразования в пространстве, составляющие базис операций машинной графики.
18. Основные виды проекций и соответствующие им аффинные преобразования.
19. Геометрические сплайны.
20. Алгоритм Брезенхема.

ЭТАЛОННЫЕ ОТВЕТЫ

Вопрос 1. ТЕХНИЧЕСКИЕ СРЕДСТВА ВВОДА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.

1. Мышь

Наиболее распространенным устройством ввода графической информации в ПЭВМ является мышь. Она подключается к компьютеру через интерфейс RS-232. При перемещении мыши и/или нажатии/отпускании кнопок мышь передает в компьютер информацию о своих параметрах (величине перемещения и статусе кнопок). Существует много различных типов устройства мыши, отличающихся как по принципу работы (механический, оптомеханический, оптический), так и по протоколу общения с ЭВМ. "Взаимопонимание" между мышью и ЭВМ при этом достигается с помощью драйвера, поставляемого вместе с мышью. Драйвер отслеживает перемещение мыши и нажатие/отпускание кнопок и обеспечивает работу с курсором мыши на экране дисплея.

Конструктивно близок к мыши манипулятор джойстик. Он представляет собой свободно передвигаемый стержень (ручку) и две кнопки-переключателя. Стержень джойстика передвигается в двух измерениях (координаты X и Y). Нажатие кнопок-переключателей фиксируется и обрабатывается программно. Обычно джойстик подключается к специальному игровому порту и в настоящее время в машинной графике используется редко.

Работа с мышью организуется через механизм прерываний. Прикладная программа вызывает прерывание 33h, передавая в регистрах необходимые параметры и получая через регистры возвращаемые драйвером значения. Существуют различные стандарты работы с мышью. Наиболее распространены стандарты IBM Microsoft. Из драйверов наиболее известны mouse.coin и gmouse.com Они поддерживают множество функции мыши, связанных с внешним видом, положением и перемещением курсора, а также с определением состояния кнопок мыши. Для программирования той или иной функции мыши требуется только знать ее номер и параметры, заносимые в регистры перед вызовом прерывания. Обычно номера функций драйвера заменяются их мнемоническими именами

в заголовочном файле (например, mouse.h), сами тексты функций собираются в отдельный файл (например, mouse.cpp). Такая пара (mouse.h и mouse.cpp) ориентируется на конкретный драйвер. В [1,4] приводятся варианты программного обеспечения - для драйверов mouse.com и gmouse.com.

Любая библиотека обычно содержит следующие функции:

- проверка наличия мыши;
- показ/сокрытие курсора мыши (при сокрытии драйвер мыши продолжает отслеживать ее перемещение);
- чтение состояния мыши (ее координат и состояния кнопок - нажато/отжато);
- передвижение курсора мыши в заданную точку;
- установка области перемещения курсора мыши.

По умолчанию форма курсора мыши определяется оборудованием и драйвером. Ее можно изменить. В текстовом режиме курсор мыши отображается на экране совместно с текстовым курсором и представляет собой прямоугольник размером в один символ. Вид изображения при перекрытии курсором мыши чего-либо определяется параметром и передаваемыми функцией изображения курсора текстового режима. Эти параметры (маска экрана и маска курсора) состоят из 16 бит и задают мерцание, цвет и фон, также изображаемый при наложении курсора символ. Маска экрана участвует в логической операции AND с атрибутами перекрытого участка экрана, далее выполняется операция XOR с маской курсора. Например, для инвертирования изображения маска экрана - 0xFFFF, маска курсора - 0x770.

В графическом режиме также имеется курсор по умолчанию (от драйвера). Обычно это небольшая стрелка. Вид курсора также можно изменить. Над маской экрана и маской курсора выполняются аналогично текстовому режиму операции AND и XOR. Но под каждую маску отводится не 16 бит, а по 16 16-битовых величин (int mask[1][15]). Для

Маска экрана	Маска курсора	Рез-т на экране
0	0	0
0	1	1
1	0	Не измен.
1	1	Инверсия

создания собственного курсора полезна таблица взаимодействия масок:

Следует обратить внимание на рациональную реализацию обработки событий от мыши. Не требуется; все время

опрашивать драйвер мыши. Ему передается адрес функции, которую следует вызвать при наступлении заданного события. Первый параметр - указатель на функцию, второй параметр - маска событий. События соединяются побитовой операцией ИЛИ. Функция, которая обрабатывает событие, получает маску вызывающего события, маску состояния кнопок мыши. координаты курсора мыши.

2. Сканеры

Сканеры являются устройствами ввода изображений. Чаще всего их действие основано на оптических принципах. Они осуществляют начальную оцифровку изображений (далее при необходимости производится чистка изображений специальными методами - см. тему "Математические основы компьютерной графики") и передачу их в ЭВМ. В настоящее время фактическим стандартом представления изображений сканерами является формат TWAIN. Этот формат поддерживает большинство драйверов различных сканеров. Конвертация из этого формата в формат какой-либо графической системы выполняется программно.

3. Световое перо

Световое перо представляет собой цилиндр, содержащий оптическую систему и фотоэлемент, вырабатывающий напряжение при попадании на него света. При прикосновении светового пера к поверхности экрана компьютера фотоэлемент генерирует электрический импульс каждый раз, когда электронный луч дисплея в процессе сканирования пробегает точку, на которую установлено световое перо. Таким образом,

считываются координаты точки экрана, на которой расположено световое перо и обеспечивается "рисование" на экране. Основное применение светового пера - автоматизированное проектирование.

4. Дигитайзер (дигитайзер, digitazer, оцифровыватель)

Устройство ввода точных двумерных координат объекта. Подключается к асинхронному порту COM1. Пример дигитайзера - изделие TRUE GRID фирмы Houston Instruments представляет собой панель размером от 130*130 мм до 280*430 мм и снабжаются курсором в виде пера и напоминающей мышь коробочки с лупой, перекрестьем и одной или несколькими клавишами. Выпускает дигитайзеры также фирма Hewlett Packard и ряд др. фирм. Возможны бинарная передача данных, ASCII-строка, целочисленный ASCII-формат.

Съем координат может производиться в следующих режимах:

- точки (point) - передача абсолютных координат точки, в которой находится курсор, по нажатию клавиши;
- триггер (triggered) - абсолютные координаты точки по запросу компьютера;
- обычный поток (stream) - непрерывная передача абсолютных координат;
- переключаемый поток (switch stream) - аналогично обычному потоку, но включается по нажатию клавиши;
- непрерывная передача относительных координат.

ВОПРОС 2. ТЕХНИЧЕСКИЕ СРЕДСТВА ПОЛУЧЕНИЯ ТВЕРДОЙ КОПИИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ.

1. Графопостроители (плоттеры)

Это электромеханические устройства, основанные на преобразовании хранящихся в памяти ЭВМ координат изображения в сигналы перемещения механических пишущих узлов. Различные типы графопостроителей имеют различные системы команд, позволяющие управлять механическими узлами, обеспечивающие нанесение изображения как в одном, так и в нескольких цветах, с различными атрибутами (пунктир, штрих-пунктир и т.п.). Обычно "плоттер подключается к компьютеру через асинхронный порт COM1. Для выполнения рисунка плоттеру передаются команды (рисование линии, рисование окружности и т.д.), цвет и координаты точек, образующих линию. Эти команды образуют графические языки плоттеров.

2. Принтеры

Практически любой современный принтер позволяет получать изображение, т.к. выводит информацию по точкам. Каждый символ представляется матрицей точек. Для большинства матричных принтеров размер матрицы 8*12. Управляет принтером специальный набор команд, обычно называемый Esc-последовательностями. Эти команды позволяют задать режим работы принтера, прогон бумаги на заданное расстояние, собственно печать. Чтобы отличить управляющие коды от выводимой информации, они обычно начинаются с кода меньшего, чем 32 (не ASCII-символ). Для большинства команд начальным является символ Esc (код 27) совокупность подобных команд образует язык управления принтером. Каждый принтер имеет свой набор команд. Однако можно выделить набор команд, реализованный на достаточно широком классе принтеров.

Наиболее просты 9-игольчатые принтеры типа Epson, Star и совместимые с ними. Они имеют команды перевода строки (LF) возврата каретки к началу строки (CR), прогона бумаги до начала новой страницы (FF) установки интервала между строками, печати с нормальной или повышенной плотностью (80 или 120 точек дюйм). 24-игольчатые принтеры (LQ-принтеры) имеют язык управления, являющийся надмножеством языка управления 9-игольчатыми принтерами. Этим достигается программная совместимость. Большинство струйных принтеров на уровне языка управления совместимо с LQ-принтерами. Одним из наиболее распространенных классов лазерных принтеров являются

принтеры серии HP LaserJet фирмы Hewlett Packard. Все они управляются языком PCL, также основанным на Esc-последовательностях.

Большинство принтеров работают с параллельным портом ЭВМ, который называется нередко принтерным портом. В устройстве самого параллельного интерфейса имеется только один специальный сигнал, который компьютер может послать в принтер — сигнал инициализации. Остальные коды управления принтером передаются в потоке данных и должны формироваться программно. Принтер может послать компьютеру 3 сигнала:

- подтверждение получения данных;
- ожидания (задержки передачи данных до тех пор, пока принтер не сможет начать обработку данных снова;
- отсутствия бумаги.

Первые два сигнала характерны для любой передачи данных. Последний сигнал является особенностью параллельного интерфейса. Следует также отметить, что параллельный интерфейс является односторонним осуществляет только вывод данных.

Некоторые принтеры имеют две модификации - для параллельного и последовательного интерфейса. Лазерные принтеры фирмы Hewlett Packard работают только с последовательным интерфейсом со скоростью передачи данных 9600 бод (бит/сек).

ВОПРОС 3. ДИСПЛЕЙ КАК ТЕХНИЧЕСКОЕ СРЕДСТВО КОМПЬЮТЕРНОЙ ГРАФИКИ.

Это основное устройство вывода информации. Большинство дисплеев в качестве формирователя изображения использует электронно-лучевую трубку (ЭЛТ). Работа ЭЛТ основана на двух физических принципах: влияние электромагнитного поля на поток электронов, движущихся в разреженном пространстве и свечение люминофоров при их бомбардировке электронами.

В памяти ЭВМ хранятся координаты точек изображения и информация об их цвете, яркости и др. (например, атрибут мерцания). Эти данные под управлением дисплейного контроллера преобразуются в сигналы управления лучом ЭЛТ. Существует 2 основных типа дисплеев, использующих ЭЛТ: векторные и растровые.

Векторные дисплеи наиболее просты, требуют меньше памяти для хранения информации. Электронный луч последовательно обходит траекторию из отрезков прямых (векторов), представляющих рисунок, воспроизводимый на экране. Изображения, формируемые векторными дисплеями, проигрывают по качеству растровым.

Растровые дисплеи являются доминирующими. Они позволяют формировать практически любые изображения. Используется тот же принцип движения луча, что и в телевизоре. Электронный луч циклически совершает движение, образующее на экране последовательность строк (растр) Движение луча начинается в левом верхнем углу, выполняется перемещение от точки А к точке В. Затем луч быстро отклоняется в точку С. Отрезок прямой АВ называется прямым ходом луча по строке, отрезок ВС - обратным. Суммарное время, затрачиваемое на это перемещение, - период строчной развертки.

А — начальная точка строки. В — конечная точка строки. С — точка, в которую отклоняется луч. Z — конечная точка раstra. Движение луча от точки А до точки Z называется прямым ходом луча по кадру. Из точки Z луч быстро перемещается в точку А, сканирование завершается. Время одного полного движения по растру - период кадра.

Реальные дисплеи имеют от 300 до 2000 строк. Изображения, формируемые растровыми дисплеями состоит из множества точек пикселей. Термин "пиксел" происходит от английских слов PICTURE ELEMENT. Множество всех пикселей на экране образует матрицу. Размерность матрицы различна для различных устройств, она

определяет разрешающую способность дисплея. Управление работой дисплея осуществляет дисплейный контроллер (видеоконтроллер, видеоадаптер, дисплейный адаптер, видеокарта). Он представляет собой плату, вставляемую в соответствующий слот, и поэтому может заменяться. Видеоадаптер выполняет 3 главные функции: хранение информации об изображении; регенерацию изображения на экране ЭЛТ; связь с центральным процессором ЭВМ. ЭВМ имеет многочисленные видеорежимы или способы изображения данных на экране дисплея. Каждый видеоадаптер имеет свой набор видеорежимов. Изображение хранится в растровом виде в памяти видеокарты. Аппаратно обеспечивается регулярное (50-70 раз в сек.) чтение этой памяти и отображение ее на экране. Поэтому работа с изображениями сводится к операциям с видеопамью. Существует 6 общепринятых стандартов видеоконтроллеров. Имеется также множество нестандартных для решения специальных задач. К стандартным видеоконтроллерам относятся:

Монохромный дисплейный адаптер (Monochrome Display Adapter - MDA) - текстовый, высокое качество изображения, низкая цена;

Цветной графический адаптер (Color Graphics Adapter - CGA). Разрешающая способность в цветном графическом режиме 320*200, в режиме монохромной графики - 640*200. Палитра из 16 цветов, в графическом режиме можно задать любые 4 цвета. Устарел, практически не используется;

Монохромный графический адаптер (Monochrome Graphics Adapter - MGA или, по имени компании-разработчика Hercules Computer Technology, Hercules Graphics Adapter - HGA). Имеет ту же разрешающую способность, что и MDA, но может работать в графическом режиме. Разрешающая способность 720*348. Изображение качественное, используется широко.

Улучшенный графический адаптер (Enhanced Graphics Adapter - EGA). Разрешение 640*350, 16 цветов. Благодаря новой организации управления памятью и формированием изображения можно смешивать цвета в различных комбинациях из палитры в 64 оттенка для каждого из 16 цветов (оттенки тона, насыщенность). Как правило, обеспечивается совместимость с CGA, в ряде моделей - с MGA (Hercules). Сейчас есть усовершенствованные модели, позволяющие при наличии специального программного обеспечения получать 43 строки на экране и разрешение 640*480;

Видеографическая матрица (Video Graphics Array - VGA). Была создана для PS/2. Развитие EGA. 640*480 точек, воспроизведение 16 цветов из палитры 4096 оттенков. 320*200 при воспроизведении одновременно 256 цветов;

Супер видеографическая матрица (Super Video Graphics Array - SVGA). Стандарта SVGA нет, он рассматривается как расширение VGA. Более высокая частота горизонтальной развертки - ряд частот: 56, 60, 72. Разрешение: 800*600, 1024*768. 1280*1024.

ВОПРОС 4. ВЕКТОРНАЯ И РАСТРОВАЯ ГРАФИКА: СУТЬ, ОТЛИЧИЯ, ОБЛАСТИ ПРИМЕНЕНИЯ.

Принципы, положенные в основу работы дисплеев, широко используются в машинной графике как способ формирования изображений. Поэтому часто встречаются термины "ВЕКТОРНАЯ ГРАФИКА" и "РАСТРОВАЯ ГРАФИКА". В первом случае выполняется кусочно-линейная аппроксимация изображений и возникает задача поиска компромисса между временем и точностью построения изображения путем подбора параметров аппроксимации. Во втором случае этот же компромисс выглядит как задача определения параметров раstra.

ВОПРОС 5. МИРОВЫЕ КООРДИНАТЫ, НОРМИРОВАННЫЕ КООРДИНАТЫ, КООРДИНАТЫ УСТРОЙСТВА, ФУНКЦИЯ КАДРИРОВАНИЯ.

Для программиста естественно желание определить графические элементы в системе координат решаемой задачи. Устройства вывода, на которых визуализируются графические элементы, требуют, как правило, использования собственных аппаратных координатных систем. Чтобы разрешить это противоречие и добиться независимости от устройств, международным стандартом GKS (Graphical Kernel System - ядро графической системы) определены 3 системы координат.

Задавая элементы своего изображения, прикладной программист использует систему мировых координат (WC - World Coordinate). Эти координаты определяют положение объекта в некотором модельном мире. **МИРОВЫЕ КООРДИНАТЫ** - независимые от устройства декартовы координаты, которые используются в прикладной программе для задания графических данных ввода-вывода. Вообще говоря, каждый примитив может быть определен в собственной системе мировых координат.

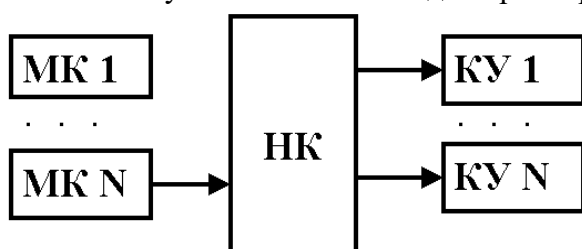
НОРМИРОВАННЫЕ КООРДИНАТЫ задаются в промежуточной, независимой от устройства системе координат и нормированы относительно некоторого диапазона (часто от 0 до 1). Относительное расположение примитивов ввода-вывода определяется отображением мировых координат в нормированные координаты. Нормированные координаты используются при хранении графических образов (в памяти, в файлах и пр.).

Пространство нормированных координат пересчитывается в координаты устройства (Device Coordinate).

КООРДИНАТЫ УСТРОЙСТВА зависят от вида устройства и измеряются в некоторой системе мер (метрической, в дюймах) или в аппаратных единицах.

Множество преобразований нормирования определяет отображение различных систем мировых координат в единое пространство нормированных координат. При работе с конкретным устройством производится отображение пространства нормированных координат в координаты устройства.

Пример - система автоматизированного проектирования печатных плат PCAD: различные типы дисплеев подключаются с помощью драйверов (конкретный драйвер указывается в файле pcaddrv.sys). Роль нормированной системы координат играет файл *.PLT, который не зависит от типа устройства. Вывод на конкретное устройство выполняют утилиты PCPRINT для принтера, PCPLOTS для плоттера.



Каждая из этих утилит позволяет настроиться на большой перечень устройств соответствующего класса. При этом выполняется преобразование нормированных координат файла *.PLT в координаты устройства. По тому же принципу работает Windows95 при формировании заданий на печать.

Если памяти достаточно, сначала формируется файл печати в формате расширенного метафайла (EMF). Метафайл формируется быстрее, т.к. не зависит от типа устройства. Далее в фоновом режиме формируется задание на печать с учетом конкретного устройства.

Отображение области действия мировых координат в область действия координат устройства называется *ФУНКЦИЕЙ КАДРИРОВАНИЯ*.

Изображение, получаемое средствами машинной графики, имеет четко определенную структуру. Атомарным (неделимым) объектом является *КООРДИНАТНАЯ ТОЧКА* или *ПИКСЕЛ*. Термин *ПИКСЕЛ*, введенный для дисплеев, нашел применение в более широком смысле. Это точка, являющаяся атомарным компонентом изображения вне зависимости от того, где и как хранится/отображается рисунок.

ВОПРОС 6. ПОНЯТИЕ ГРАФИЧЕСКОГО ПРИМИТИВА. НАИБОЛЕЕ РАСПРОСТРАНЕННЫЕ ГРАФИЧЕСКИЕ ПРИМИТИВЫ И ОПЕРАЦИИ НАД НИМИ

ГРАФИЧЕСКИЙ ПРИМИТИВ представляет собой либо координатную точку, либо упорядоченную последовательность (не совокупность, а именно последовательность!) координатных точек. Различают графические объекты (и соответственно графические примитивы): нульмерные (точки), одномерные (линии), двумерные (поверхности), трехмерные (тела).

В компьютерной графике все, что относится к двумерным изображениям, обозначают 2D (2-dimension), к трехмерным - 3D (3-dimension).

ВОПРОС 7. ОСНОВНЫЕ ОТЛИЧИЯ ТЕКСТОВОГО И ГРАФИЧЕСКОГО РЕЖИМА ВИДЕОАДАПТЕРА.

Любой современный видеоадаптер может работать в двух режимах: текстовом и графическом. В текстовом режиме экран делится на ячейки, соответствующие размеру символа. Обычно это 40 или 80 колонок, 25 или 50 строк. Каждая ячейка содержит атрибут и символ. Символ выводится на экран в ASCII коде, атрибут указывает, как представляется символ на экране (цвет, интенсивность, мерцание, подчеркивание, инверсное изображение в зависимости от типа видеоадаптера). Язык программирования C и другие языки такого же класса имеют средства управления выводом текста, в том числе управления атрибутами символов.

В графическом режиме экран представляется совокупностью пикселей. Количество точек (разрешающая способность) зависит от типа видеоадаптера и установленного для него режима.

В текстовом режиме верхний левый угол экрана имеет позицию (1,1), координата X растет вправо, координата Y растет вниз. В графическом режиме верхний левый угол имеет координаты (0,0), координаты X и Y направлены аналогично.

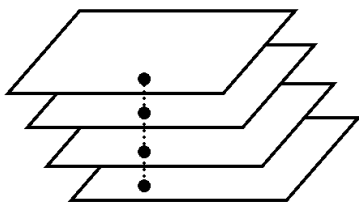
ВОПРОС 8. ЧЕМ ОТЛИЧАЮТСЯ С ТОЧКИ ЗРЕНИЯ МАШИНОЙ ГРАФИКИ ВИДЕОАДАПТЕРЫ EGA, VGA, SVGA, MGA.

Режим обозначается номером и определяется разрешением экрана и количеством цветов.

Номер режима	Разрешение	Кол-во цветов	Номер режима	Разрешение	Кол-во цветов
0Dh	320x200	16	11h (VGA)	640x480	2
0Eh	640x200	16	12h (VGA)	640x480	16
0Fh	640x320	2	13h (VGA)	320x200	256
10h	640x320	16			

Каждая видеоплата содержит собственный BIOS для работы с ней и поддержки основных своих функций. Через BIOS можно определить тип адаптера - EGA или VGA, установить нужный режим, системный шрифт заданного размера (8,14 или 16 пикселей высоты), палитру. Для 16-цветных режимов под каждый пиксел отводится 4 бита ($2^4=16$). Однако эти биты располагаются не последовательно в одном байте, а разнесены по 4 блокам (битовым или цветовым плоскостям) видеопамати. Вся видеопамать (обычно 256 К) делится на 4 равные части. Каждому пикселу соответствует по 1 биту каждой плоскости, причем эти биты расположены одинаково относительно начала плоскости (параллельно). Когда процессор выполняя операции чтения/записи видеобуфера по некоторому адресу, этот адрес относится не к одному, а к 4 байтам, каждый из которых

размещается в своей битовой плоскости. При выполнении операции чтения из видеобuffers (например, командами MOV reg,mem; LODS; CMP reg,mem и др.) из него извлекается не 1, а 4 байта. Но данные пересылаются не в процессор, а в четыре 8-битовых регистра-защелки (latch - задвижка, щеколда). Каждый из этих регистров соответствует своей битовой плоскости. При выполнении операций записи в видеопамять производится параллельная модификация всех 4 битовых плоскостей. Таким образом за один раз обрабатывается информация о 8 пикселах. Если к видеобuffers обратиться при помощи команд, оперирующих словами, а не байтами, результаты могут быть ошибочными, т.к. алгоритм выполнения операций процессора и видеокарты разный, и результат одной части операции перезаписывается другой ее частью.



Регистры видеокарты делятся на группы. Каждой группе соответствует пара последовательных портов (порт адреса и порт значения). Для записи в регистр значения надо записать сначала номер регистра в порт адреса, затем значение в следующий порт. Добраться до регистров видеокарты можно с помощью ассемблера или функций языка C inportb, outportb (запись в аппаратный порт). Прототипы функций - в <dos.h>.

Передачей данных между процессором, регистрами-защелками и видеобuffers управляет графический контроллер. В адаптере EGA это 2 микросхемы или отдельная СБИС, в адаптере VGA он входит в СБИС видеографической матрицы.

Графический контроллер имеет 9 регистров, адресуемых через порт 3CEh. Значения регистров задаются через порт 3CF. Содержимое регистров графического контроллера управляет обработкой данных регистров-защелок при чтении/записи. Часть операций в качестве операндов ИСПОЛЬЗУЮТ байт, т.е. воздействуют отдельно на каждый регистр. Операндом других операций является пиксел, т.е. содержимое регистров-защелок рассматривается как набор из 8 пикселов. Такие операции воздействуют на каждый пиксел в отдельности.

Т.к. разрядность процессора не более 32, требуется специальное формирование значения для пересылки в процессор. Оно осуществляется с помощью масок и зависит от режима чтения/записи. Режим задается в специальном регистре графического контроллера. Этот регистр имеет номер 5. Имеется 2 режима чтения и 3 режима записи для EGA. Для VGA имеется еще один режим записи. Бит 3 регистра определяет режим чтения (0 или 1), биты 1 и 0 - режим записи. Остальные биты этого регистра обычно нулевые.

В режиме чтения 0 в процессор передается значение одного из 4 регистров-защелок. Указателем номера регистра-защелки служит специальный регистр считываемого банка (еще одно название битовой плоскости) Этот регистр имеет номер 4. Такое последовательное чтение битовых плоскостей применяется, например, при записи изображения на диск.

В режиме чтения 1 задействованы 2 регистра видеокарты, управляющие цветами. Этот режим позволяет быстро находить пикселы, имеющие заданный цвет (требуется, например, при закрасивании, при разделении фоновых и нефоновых пикселов). Однако гарантированно быстро узнать цвет конкретного пиксела нельзя. Максимально для этого может потребоваться 16 раз считывание (по количеству цветов).

Режим записи 0 является наиболее сложным, но дает большие возможности. Операция записи процессора инициирует комбинацию байтных и пиксельных операций. Байт данных от процессора можно использовать для модификации содержимого любых или всех битовых плоскостей и одновременно некоторое заданное значение пиксела можно использовать для модификации всех или любых пикселов. Значение пиксела - еп цвет. В операции задействованы 4 служебных регистра графического адаптера, вместе с байтом данных от процессора воздействующих на регистры-защелки. Например, регистр битовой маски (номер 8) позволяет выделить нужный пиксел, чтобы сопоставить ему

определенный цвет. Регистр маски плоскости (относится к группе регистров, адресуемых через порт 3C4, порт данных - 3C5) защищает от изменения определенные плоскости. Для формирования значений используются также сдвиговые операции.

В режиме записи 1 значения регистров-защелок непосредственно копируются в соответствующие биты вые плоскости. Другие регистры не действуют, посланное процессором значение не учитывается. Этот режим позволяет быстро копировать содержимое видеопамати группами по 8 пикселей. Очевидно, режим может действовать только после заполнения регистров-защелок, когда процессор прочитает данные из видеобуфера. Обычно этот режим применяется при перемещении изображения из одной области экрана в другую.

В режиме записи 2 младшие 4 бита байта, посланного процессором, задают цвет отображения не защищенных битовой маской пикселей. Как уже отмечалось, регистр битовой маски защищает от изменения определенные плоскости. Регистр 3 графического контроллера устанавливает способ наложения новых пикселей на существующее изображение, т.е. логическую операцию, применяемую к регистрам-защелкам и значению посланному процессором. Этот режим удобен для записи в видеобуфер (на экран) отдельных пикселей.

Режим записи 3 поддерживается только адаптером VGA. В [3,4] излагается способ формирования данных для записи в битовые плоскости.

Работа VGA в 256-цветном режиме с разрешением 320*200 имеет особенности. Для одновременного отображения такого количества цветов под каждый пиксел отводится 8 бит. Эти биты идут последовательно образуя 1 байт. Плоскости не используются, видеопамать начинается с адреса 0xA000:0. Точке с координатам (x,y) соответствует байт памяти по адресу 320*y+x. Это стандартный режим с номером (mode) 13.

Существуют также нестандартные режимы адаптера VGA при работе с 256 цветами. Они программируются на ассемблере и позволяют установить повышенное разрешение (320*240 или 360*480). Здесь используются битовые плоскости, в которых в определенном порядке хранятся пиксели. В одной битовой плоскости хранятся пиксели 0,4,8 и т.д., в другой - 1,5,9 и т.д. Здесь также задействованы все служебные регистры, и меняется интерпретация находящихся в видеопамати значений.

Видеокарты SVGA совместимы с VGA, но имеют большой набор дополнительных режимов. VGA является стандартом, SVGA - его расширение.

В 256-цветном режиме в адаптерах SVGA под каждый пиксел отводится 1 байт, вся видеопамать разбивается на банки одинакового размера (обычно по 64 К). Область адресного пространства

0xA000:0 -0xA000:0xFFF соответствует выбранному банку. Ряд карт позволяет работать сразу с двумя банками.

Практически все различия между картами заключаются в установке режима с заданным разрешением и установке банка с заданным номером. Можно построить библиотеку, распознающую наличие основных SVGA карт (Triedent, Cirrus Logic и др.) и обеспечивающую работу с ними. Связь - через порты 0x3C4 и 0x3CE, работать можно на Си с привлечением ассемблера.

Ассоциацией стандартов в области видеоэлектроники VESA (Video Electronic Standards Association) сделана попытка стандартизации работы с различными SVGA-платами путем добавления в BIOS платы (у видеоадаптеров - свой BIOS) некоторого стандартного набора функций, обеспечивающего получение необходимой информации о карте, установку заданного режима и банка памяти. При этом вводится стандартный набор расширенных режимов. Номер режима - 16-битовое число, биты с 9 по 15 зарезервированы и должны быть; равны 0, бит 8 для VESA-режимов = 1, для «родных» режимов карты = 0.

Таблица основных VESA-режимов:

Номер	Разрешение	Бит на пиксел	Кол-во цветов	Номер	Разрешение	Бит на пиксел	Кол-во цветов
100h	640x400	8	256	111h	640x480	16	64К
101h	640x480	8	256	112h	640x480	24	16М
102h	800x600	4	16	113h	800x600	15	32К
103h	800x600	8	256	114h	800x600	16	64К
104h	1024x768	4	16	115h	800x600	24	16М
105h	1024x768	8	256	116h	1024x768	15	32К
106h	1280x1024	4	16	117h	1024x768	16	64К
107h	1280x1024	8	256	118h	1024x768	24	16М
10Dh	320x200	15	32К	119h	1280x1024	15	32К
10Eh	320x200	16	64К	11Ah	1280x1024	16	64К
10Fh	320x200	24	16М	11Bh	1280x1024	24	16М
110h	640x480	15	32К				

Ряд SVGA-карт поддерживает т.н. непалитровые режимы. Здесь для каждого пиксела вместо индекса в палитре непосредственно задается его RGB-значение. Обычно такими режимами являются HiColor (15 или К бит на пиксел) и TrueColor (24 бита на пиксел). Видеопамять, этих режимов устроена аналогично 256-цветным SVGA: под каждый пиксел отводится 2 байта для HiColor и 3 байта для TrueColor, байты расположены подряд и сгруппированы в банки. Наиболее проста организация TrueColor (16 млн. цветов) - 1 байт под каждую из компонент цвета. Для HiColor под каждый пиксел отводится 2 байта. Здесь возможны варианты:

- каждая компонента занимает по 5 бит, последний бит не используется. Это дает всего 32 тысячи цветов;
- красная и синяя компоненты занимают по 5 бит, зеленая - 6 бит. Это дает всего 64 тысячи цветов.

ВОПРОС 9. ОСОБЕННОСТИ ПРЕДСТАВЛЕНИЯ ЦВЕТА В ВИДЕОАДАПТЕРАХ EGA И VGA.

Известно, что любой цвет является композицией трех основных цветов: Red (красный), Green (зеленый), Blue (синий). Дополнительные цвета - смесь основных:

При различной аппаратной настройке монитора magenta, например, может быть пурпурным, сиреневым, малиновым, вишневым; голубой - бирюзовым.

В зависимости от интенсивности (яркости) каждого из основных цветов в смеси получаем различные оттенки. Следовательно, 4-битовая комбинация позволяет закодировать 15 базовых цветов. Кодирование стандартно: 1 - интенсивность, RGB - цвета.

Т.к. емкость видеопамяти ограничена, в ряде случаев возникает конфликт между цветностью и разрешающей способностью. Этим объясняется возможность цветных видеоадаптеров работать в разных режимах (модах), позволяющих увеличивать разрешающую способность (количество точек) за счет уменьшения количества цветов и наоборот. (См. параметр graphmode функции initgraph()) Например, в простейшем случае для адаптера CGA возможны 2 варианта представления изображений: 2 бита на каждый пиксел (4 цвета, 320*200 точек) и 1 бит на каждый пиксел (2 цвета, 640*200 точек).

Для VGA можно получить 640*200 точек при воспроизведении 16 цветов из палитры 4096 оттенков или 320*200 точек при воспроизведении одновременно 256 цветов. Т.к. видеопамять VGA 256 К, можно также уменьшить число страниц видеобuffers (вместо 2 получить 1, 16 цветов, 640x480 точек).

Практически любой видеоадаптер способен отобразить гораздо больше цветов, чем определяется количеством бит, отведенным под один пиксел. Например, монитор EGA адаптера способен отображать 64 цвета т.к. имеет 6-битовый видеосигнал. Видеоадаптер переводит 4-битовый цвет пиксела в 6-битовый видеосигнал. Для перевода используется

некоторое подобие таблицы, называемое палитрой. Фактически адаптер имеет К специальных внутренних регистров, где для каждого логического цвета хранится его 6-битовое значение видеосигнала (6 бит, т.к. 3 основных цвета + бит интенсивности, следовательно, на каждый цвет - 2 бита). Цвет в палитре задается байтом вида 00rgbRGB. Малые буквы обозначают бит интенсивности соответствующего цвета,

Существует возможность менять в таблице соответствие логического цвета и видеосигнала, выбирая его из 64 возможных цветов.

(4-битовый атрибут) & (Регистр используемой цветовой матрицы) = Регистр палитры 0-0F h = 6-битовый сигнал, подаваемый на дисплей.

Реализация 16-цветной палитры для VGA гораздо сложнее. Для каждого цвета имеется 18-битовая раскладка по компонентам (6 бит на каждый из 3 основных цветов - в EGA было по 2 бита на цвет). Дополнительные биты дают более тонкую раскладку по интенсивности, т.е. обеспечивают более разнообразное смешивание 3 основных цветов. Схема получения 18-битового сигнала на начальном этапе повторяет EGA, далее задействован еще ряд регистров, операции логического И. .

Помимо поддержки EGA видеоадаптер VGA имеет 256 специальных регистров, где для каждого цвета хранится его 18-битное представление. Обычно BIOS записывает в эти регистры набор цветов, принимаемый по умолчанию. Этот режим используется для получения 256 цветов при разрешении 320*200. Распределение цветов по регистрам видеоадаптера VGA:

		Интенсивность	Интенсивность
0h – 0Fh	CGA-совместимые, цвета (по умолчанию)		
10h-1Fh	Шкала серого		
20h - 67h	Синий, красный, зеленый	Высокая	Высокая
		Средняя	
		Низкая	
68h – Afh		Высокая	Средняя
		Средняя	
		Низкая	
B0h - F7h		Высокая	Низкая
		Средняя	
		Низкая	
F8h - FFh	Черный		

Регистры 0h - 0Fh (h - hex, 16-ричный) обеспечивают CGA-совместимый набор цветов, принимаемый по умолчанию. Регистры 10h - 1Fh содержат упорядоченный по возрастанию набор оттенков серого. Следующие 216 регистров (20h - F7h) содержат 3 группы по 72 цвета, где первая группа (20h - 67h) - цвета повышенной яркости, вторая группа (68h - AFh) - средней, третья (B0h - F7h) - пониженной. Каждая такая группа состоит из трех диапазонов цветов, упорядоченных по снижению яркости. Снижение яркости можно трактовать как увеличение количества белого цвета, подмешанного к основному. Цвета в каждом диапазоне располагаются в порядке перехода от синего через красный к зеленому.

При 16-цветной работе VGA исходному логическому номеру цвета для 6-битовой палитры EGA сопоставляется, как и ранее, значение от 0 до 63. Но это уже не RGB-разложение цвета, а номер одного из 256 регистров, содержащих физический цвет. Для установки значений регистров служит функция void far setrgbpalette(int color, int red, int green, int blue);

color - логический номер цвета (от 0 до 15 или от 0 до 255 в зависимости от graphmode), остальное - его RGB-интенсивности (используются только младшие байты, только 6 битов каждого байта).

Функция `setpalette()` присваивает одному из 16 логических цветов значение физического цвета из диапазона 0 - 63. Функция `setrgbpalette()` делает то же самое, но диапазон цветов шире

ВОПРОС 10. КАК ПРОГРАММНО ОСУЩЕСТВЛЯЕТСЯ УПРАВЛЕНИЕ ПРИНТЕРОМ.

Принтеры очень хорошо приспособлены для вывода графики, т.к. строят изображение по точкам.

Принтером управляет специальный набор команд, описываемый в документации на принтер. Эти команды отражают аппаратные особенности принтера. Однако есть достаточно общий набор команд, реализованный для большинства принтеров. Это команды управления прогоном бумаги, плотностью печати, шрифтом (если принтер имеет возможность работы с различными шрифтами - зашиваются в ПЗУ) и т.д.

Каждая команда является набором символов (имеет символическое имя) или цифр (кодов). Коды просто посылаются на принтер во входном потоке. Чтобы отличить команды от того, что следует напечатать, они предваряются неотображаемым символом, т.е. символом с кодом, меньшим 32. Такому коду не соответствует ни один символ кода ASCII. Обычно в качестве символа начала управляющей последовательности выступает Esc (код 27). Поэтому говорят об Esc-последовательностях управления принтером.

При работе на языке C Esc-последовательности записываются в операторе вывода. Символ Esc выглядит как `'\x1B'`. Послать его можно в любой функции вывода, переключив поток вывода с экрана на принтер, или непосредственно в принтерный порт с помощью функции

```
int biosprint(int cmd, int byte, int port);
```

где `cmd`: 0 - печать символа `byte`, 1 - инициализация порта принтера, 2 - чтение статуса принтера;

`byte` - от 0 до 255 (что выводим/посылаем на принтер).

`port` - определение принтерного порта: 0 - LPT1; 1-LPT2 и

```
intbyte='\x1B';
```

```
biosprint(0,byte,0);
```

24-игольчатые (или LQ) принтеры включают (расширяют) язык управления 9-игольчатых принтеров. Большинство струйных принтеров по языку управления совместимо с LQ-принтерами.

Среди лазерных принтеров наиболее распространены HP LaserJet фирмы Hewlett Packard. Они управляются языком PCL. Большинство лазерных принтеров других фирм тоже поддерживает этот язык. Для выделения управляющей информации также используются Esc-последовательности, но кодовая строка длиннее, т.к. эти принтеры предоставляют больше возможностей по управлению.

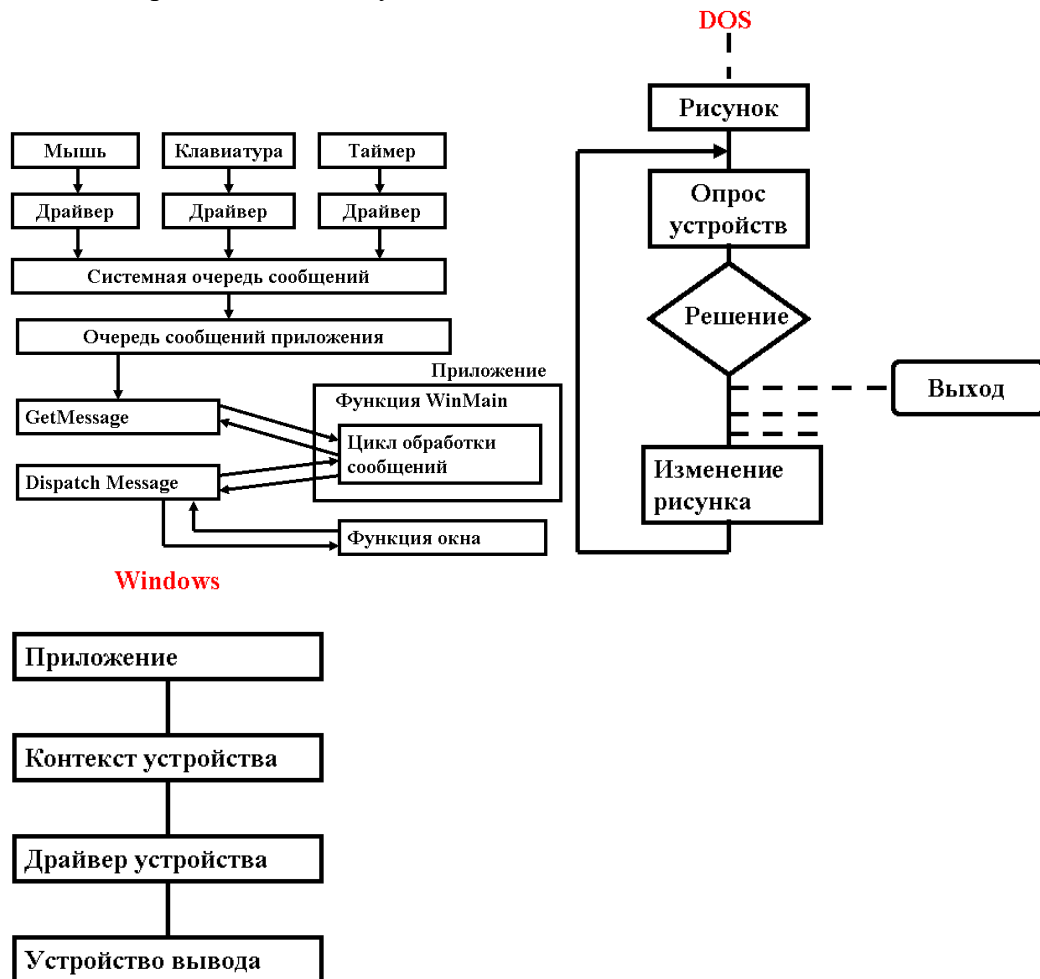
ВОПРОС 11. ОСНОВНЫЕ ОТЛИЧИЯ В ПОДХОДАХ MS DOS И WINDOWS ПРИ РАЗРАБОТКЕ ГРАФИЧЕСКИХ ПРИЛОЖЕНИЙ.

Операционная система Windows выводит графику посредством интерфейса графических устройств GDI - Graphic Device Interface. Текст тоже рассматривается как графика. GDI обеспечивает вывод на экран, принтер, плоттер и др. GDI избавляет приложения Windows от необходимости учитывать многие особенности устройств вывода. Например, как мы уже видели в DOS, адресация видеопамати адаптеров CGA, EGA, VGA, SVGA выполняется по-разному. К тому же представление видеоданных сильно зависит от видеорежима (разное количество байт на пиксел в зависимости от mode). Однако приложения Windows не работают непосредственно с видеопамтью. Вызываются соответствующие функции GDI реализованного в виде DLL. Функции GDI также не работают с аппаратурой. Для выполнения нужной графической операции GDI вызывает драйвер устройства вывода, ориентированный на особенности аппаратуры.

Таким образом; GDI позволяет организовать вывод на некоторое логическое устройство. Функции GDI и драйверы обеспечивают независимость приложений от аппаратуры. Это многие рассматривают как преимущество перед MS DOS, т.к. в MS DOS для повышения производительности приходится работать непосредственно с регистрами видеоконтроллера и видеопамятью (как это рассматривалось выше). С другой стороны, недоступность операций низкого уровня не позволяет нам влиять на производительность и отдаёт решение этих вопросов только на откуп создателей соответствующих системных программных средств.

Логические устройства дают программисту большую свободу выбора выразительных средств. Например, логический видеомонитор имеет огромное разрешение, способность отображать практически любой цвет (Реально - до 16 млн. цветов). Если задан цвет для палитры в 16 млн. цветов, а устройство не имеет такой возможности, GDI выбирает наиболее близкий к требуемому цвет. Для монохрома - градации серого.

Ситуация, когда приложение запрашивает у Windows одно, а получает другое, возникает и при работе со шрифтами. Это повышает независимость от аппаратуры по сравнению с MS DOS. В MS DOS при работе с видеоадаптерами указываются конкретные цвета, из файлов загружаются конкретные шрифты. Для новых устройств в программу вносятся изменения (хотя бы в связи с новыми именами файлов). Следовательно, программы MS DOS более аппаратно зависимы. Приложения Windows не меняются при смене аппаратуры, требуется только соответствующий драйвер. Чем лучше аппаратура, тем ближе цвет и шрифт будут к запрошенным. Таким образом, в MS DOS можно запросить только то, что имеется, в WINDOWS предложен другой подход к разработке приложений: запрашивай максимум того, что надо.



ВОПРОС 12. ОСНОВНЫЕ ЭТАПЫ ПРЕОБРАЗОВАНИЯ И МОДЕЛИ, ИСПОЛЬЗУЕМЫЕ ПРИ ПЕРЕХОДЕ ОТ ИЗОБРАЖЕНИЙ РЕАЛЬНОГО МИРА К КОМПЬЮТЕРНЫМ.

Для описания графической информации используется иерархическая совокупность математических моделей. В зависимости от формы представления существует 4 основных модели графических изображений:

- Ми (M0) - исходное (непрерывное) изображение;
- Мр (M1) - растровая модель (изображение);
- Мв (M2) - векторная модель (изображение);
- Мп (M3) - прикладная модель (описание изображения в терминах

хранения).

Технология преобразования графических представлений - отображение Q: Ми— Мп. Технологический процесс - отображение P_i, осуществляющее получение i модели по (i-1)-ой:

$$P_i: M(i-1) \rightarrow M_i$$

Технологическая операция (этап) - отображение T(i,j). внутри осуществляющее преобразование одной модели: T_{i,j}: M(i,j-1) → M_{i,j}

Укрупненно технология преобразования изображения из начальной модели в конечную:

Исходная модель (Ми)	
Процесс сканирования	Задание дискретности, координат и др. Распознавание цвета ... Запись растра на МД (Мр)
Растровая модель (Мр)	
Процесс растр-векторного преобразования.	Считывание с МД Фильтрация шумов Выделение средних линий объектов (выделение скелетов) Выделение контуров объектов Вычисление характеристик объектов Выделение производных элементов Формирование векторной модели (Мв)
Векторная модель (Мв)	
Процесс распознавания и формирования хранимого вида	Разделение объектов Распознавание логических объектов (возможно с точностью до графического примитива) Формирование структуры хранения
Прикладная модель (Мп)	

ВОПРОС 13. ОСНОВНЫЕ ЭТАПЫ РАСТР-ВЕКТОРНОГО ПРЕОБРАЗОВАНИЯ ГРАФИЧЕСКИХ ОБЪЕКТОВ.

Процесс сканирования зависит от принципов работы сканера. Поэтому его не рассматриваем. Наиболее объемным и важным является второй этап, называемый также оцифровкой изображения.

Бинарные изображения более просты. По сравнению с полутоновыми изображениями они имеют ограниченный класс шумов. Основной метод подавления шумов - логическая фильтрация. Наиболее часто встречающийся вид искажений графических изображений - неоднородность формы линий. Это может выражаться в

- изменении толщины (наличии слишком широких или узких участков линий);
- наличии изолированных черных пятен небольших размеров или изолированных пустот внутри линий;

- разрывах линий, слиянии нескольких линий (встречается редко, устранение трудоемко).

Основным фильтром, используемым для подавления шумов бинарных изображений, является сглаживающий фильтр, изменяющий значение центрального элемента окрестности в зависимости от количества однотипных с ним элементов в заданной окрестности. Алгоритм последовательного просмотра растра и применения каждому элементу данного фильтра называется алгоритмом логического сглаживания.

Выделение средних линий (скелетизация) позволяет описать геометрические особенности объекта, что удобно при последующей обработке. Существует два типа алгоритмов выделения средней линии.

1. Утоньшение, которое можно рассматривать как итерационное преобразование множества, имеющего непустую внутренность, в множество единичной ширины поперечного сечения путем последовательного его сжатия с обеих сторон.

2. Выделение скелета или средних осей объекта, т.е. выделение всех точек, равноудаленных по меньшей мере от двух точек на границе объекта. Методы выделения контуров условно можно разделить на следующие группы:

- выделения перепадов яркости;
- отслеживания (или обхода) контуров;
- сканирующие.

В методах первого класса в окрестностях каждой точки вычисляют градиент перепада яркости. Точки резкого изменения градиента выделяются как контурные. Так строится контурная модель, часто состоящая из незамкнутых штрихов. Эти методы используют, в основном, для полутоновых и цветных изображений. На основании такой модели очень трудно описать форму объекта. Поэтому часто исходные изображения сводятся к бинарным. На последних в основном используются методы двух других классов.

Методы отслеживания наиболее проработаны и просты в реализации. Обычно сначала выделяются границы, потом выполняется их аппроксимация. Это требует больших затрат памяти и времени. Более универсальный подход - совмещение отслеживания и аппроксимации. Разработаны специальные алгоритмы.

Сканирующие методы позволяют выделять контуры объектов при однократном просмотре исходного изображения. Для этого используются описания двух соседних строк изображения, списковые структуры.

Ряд характеристик объекта удобнее вычислять при растровом представлении. Это площадь объекта, его длина, периметр, центр симметрии дискретного объекта, количество объектов изображения. Разработаны математические методы вычисления этих и других характеристик.

Выделение производных элементов производится на основе скелетированного изображения. Фактически данный этап сводится к построению некоторой структуры данных, элементами которой являются выбранные производные элементы и связи между ними. Разработаны различные способы представления скелетированных изображений, в специальной литературе даны сведения об их применимости.

Растровый формат содержит больше информации о взаимоположении объектов изображения. Это полезно при их распознавании. Векторный формат более экономичен по памяти, более удобен в обработке. Чтобы соединить достоинства обоих форматов из растрового представления извлекают характеристики, используемые для распознавания данных и помещают их в структуру хранения. Возможно, что при этом множество всех характеристик объекта будет избыточным. Сейчас перевод растрового представления в векторное осуществляется не всегда, т.е. формирование векторного представления и связанные с ним операции пропускаются.

Третий этап преобразования (Мв—Мп) выполняется с помощью достаточно разработанной теории распознавания образов. Однако существует разрыв между теорией

и практикой. Часто в конкретных системах используются технические решения, ориентированные на тот класс объектов, с которым данная система работает. Что касается структур хранения, то современные системы управления базами данных (СУБД) позволяют хранить так называемые BLOB (binare large objects). Конкретный вид бинарной информации при этом значения не имеет. Наиболее распространенные форматы представления изображений будут рассмотрены далее.

До сих пор рассматривался процесс получения машинного вида изображения, введенного в компьютер извне. Чтобы создавать изображения некоторого класса на компьютере, требуется другой подход. Рассмотрим общие принципы построения таких моделей графических объектов. Процесс построения делится на 6 этапов:

1. Анализ структуры объекта по принципу иерархии входящих в него элементов с выделением базовых, которые нецелесообразно расчленять далее по соображениям целостности элементов или по характеру задач;

2. Построение математических моделей базовых элементов $\{M_{эi}\}$ и представление их в памяти ЭВМ;

3. Анализ структуры объекта с фиксацией всех связей между элементами;

4. Объединение математических моделей базовых элементов для построения математической модели объекта (изображения): $M_{и} = \{M_{эi}\}$;

5. Дополнение математической модели системными (системообразующими) параметрами $\{SP\}$, характеризующими объект как систему взаимосвязанных элементов. Примеры параметров: размеры, определяющие взаимное положение элементов, сведения о предельных отклонениях, условиях сопряжения и т.д.;

6. Объединение в группы GP одинаковых параметров математических моделей элементов с целью минимизации общего объема сведений в математической модели объекта M.

Таким образом, математическую модель объекта в общем случае можно представить совокупностью математических моделей элементов, системных параметров и групп параметров: $M_{и} = \{\{M_{эi}\}, \{SP\}, \{GP\}\}$.

К построенной по таким принципам математической модели предъявляются требования:

1. Простота и компактность представления объекта с целью минимизации памяти;

2. Возможность редактирования (дополнение другими элементами, деформирование, выделение сегмента и т.д.), перехода от одних единиц измерения к другим, решения наиболее распространенных задач, связанных с объектом (определение поперечных сечений, моментов инерции, центра тяжести и т.д.);

3. Соответствие представления, объекта возможностям технических средств вывода информации или его преобразование к виду, удовлетворяющему требованиям такого представления.

Все перечисленное справедливо, в основном, для векторных моделей, которые широко используются в инженерной практике. Для фотореалистических изображений некоторые требования не нужны, более существенны другие требования, которые будут рассмотрены позже.

ВОПРОС 14. ПОНЯТИЕ АФФИННЫХ ПРЕОБРАЗОВАНИЙ И ИХ ПРИКЛАДНОЕ ЗНАЧЕНИЕ ДЛЯ ЗАДАЧ КОМПЬЮТЕРНОЙ ГРАФИКИ.

Преобразования на плоскости и в пространстве

В соответствии с дискретным принципом работы ЭВМ решение любой задачи разбивается на некоторую последовательность шагов (этапов), образующих алгоритм. В компьютерной графике также выделяются элементарные этапы, из которых составляется графический алгоритм (графические преобразования).

Основу многих операций компьютерной графики составляют так называемые аффинные преобразования. Греческое слово АФФИНИС означает родственный. Аффинная геометрия - раздел геометрии, изучающий свойства фигур на плоскости и в пространстве, сохраняющиеся при любых аффинных преобразованиях, т.е. инвариантных относительно таких преобразований. Аффинные преобразования обеспечивают точечное взаимно однозначное отображение плоскости или пространства на себя, при котором 3 точкам, лежащим на одной прямой, соответствуют 3 точки, также лежащие на одной

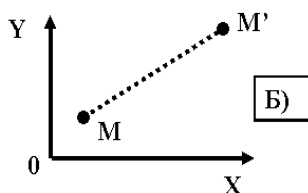
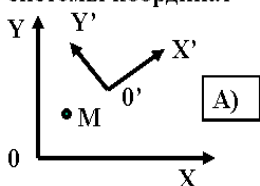
$$M(x,y) \longrightarrow M'(x' = ax + by + c, y' = dx + ey + f),$$

где a, b, c, d, e, f – произвольные числа, связанные неравенством

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0$$

А) сохраняется точка, изменяется система координат

Б) изменяются координаты точки относительно неизменной системы координат



прямой. Аффинные преобразования переводят пересекающиеся прямые в пересекающиеся прямые, параллельные прямые в параллельные прямые. Плоскость аффинно отображается на некоторую плоскость. Существует множество аффинных преобразований. К ним относятся преобразования подобия, сдвиги, сжатия и др. Следовательно, прорисовываются элементарные операции компьютерной графики,

не нарушающие геометрических свойств отображаемого объекта.

Одной из наиболее простых операций преобразования является пересчет координат точки или перенос системы координат. Пусть имеется точка с координатами (x, y) . Поставим этим координатам в соответствие координаты

Этот случай можно трактовать двояко:

А) сохраняется точка, изменяется система координат

Б) изменяются координаты точки относительно неизменной системы координат, т.е. формулы задают отображение, точки $M(x, y)$ в точку $M'(x', y')$ в той же координатной системе.

Обычно принимается вторая трактовка, т.к. мы работаем в стабильных координатах устройства.

В принципе можно строить изображения в любой системе-координат: прямолинейной и криволинейной, прямоугольной и непрямоугольной. Для простоты и в соответствии с привычным представлением будем рассматривать представление в прямоугольной декартовой системе координат.

В машинной графике часто используется матричное представление. Точку можно представить с помощью вектор-столбцов для плоскости $|x \ y|$, пространства $|x \ y \ z|$. Тогда преобразования точек сводятся к операциям над матрицами, что хорошо соответствует возможностям вычислительной техники. В общем виде задача выглядит так. Даны матрицы A и B и задана их взаимосвязь $AT = B$. Необходимо найти матрицу преобразования. Решением является $T = A^{-1}B$, где A^{-1} - обратная от квадратной матрицы A . Матрица T - фактически коэффициентов

$$\begin{vmatrix} a & b \\ d & e \end{vmatrix} \quad \boxed{\begin{matrix} x' = ax + by + c, \\ y' = dx + ey + f \end{matrix}} \quad (1)$$

Её можно трактовать и как оператор. Тогда перемножение матриц используется для того, чтобы выполнить геометрическое преобразование над системой точек, представленных с помощью векторов положения отдельных точек, содержащихся в матрице A . Интерпретация матричного умножения как геометрического оператора является основой математических преобразований в машинной графике.

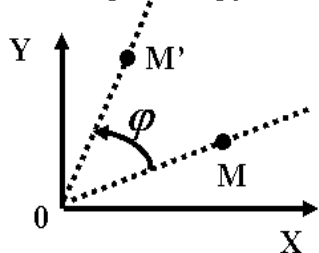
В аффинных преобразованиях особую роль играют несколько частных случаев, комбинация которых позволяет описать любое преобразование одной точки в другую в соответствии с формулами (1).

Фактически эти случаи описывают конкретный вид матрицы T, т.е. дают коэффициенты преобразования исходных координат точки. Последовательность выполнения этих преобразований - алгоритмы машинной графики.

Начнем рассмотрение элементарных операций с плоскостных. Можно выделить 4 таких операции:

поворот, растяжение (сжатие), отражение, перенос.

1. Поворот вокруг начала координат на угол φ



$$x' = x \cos \varphi - y \sin \varphi$$

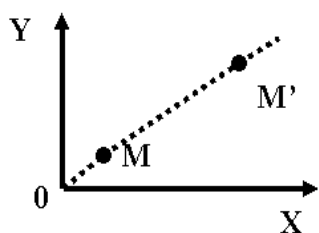
$$y' = x \sin \varphi + y \cos \varphi$$

$$\begin{vmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{vmatrix} \begin{vmatrix} 0 \\ 1 \end{vmatrix} \begin{matrix} -1 \\ 0 \end{matrix}$$

Матрица преобразования T при этом имеет вид.

Для частного случая поворота на 90° (поворот против часовой стрелки относительно начала координат) вид матрицы $\begin{vmatrix} 0 & -1 \\ 1 & 0 \end{vmatrix}$

2. Растяжение (сжатие) вдоль координатных осей



$$\begin{matrix} x' = a * x \\ y' = e * y \end{matrix} \begin{vmatrix} a & 0 \\ 0 & e \end{vmatrix}$$

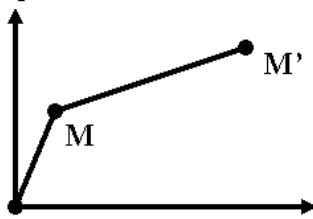
Растяжению можно сопоставить положительные, сжатие - отрицательные значения коэффициентов.

Другое название операции - изменение масштаба. Как частный случай изменения масштаба можно рассматривать сдвиг вдоль одной из координатных осей. Эти операции перекрываются с ниже рассмотренными операциями зеркального отражения относительно координатных осей.

3. Отражение относительно осей

Вид матрицы	Действие	Рисунок
$\begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix}$	Положение точки не меняется	
$\begin{vmatrix} a & 0 \\ 0 & 1 \end{vmatrix}$	Изменение положения по оси X	
$\begin{vmatrix} 1 & 0 \\ 0 & e \end{vmatrix}$	Изменение положения по оси Y	
$\begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}$	Зеркальное отражение относит. оси Y	
$\begin{vmatrix} 1 & 0 \\ 0 & -1 \end{vmatrix}$	Зеркальное отражение относит. оси X	
$\begin{vmatrix} -1 & 0 \\ 0 & -1 \end{vmatrix}$	Зеркальное отражение относит. начала координат	

4. Перенос

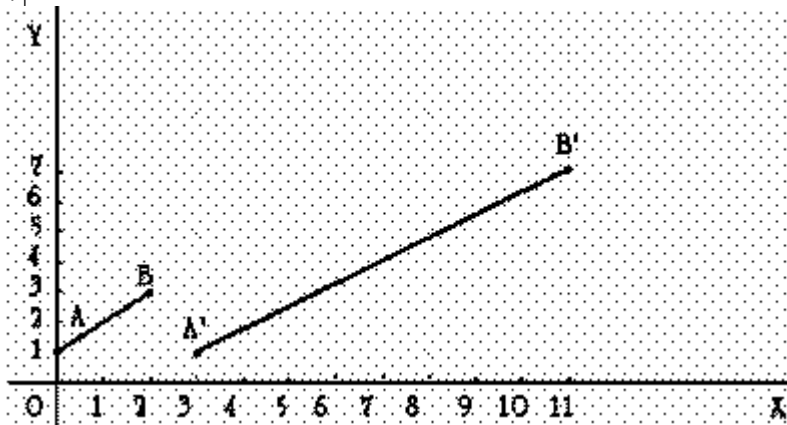


$$\begin{aligned}x' &= a * x + c; \\y' &= e * y + f\end{aligned}$$

В курсе аналитической геометрии доказывается, что любое преобразование вида (1) всегда можно представить как суперпозицию простейших 4 преобразований, рассмотренных выше.

ВОПРОС 15. ЭЛЕМЕНТАРНЫЕ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ НА ПЛОСКОСТИ, СОСТАВЛЯЮЩИЕ БАЗИС ОПЕРАЦИЙ МАШИННОЙ ГРАФИКИ.

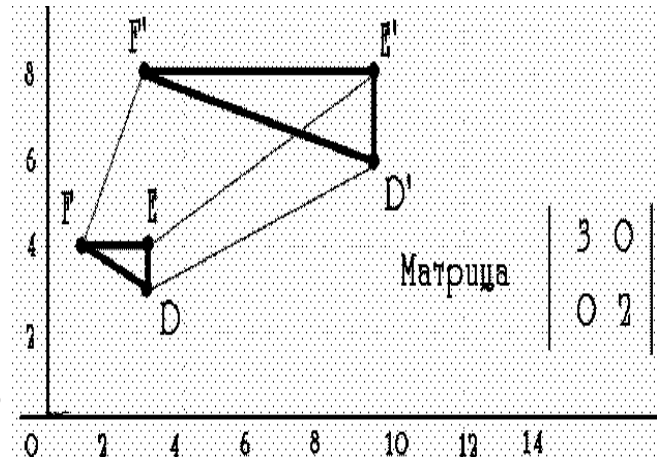
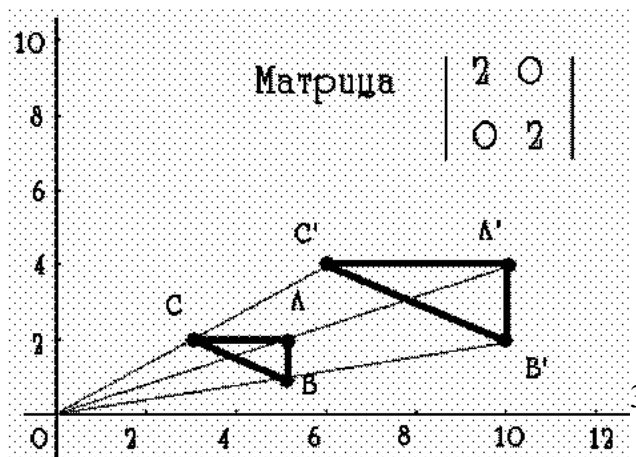
Выше было рассмотрено преобразование точек. Т.к. прямая задается координатами двух ее точек, для операций над прямой также можно использовать умножение матриц. Пусть имеется прямая с координатами концов, заданными векторами $A[0,1]$ и $B[2,3]$. Возьмем произвольно матрицу преобразования T . Эта матрица осуществляет растяжение (или сдвиг). Выполним операцию умножения для точек A и B . $A'=A*T=[3\ 1]$; $B'=B*T=[11\ 7]$



Матрица размером $2*2$ преобразует любую прямую в другую прямую. Точки результирующей линии взаимно однозначны точкам исходной линии (аффинные преобразования). Можно доказать, что данное утверждение распространяется не только на конечные, но и на серединные и все другие точки отрезка прямой. Для машинной

графики существенно, что положение любой прямой линии преобразуется в новое положение преобразованием ее граничных точек и последующим проведением линии между полученными точками. Доказано, что использование матрицы $2*2$ для преобразования параллельных линий не нарушает параллельности. Как следствие, в результате действия такой матрицы параллелограмм преобразуется в другой параллелограмм. При преобразовании пересекающихся прямых точка пересечения исходной пары линий преобразуется в точку пересечения результирующей пары.

Таким образом, матричное умножение можно использовать для графических построений. Например, можно равномерно или неравномерно менять масштаб фигуры. Если на вершины треугольника воздействует матрица $\begin{vmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \end{vmatrix}$ - , координаты увеличиваются в 2 раза. Если члены матрицы не равны, фигура искажается.



Применение общего матричного преобразования к единичному квадрату с одним углом в начале координат порождает параллелограмм

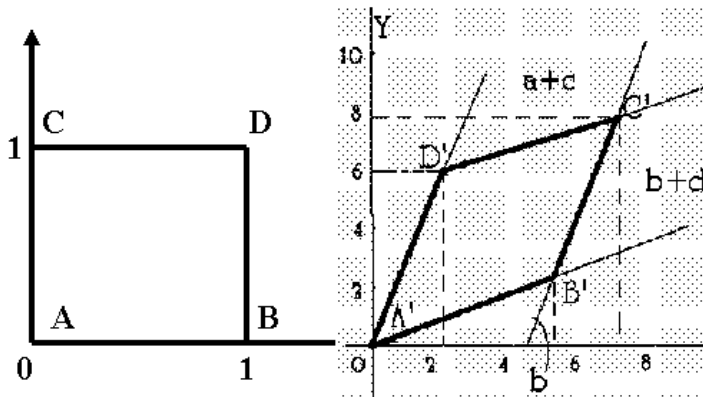
Т.к. конкретный вид матриц преобразования всегда известен, можно посчитать площадь полученной фигуры на основе площади исходной фигуры.

ВОПРОС 16. ПОНЯТИЕ И ПРИКЛАДНОЕ ЗНАЧЕНИЕ ОДНОРОДНЫХ КООРДИНАТ.

Применение матрицы общего вида 2*2 к началу координат дает результат:

$$[0 \ 0] * \begin{vmatrix} a & b \\ d & e \end{vmatrix} = [0 \ 0] = [x' \ y']$$

$$\begin{array}{l} A \rightarrow \\ B \rightarrow \\ C \rightarrow \\ D \rightarrow \end{array} \begin{vmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{vmatrix} \times \begin{vmatrix} a & b \\ d & e \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ a & b \\ a+d & b+c \\ d & e \end{vmatrix} \begin{array}{l} \leftarrow A' \\ \leftarrow B' \\ \leftarrow C' \\ \leftarrow D' \end{array}$$



Таким образом, начало координат инвариантно относительно общего преобразования 2x2. Это является ограничением, которое преодолевается с помощью однородных координат.

Пусть M - произвольная точка плоскости с координатами (x,y), вычисленными относительно заданной координатной системы. Однородными координатами этой точки называется любая тройка одновременно неравных нулю чисел x1, x2, x3, связанных с заданными числами соотношением: x1/x3=x; x2/x3=y

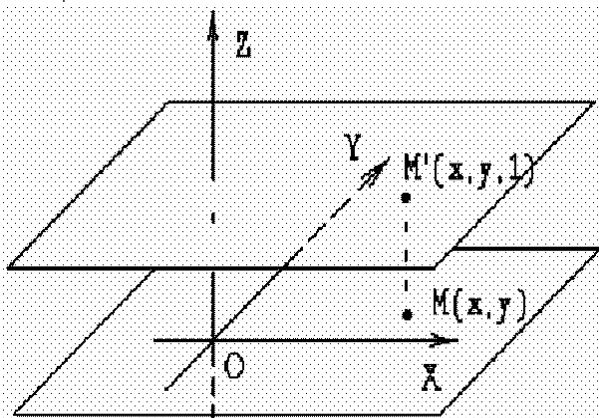
Фактически, однородные, координаты дают смещение плоскости, на которой рассматривается фигура, относительно начала координат или, в другой интерпретации, смещение начала координат. Это, по существу, переход к 3-мерному представлению плоских объектов, частный случай пространственного представления.

$$\begin{array}{l} x1 / x3 = x; \quad x2 / x3 = y \\ [x \ y \ 1] \quad [x' \ y' \ 1] \end{array} \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ m & n \end{vmatrix} \begin{array}{l} \\ \\ \\ \end{array} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ m & n & 1 \end{vmatrix}$$

$$[x \ y \ 1] * \begin{vmatrix} 1 & 0 \\ 0 & 1 \\ m & n \end{vmatrix} = [x+m \ y+n] = [x' \ y']$$

Рассмотрим более подробно. Введем в 2-мерное представление 3 компонент, равный единице. Тогда вектора будут иметь вид: $[x \ y \ 1]$; $[x' \ y' \ 1]$. Матрица преобразования примет вид: $|10 \ 01 \ mn|$, так как для выполнения умножения матриц число столбцов, описывающих точку, должно равняться числу строк в матрице преобразования. $[x \ y \ 1] * |10 \ 01 \ mn| = [x+m \ y+n] = [x' \ y']$.

Матрица $3*2$ не квадратная, поэтому она не имеет обратной матрицы. (Вспомним: даны матрицы A и B , задана их взаимосвязь $AT = B$. Требуется найти матрицу преобразования. Решением является $T = A^{-1} B$, где A^{-1} - обратная от квадратной матрицы A . Матрица T - фактически матрица коэффициентов можно трактовать и как оператор.) Чтобы получить квадратную обращаемую матрицу преобразования, дополним ее $|100 \ 010 \ mn1|$



Третья (единичная) компонента векторов точек не меняется при добавлении элементов.

Полученные результаты можно трактовать как поднятие плоскости, в которой мы работаем, на 1 по оси z.

В более общем случае координаты точки можно представить как (hx, hy, h) , $h \neq 0$. Следовательно, поднимаем плоскость на h . Практический смысл этого связан с изменением масштаба при вписывании изображения в координаты устройства

(мировые координаты - нормированные координаты - координаты устройства). Как правило, координаты устройства являются целыми числами (пиксели), следовательно, точку с координатами $(0.5, 0.1, 2.5)$ представить нельзя. При $h=10$. получаем $(5, 1, 25)$. Другой случай - координаты $(80000 \ 40000 \ 1000)$, что создает угрозу арифметического

1. Вращение (rotation)	$[R] = \begin{vmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{vmatrix}$	2. Растяжение/сжатие (dilatation)	$[D] = \begin{vmatrix} a & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & 1 \end{vmatrix}$
3. Отражение зеркальное (reflection)	$[M] = \begin{vmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	4. Перенос (translation)	$[T] = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & f & 1 \end{vmatrix}$

переполнения. $h=0,001$ дает $(80 \ 40 \ 1)$.

Матрицы третьего порядка позволяют описать любое аффинное преобразование на плоскости. Обычно задачу разбивают на этапы и пользуется 4 элементарными операциями. Матрицы 3 порядка для этих случаев

Как использовать эти преобразования? Сначала разбивают действие на элементарные преобразования из 4 приведенных выше, получают матрицы. Потом перемножают эти матрицы в порядке получения. Результирующая матрица является матрицей преобразования.

17. ЭЛЕМЕНТАРНЫЕ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ В ПРОСТРАНСТВЕ, СОСТАВЛЯЮЩИЕ БАЗИС ОПЕРАЦИЙ МАШИННОЙ ГРАФИКИ.

Для пространства однородные координаты выглядят как: $(x,y,z,1)$ или, в более общем случае $-(hx,hy,hz,1)$ Преобразование осуществляется аналогично плоскости по формуле $AT = B$, но матрица преобразования T имеет размер 4×4 для однородных координат.

Аналогично плоскости любое аффинное преобразование в пространстве представимо как суперпозиции вращений, отражений, растяжений и переносов.

Аффинные преобразования в пространстве

$$(x, y, z, 1) \quad (hx, hy, hz, 1) \quad h \neq 0$$

1. Вращение вокруг:
а) оси абсцисс на угол φ $[R_x] = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \varphi & \sin \varphi & 0 \\ 0 & -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

б) оси ординат на угол ψ $[R_y] = \begin{vmatrix} \cos \psi & 0 & -\sin \psi & 0 \\ 0 & 1 & 0 & 1 \\ \sin \psi & 0 & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

3. Отражение (зеркальное) относительно:
а) плоскости XY (изменяется знак координаты Z) $[M_z] = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

б) плоскости YZ $[M_x] = \begin{vmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

в) плоскости ZX $[M_y] = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

в) оси аппликат на угол χ $[R_z] = \begin{vmatrix} \cos \chi & \sin \chi & 0 & 0 \\ -\sin \chi & \cos \chi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

2. Растяжение (сжатие) или изменение масштаба $D = \begin{vmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$

$a > 0, b > 0, c > 0$ – коэффициенты изменения масштаба относительно осей X, Y, Z (растяжение).

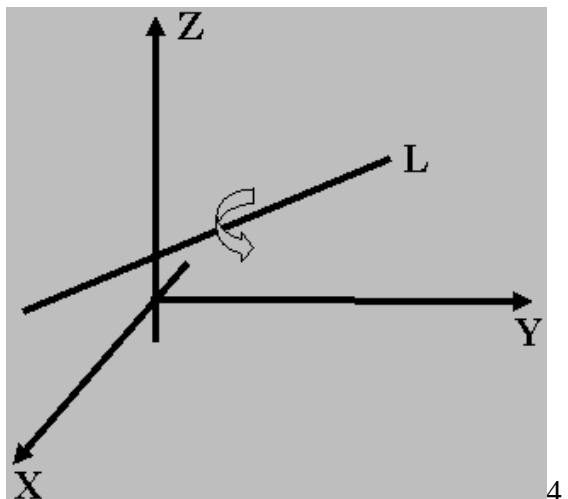
4. Перенос (координаты X на k , y на m , Z на n , $(k \ m \ n)$ – вектор переноса) $[T] = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ k & m & n & 1 \end{vmatrix}$

Пример: выполнить вращение вокруг произвольной оси.

Решение:

- а) путем переноса сдвинуть ось вращения так, чтобы она проходила через начало координат;
- б) поворот (возможно, неоднократный) оси вращения на некоторый угол до совмещения с одной из координатных осей;
- в) поворот вокруг оси;
- г) перенос, обратный п. б)

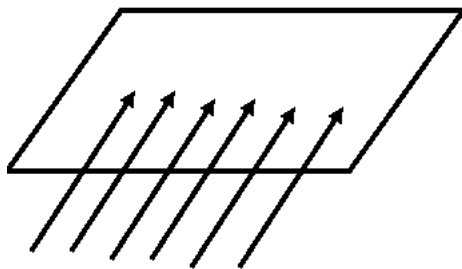
1. Определитель (детерминант) любой матрицы вращения равен 1. Т.к. вращения описываются умножением матриц, трехмерные вращения некоммутативны, т.е. порядок умножения влияет на конечный результат.



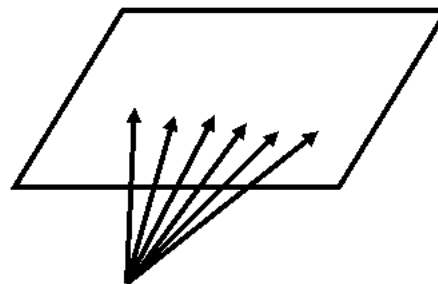
ВОПРОС 18. ОСНОВНЫЕ ВИДЫ ПРОЕКЦИЙ И СООТВЕТСТВУЮЩИЕ ИМ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ.

Геометрическая операция проектирования (проецирования) выполняет отображение объектов на картинной плоскости. При этом через каждую точку объекта проводятся прямые, начинающиеся в некотором центре. Точки пересечения этих прямых с плоскостью изображения образуют проекцию.

Различают проектирование параллельное и центральное. При центральном проектировании все прямые исходят из одной точки - центра пучка К центральным относятся перспективные проекции (прямая - обычная и обратная - иконы). При параллельном проектировании точка центра пучка находится в бесконечности. Каждый из двух основных классов разбивается на несколько подклассов в зависимости от взаимного расположения картинной плоскости и координатных осей.



Параллельное проектирование



Центральное проектирование

КЛАССИФИКАЦИЯ ПРОЕКЦИИ

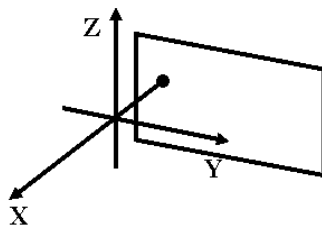
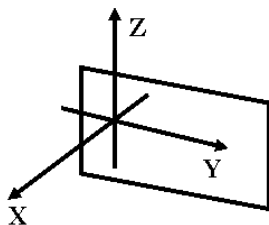
* Параллельные проекции:

1. Ортографическая проекция
2. Аксинометрическая Проекция
 - 2.1. Триметрическая проекция
 - 2.2. Диметрическая проекция .
 - 2.3. Изометрическая Проекция
3. Косоугольная проекция
 - 3.1 Свободная проекция,
 - 3.2. Кабинетная проекция

* Перспективные проекции:

- 1.Одноточечная проекция,
- 2.Двухточечная проекция,
3. Трехточечная проекция

При ортографической проекции картинная плоскость совпадает с одной из координатных плоскостей или параллельна ей. Матрица проектирования вдоль оси x на плоскость yz имеет вид:



Матрица проектирования вдоль оси X на плоскость yz

$$[P_x] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p & 0 & 0 & 1 \end{bmatrix}$$

Матрицы для других осей координат

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & q & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & r & 1 \end{bmatrix}$$

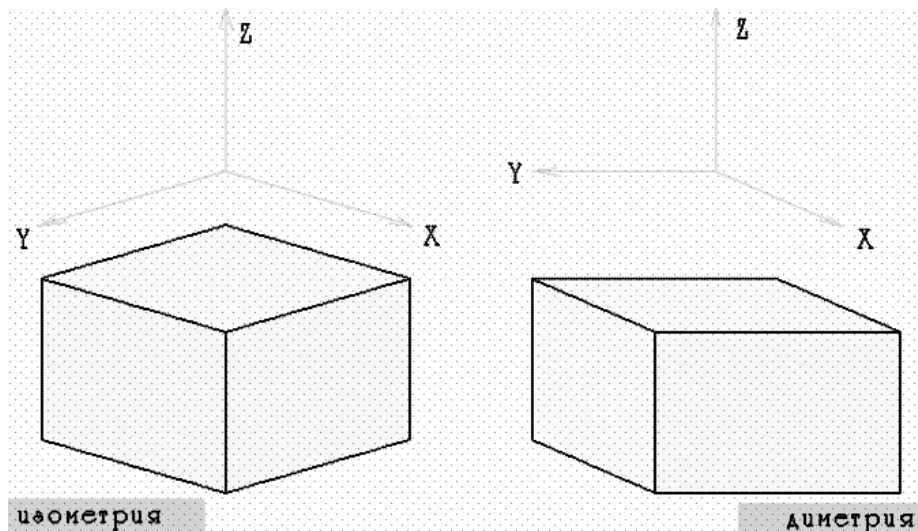
Если плоскость проектирования параллельна координатной плоскости, матрица $[P_x]$ умножается на матрицу сдвига

При аксонометрической проекции проектирующие прямые перпендикулярны плоскости отображения. Предмет вместе со

связанной с ним системой координат проектируется на некоторую плоскость. В соответствии, с взаимным расположением плоскости проектирования и координатных осей различают проекции:

- триметрию (плоскость находится относительно осей координат под различными углами для каждой оси) - используется редко;
- диметрию (два из трех углов между осями и плоскостью равны, т.е. две оси одинаково сокращены);
- изометрию (все углы равны, т.е. масштаб по всем осям одинаков).

Каждая из этих проекций получается комбинацией поворотов, после чего следует параллельное (ортографическое) проектирование. Можно вывести соответствующие матрицы, некоторые подходы обозначены в [1].



Косоугольные проекции используют пучок прямых, не перпендикулярных плоскости отображения. Выделяют два вида косоугольных проекций:

- свободную (угол наклона проектирующих прямых к плоскости отображения равен 45°);
- кабинетную

(частный случай свободной, масштаб по третьей оси в 2 раза меньше).

Все это также представимо в матричном виде.

Так, при проектировании на плоскость ху точки с координатами $x=0, y=0, z=1$ имеем $(0\ 0\ 1\ 1) \rightarrow (a\ b\ 0\ 1)$.

Вид матрицы;

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a & b & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Для свободной проекции $a=b=\cos \pi / 4$.

Для кабинетной проекции $a=b=0.5 * \cos \pi / 4$

Перспективные (центральные) проекции более сложны. Они получаются путем перспективного преобразования и проектирования на некоторую плоскость наблюдения. Вводятся параметры, описывающие центр проектирования, т.е. точку, откуда "смотрят" на объект. Пусть эта точка имеет координаты $(0\ 0\ c)$. Тогда матрица перспективного преобразования в пространстве имеет вид:

Выкладки получения: Член матрицы, характеризующий местоположение центра проектирования, выводится из уравнения прямой, проведенной из центра проектирования в некоторую точку объекта. Если мы хотим, например, плоскость наблюдения совместить с плоскостью ху, матрица примет вид, значение в третьем столбце третьей строки отражает факт $z = 0$. В общем случае при нахождении центра

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1/c \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

проектирования на осях z или y или x матрицы преобразования будут иметь соответственно вид:

Это матрицы одноточечных преобразований.

1	0	0	-1/a
0	1	0	-1/b
0	0	1	-1/c
0	0	0	1

Для произвольного центра проектирования с координатами (a, b, c) матрица преобразования:

Из школьного курса черчения и курса начертательной геометрии известно, что по нескольким проекциям можно восстановить изображение. Для компьютерной графики это соответствует обратным матричным преобразованиям. Т.к. для восстановления изображения используются несколько проекций, преобразования весьма сложны и длительны. Этим объясняется высокая ресурсоемкость подобных задач.

Как все это реализовать (запрограммировать)? Здесь удобно использовать объектно-ориентированный подход. Фактически, мы работаем с векторами, т.к. координаты точки - 3 числа. Для класса ВЕКТОР (vector) можно определить систему операций (например, поэлементное сложение и вычитание векторов и т.д.). Определяются также основные функции. Аналогично вводится класс МАТРИЦА (matrix).

19. ГЕОМЕТРИЧЕСКИЕ СПЛАЙНЫ.

В машинной графике часто применяют приближенное изображение фигур. Плавные линии и гладкие поверхности подвергаются кусочно-линейной аппроксимации. Нередко возникает и задача восстановления изображений по известным точкам (сосканированным или полученным иным путем). Один из способов приближенного представления плоских и пространственных изображений - геометрические сплайны. Это отдельное направление компьютерной графики. Рассмотрим его обзорно, на уровне общих понятий.

Термин "сплайн" происходит от английского spline. Так называлась гибкая полоска стали, с помощью которой чертежники через заданные точки проводили плавные кривые. Используя теорию упругости, можно доказать, что результирующая кривая приближенно является кусочным кубическим многочленом. Он непрерывен и имеет непрерывные первую и вторую производные. Следовательно, кривая имеет постоянную кривизну и разрывы возникают лишь в третьей производной, что для человеческого глаза практически незаметно. Результирующая кривая или поверхность выглядит гладкой. Если вдоль сплайна совершается механическое движение непрерывность второй производной обеспечивает непрерывность ускорения и, следовательно, отсутствие резки изменений приложенной силы. Это очень важно при механической обработке. Поэтому сплайны используют например, при проектировании траектории движения режущего инструмента.

Первая работа по теоретическому исследованию сплайнов вышла в 1946 г. Сначала сплайны рассматривались как удобный инструмент приближения функций. Однако быстро обнаружилось множество сплайнов различных типов. Они стали использоваться в численных методах, системах автоматизированного проектирования (САПР), автоматизированных системах технологической подготовки производства (АСТПП) и т.д.

Сплайны позволяют заменить аналитическое описание кривых и поверхностей их построением по точкам. Задают несколько точек, лежащих на искомой кривой или поверхности, и через эти точки с помощью сплайнов проводят плавные кривые/поверхности (задача интерполяции). Используют сплайны и для сглаживания. Тогда линию или поверхность проводят вблизи выбранных точек. Возникает задача поиска требуемой сплайн-функции. Для кривых (на плоскости) это задача нахождения сплайн-функции одной переменной.

Пусть на плоскости заданы m точек с координатами (x_i, y_i) , $i = 0, 1, \dots, m$, причем $x_0 < x_1 < \dots < x_m$

(точки упорядочены по возрастанию абсциссы). Для этого набора требуется найти сглаживающую кривую, максимально приближенную к исходным точкам. Здесь возможны две крайности:

1. Кривую можно провести непосредственно через исходные точки. Тогда получаем интерполяцию с помощью ломаной. Достоинство: простота, постепенность построения (поэтапность). Недостаток: отсутствие гладкости. даже первая производная имеет разрывы в узлах;

2. Из курса математического анализа известно, что существует интерполяционный многочлен Лагранжа, график которого проходит через все заданные точки.

$$L_m(x) = \sum_{i=1}^m y_i \frac{\omega_m(x)}{(x-x_i)\omega_m(x_i)}$$

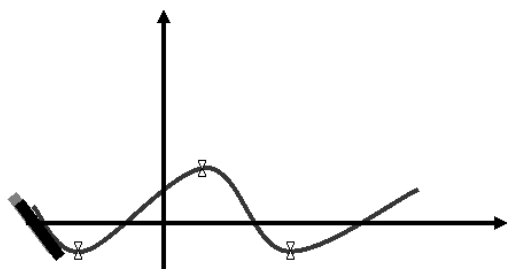
$$\omega_m(x) = \prod_{j=0}^m (x-x_j)$$

где (x, y) $i = 1 \dots m$ - заданные точки

Достоинство: относительная простота описания, удобство подсчета коэффициентов. Недостаток: изменение одной точки влечет пересчет всех коэффициентов

многочлена. Такое изменение весьма влияет и на вид кривой

Сплайны сохраняют достоинства и уменьшают недостатки обоих подходов. Они строятся с помощью многочленов (второй подход), но последовательно (первый подход). При этом важно определить степень каждого многочлена и подобрать его коэффициенты так, чтобы обеспечить гладкое сопряжение соседних звеньев,



Как это соотносится с происхождением названия "сплайны"? Если гибкую стальную линейку поставить на ребро, закрепить в крайней точке последовательности и разместить так, чтобы она проходила через остальные точки, получим некоторую функцию, описывающую профиль этой линейки.

Эта функция обладает свойствами:

- с довольно большой точностью часть графика этой функции, заключенного между любыми 2 соседними точками, можно считать многочленом третьей степени;
- на всем промежутке от начальной до конечной точки эта функция дважды непрерывно дифференцируема

Полученная функция называется интерполяционным кубическим сплайном.

Более точная формулировка:

ИНТЕРПОЛЯЦИОННЫЙ КУБИЧЕСКИЙ СПЛАЙН - функция $S(x)$, обладающая свойствами:

1. График функции $S(x)$ проходит через каждую точку заданного массива, т.е. $S(x_i) = y_i, i=0..m$;

На каждом из отрезков $[x_i, x_{i+1}] i=0,1,..m-1$ функция является многочленом третьей степени

3. На всем отрезке задания (x_0, x_m) функция $S(x)$ имеет непрерывную вторую производную.

Т.к. на каждом из отрезков $[x_i, x_{i+1}]$ сплайн $S(x)$ определяется 4 коэффициентами, для его полного построения необходимо найти $4*m$ чисел. Эти коэффициенты ищутся, исходя из условия 3, т.е. непрерывности сплайна во всех внутренних узлах, а также непрерывности первой и второй производной. Условия непрерывности накладывают ограничения на коэффициенты. Следовательно, для их определения получаем для всех

внутренних узлов $4 \cdot m - 2$ равенств. Недостающие граничные условия можно получить, задав, например, значения первых производных на концах интервала x_0 и x_m . Могут быть и граничные условия других типов. Таким образом, для определения коэффициентов решается система линейных уравнений.

Для изображений в пространстве нужно построить поверхность, проходящую через данные точки пространства. Используется функция 2 переменных, т.е. интерполяционный бикубический сплайн. ИНТЕРПОЛЯЦИОННЫЙ БИКУБИЧЕСКИЙ СПЛАЙН - функция 2 переменных $S(x,y)$, обладающая свойствами:

1. График функции $S(x,y)$ проходит через каждую точку заданного массива, т. $S(x_i, y_j) = Z_{ij}$, $i=0..m, j=0..n$;

2. На каждом из прямоугольников $[x_i, x_{j+1}] \cdot [y_j, y_{j+1}]$, $i=0..m-1, j=0..n-1$ функция является многочленом 3-ей степени по каждой из переменных:

3. На всем прямоугольнике задания $[x_0, x_m] \cdot [y_0, y_n]$ функция $S(x,y)$ имеет по каждой переменной непрерывную вторую производную.

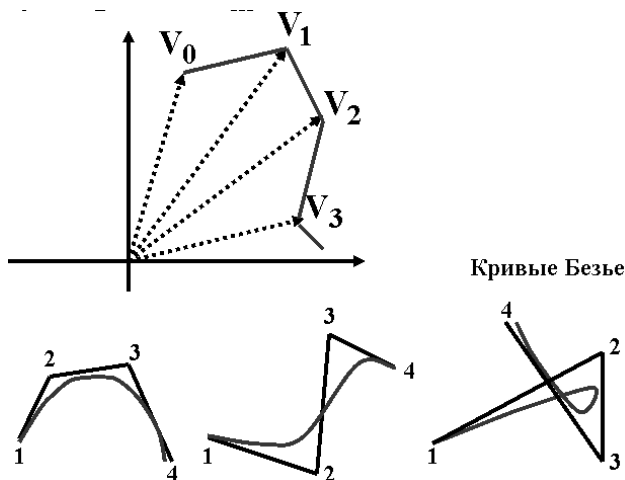
Количество коэффициентов бикубического сплайна равно $16 \cdot m \cdot n$. Как и для одномерного случая, для получения коэффициентов строится система линейных уравнений.

Достоинство сплайнов - относительная простота. Методы решения систем линейных уравнений хорошо известны. Недостаток: изменение хотя бы одной точки влечет большой объем пересчета коэффициентов. Если точки задаются приближенно, нет необходимости жесткого прохождения функции через каждую точку, можно использовать методы сглаживания. Это ведет к тому, что ослабляется требование однозначного проектирования искомой кривой на координатную ось или поверхности на координатную плоскость. Таким образом, кривая может быть замкнутой, самопересекающейся и т.д., возможно несколько проходов по одним и тем же точкам при формировании изображения, ослабляются требования к задаваемому массиву исходных точек.

Пусть в пространстве (или на плоскости) задан упорядоченный набор точек, определяемый векторами $V_0 - V_1 - \dots - V_m$. Ломаная $V_0 - V_1 - \dots - V_m$ называется контрольной ломаной, порожденной массивом $V = \{V_0, V_1, \dots, V_m\}$.

Если рассмотреть для начала наиболее простой двумерный вариант, то ставится задача нахождения сглаживающей кривой. В качестве одной из таких кривых может выступать кривая Безье.

Математически кривая Безье описывается полиномиальной функцией, которая



осуществляет интерполяцию между начальной и конечной точками интервала. Кривая Безье является гладкой и касается начального ($V_0 - V_1$) и конечного ($V_{m-1} - V_m$) отрезков контрольной ломаной. Каждый член полинома соответствует одной из точек интервала. Точки могут рассматриваться в любом порядке, а не только по возрастанию одной из координат). Следовательно, в зависимости от порядка точек получаются разные кривые. Пример - для 4 точек (кубическая кривая Безье):

Кроме достоинств (гладкость, известные, хорошо реализуемые методы расчета коэффициентов) кривые Безье имеют и общие для сплайнов недостатки:

- степень коэффициентов полинома на 1 меньше количества точек, т.е. зависит от количества точек
- изменение хотя бы одной точки приводит к изменению вида кривой, при добавлении/изменении хотя бы одной точки набора требуется пересчет всех коэффициентов.

На практике применяют построение кривой по фрагментам, выбирая, как правило, по 4 точки, т.е. пользуясь кубическими кривыми Безье. Чтобы фрагменты хорошо сопрягались, на исходные точки накладывается ряд условий. Например, необходимо, чтобы каждые 3 соседние точки лежали на одной прямой.

Как улучшить результаты, получаемые с помощью кривых Безье? В уравнении, задающем кривую Безье, векторные составляющие (координаты точек опорного массива) постоянны, пока мы их не захотели изменить. Следовательно, улучшение возможно за счет коэффициентов многочлена (изменяем способ их подсчета).

Класс сплайновых кривых гораздо шире кривых Безье. Кривые Безье относятся к классу полиномов Бернштейна (строятся в базисе Бернштейна). Существует другой базис, называемый В-сплайн базисом (от base - базовый), для которого базис Бернштейна является частным случаем. Здесь принят иной способ вычисления коэффициентов. Причем эти коэффициенты не зависят от координат конкретных точек четверки. Коэффициенты вычисляются по рекуррентным (рекурсия!) формулам. Поэтому изменение одной вершины исходного массива не влечет полный пересчет всех коэффициентов. Пересчитывается только некоторая часть из них (5 слагаемых). Здесь также ограничиваются обычно кубическими многочленами, т.е. используют кубические В-сплайны.

Часто возникает желание подправить построенную с помощью В-сплайнов кривую без изменения массива опорных точек. Для этого в уравнение кривой вводятся параметры. Таким образом получаются обобщение кубических В-сплайнов, называемые рациональными кубическими В-сплайнами и бета-сплайнами.

В рациональных кубических В-сплайнах каждый коэффициент умножается на вес (параметр формы) - неотрицательное число. Сумма этих чисел положительна (т.е. по крайней мере одно из чисел ненулевое). Если все веса равны между собой, получаем элементарную кубическую В-сплайновую кривую.

В бета-сплайнах весовые коэффициенты (в формулах для которых присутствует обозначение бета) ищутся исходя из условий гладкости в точках сопряжения частичных кривых. Математические формулы, обеспечивающие это свойство, известны (непрерывность функции, первой и второй производной). Следовательно, путем ряда математических преобразований ищутся коэффициенты многочленов, в состав которых входят 2 числовые параметра. Эти параметры называются параметрами формы бета-сплайновой кривой: (α_1 - параметр скоса. (α_2 - параметр напряжения ($\alpha_1 > 0$, $\alpha_2 \geq 0$)). При $\alpha_1 = 1$, $\alpha_2 = 0$ получается кубическая В-сплайновая кривая.

Для поверхностей добавляется еще одна координата. Следовательно, вместо контрольной ломаной получается контрольный многогранник, а точнее - опорный граф заданного массива точек. Построение ведется по элементарным фрагментам, которые задаются не 4, а 16 точками, т.к. имеем дело с плоскостями, составляющими контрольный многогранник. Сглаживающие поверхности строятся на тех же принципах, что и сглаживающие кривые: построение ведется по фрагментам, используются бикубические многочлены. Получаются бикубические поверхности Безье, бикубические В-сплайновые поверхности. Они наследуют основные свойства одноименных кубических кривых применительно к поверхностям.

20. АЛГОРИТМ БРЕЗЕНХЕМА.

подавляющее большинство графических устройств - растровые, т.е. изображение представляется в виде матрицы или решетки пикселей (растра). Это требует специальных алгоритмов аппроксимации изображений.

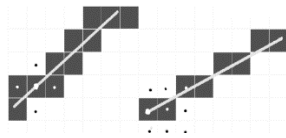
Растр можно представить решеткой, состоящей из единичных квадратов. Центр каждого квадрата - координаты соответствующего пиксела. Когда инициализируется точка с координатами (p,q), вся ее квадратная окрестность закрашивается каким-либо цветом.

Плавно изогнутые линии подвергаются кусочно-линейной аппроксимации, минимальный шаг - 1 пиксел. Возможность соединения двух пикселей растровой линией называется СВЯЗНОСТЬЮ. Растровая линия - суть последовательный набор соседних пикселей. Пусть есть 2 пиксела с координатами P1(x1,y1) и P2(x2, y2).

Методы получения растрового приближения геометрических фигур различны, т.к. одну фигуру

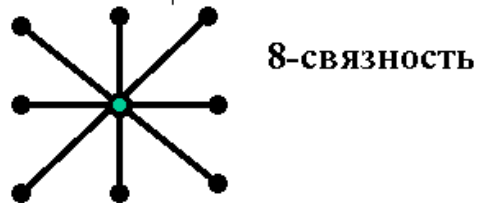
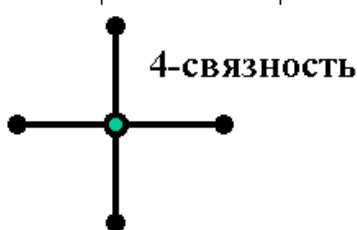
можно представить неоднозначно. Это касается, в основном, граничных точек.

Можно строить 4-и 8-связное представление. Возможны и другие растровые модели. Все зависит от того, какими свойствами требуется наделить полученный образ.

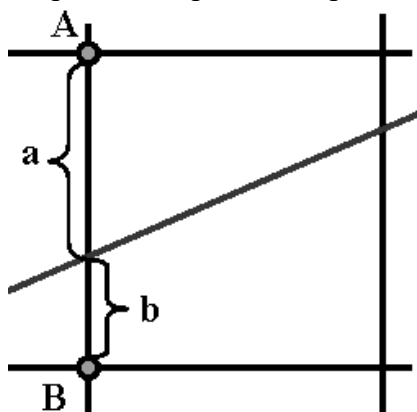


Пикселы P1 и P2 называются 4-СВЯЗНЫМИ, если у них отличаются только x-координаты или только y-координаты, причем только на 1: $|x1 - x2| + |y1 - y2| \leq 1$

Пикселы P1 и P2 называются 8-СВЯЗНЫМИ, если у них отличаются x-координаты и/или y-координаты, но не более, чем на 1: $|X1 - X2| \leq 1; |Y1 - Y2| \leq 1$



Очевидное решение - инициировать последовательно все точки растра, квадратные окрестности которых пересекают данный отрезок. Результатом является 4-связная растровая развертка отрезка. 8-связная решетка включает те точки растра, боковые стороны квадратных окрестностей которых пересекаются с отрезком.



Пусть отрезок имеет координаты концов M(x1,y1) и M2(x2,y2). Уравнение прямой $y=k*x+b$. Для простоты предположим, что угловой коэффициент $0 < k \leq 1$ (k не отрицателен и не превосходит 1), т.е. $0 \leq (y2-y1) \leq (x2-x1)$. Уравнение отрезка имеет вид $y = k*(x-x1)+y1$, где $k = (y2-y1)/(x2-x1)$.

В 1965 г. Брезенхейм (Bresenham) предложил простой целочисленный алгоритм растрового построения отрезка. Сначала он использовался в графопостроителях. Алгоритм основан на том, что при построении всегда

берется ближайший по вертикали пиксел, т.е. точка, ближайшая к исходной прямой. Очевидная оценка близости: если отклонение исходной прямой по оси ординат от текущего значения превышает 0.5, увеличиваем координату y на 1, корректируем текущее значение отклонения. Чтобы перейти к целочисленным расчетам, изменяют масштаб, умножая отклонение и приращение по оси y $d = (y_2 - y_1) / (x_2 - x_1)$ на некоторое целое $2 * n$. В качестве n естественно выбирается величина $x_2 - x_1$, т.е. отклонение вычисляется как $d = (x_2 - x_1) * (b - a)$. Если $d > 0$, значение y увеличивается на 1, иначе – не изменяется. d при этом уменьшается на "масштабированную" единицу, (вместо $\Delta y = \Delta y - 1$ пишем $\Delta y = \Delta y + 2 * ((y_2 - y_1) - (x_2 - x_1))$) или увеличивается на величину "масштабированного" приращения (вместо $\Delta y = \Delta y + ((y_2 - y_1) / (x_2 - x_1))$ пишем $\Delta y = \Delta y + 2 * (y_2 - y_1)$).

До сих пор рассматривался случай $0 < k < 1$ в уравнении прямой $y = k * x + b$. Для случая $|k| > 1$ переменные x и y меняются местами.

ПРАКТИЧЕСКАЯ ЧАСТЬ

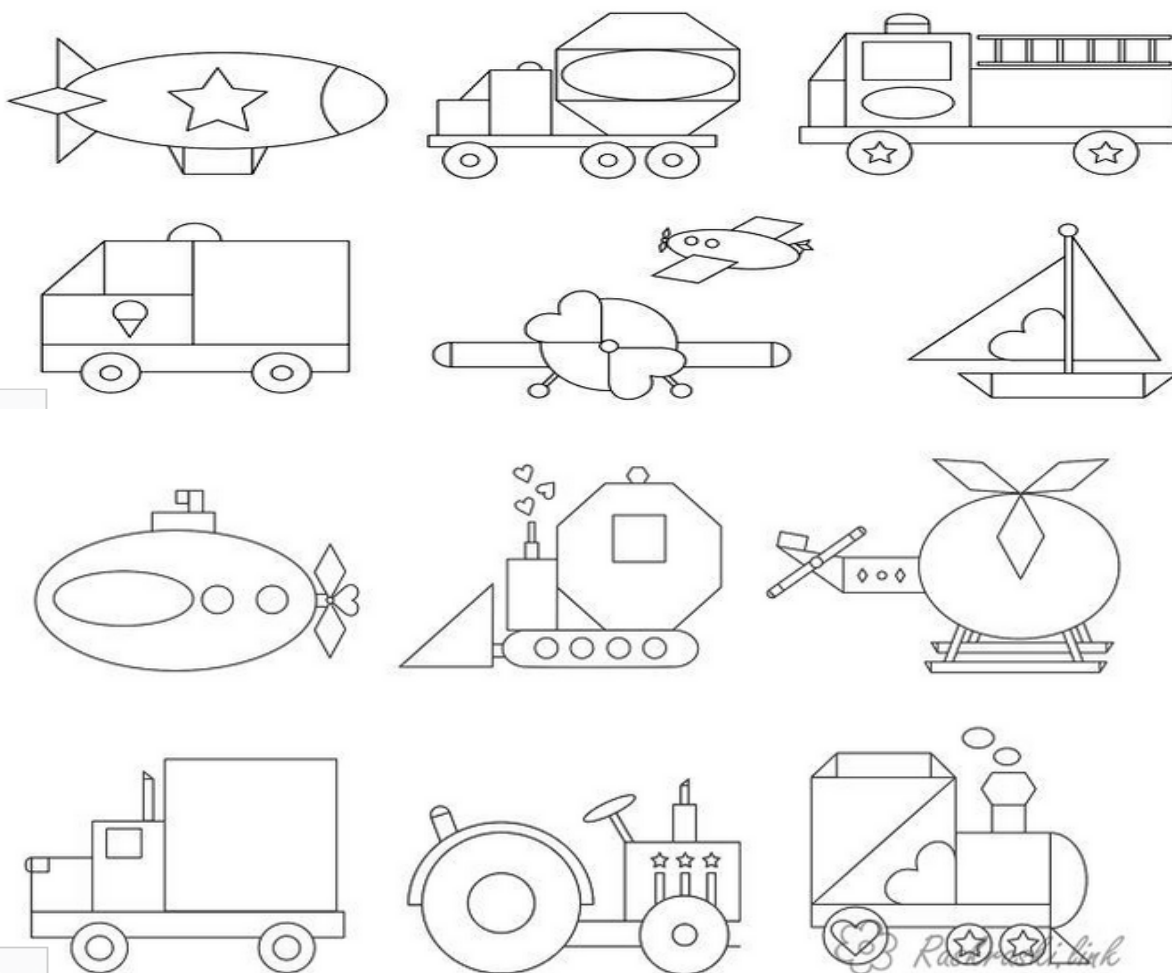
Задание №1. Изображение предметов с помощью стандартных фигур с использованием возможностей графического редактора

Время на подготовку – 120 минут, время на представление – 15 минут

Инструкция:

1. Рассмотрите панель стандартных геометрических фигур.
2. Используя стандартные фигуры геометрических тел, создайте следующие изображения
3. Раскрасьте получившиеся предметы. Если не указаны цвета, заливку можно выполнить по своему усмотрению.
4. При представлении работы пояснить выбор редактора, обосновав его возможности, описать правила, использованные при создании стиля созданной страницы, пояснить выбор цветов.

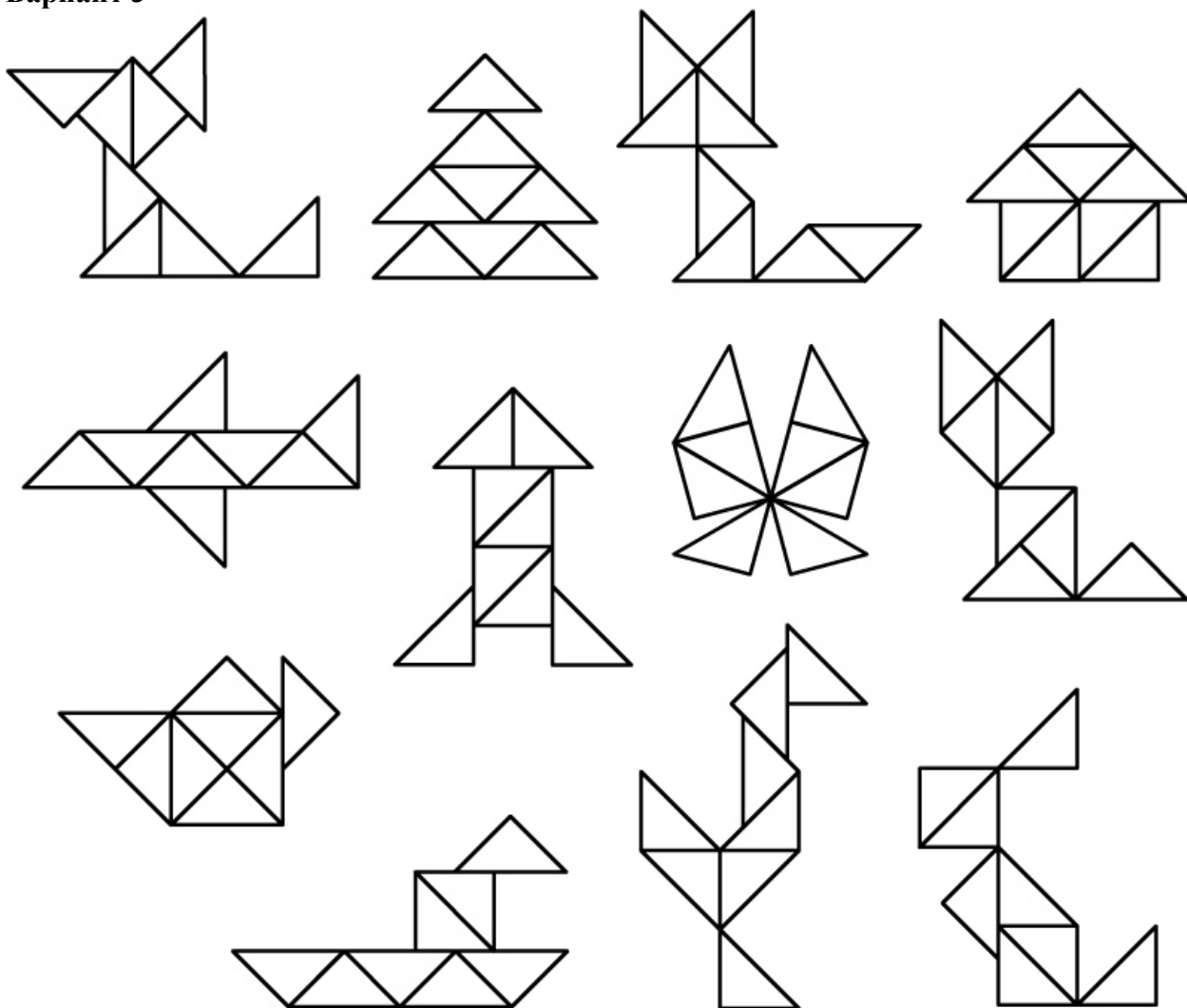
Вариант 1



Вариант 2



Вариант 3



Вариант 4



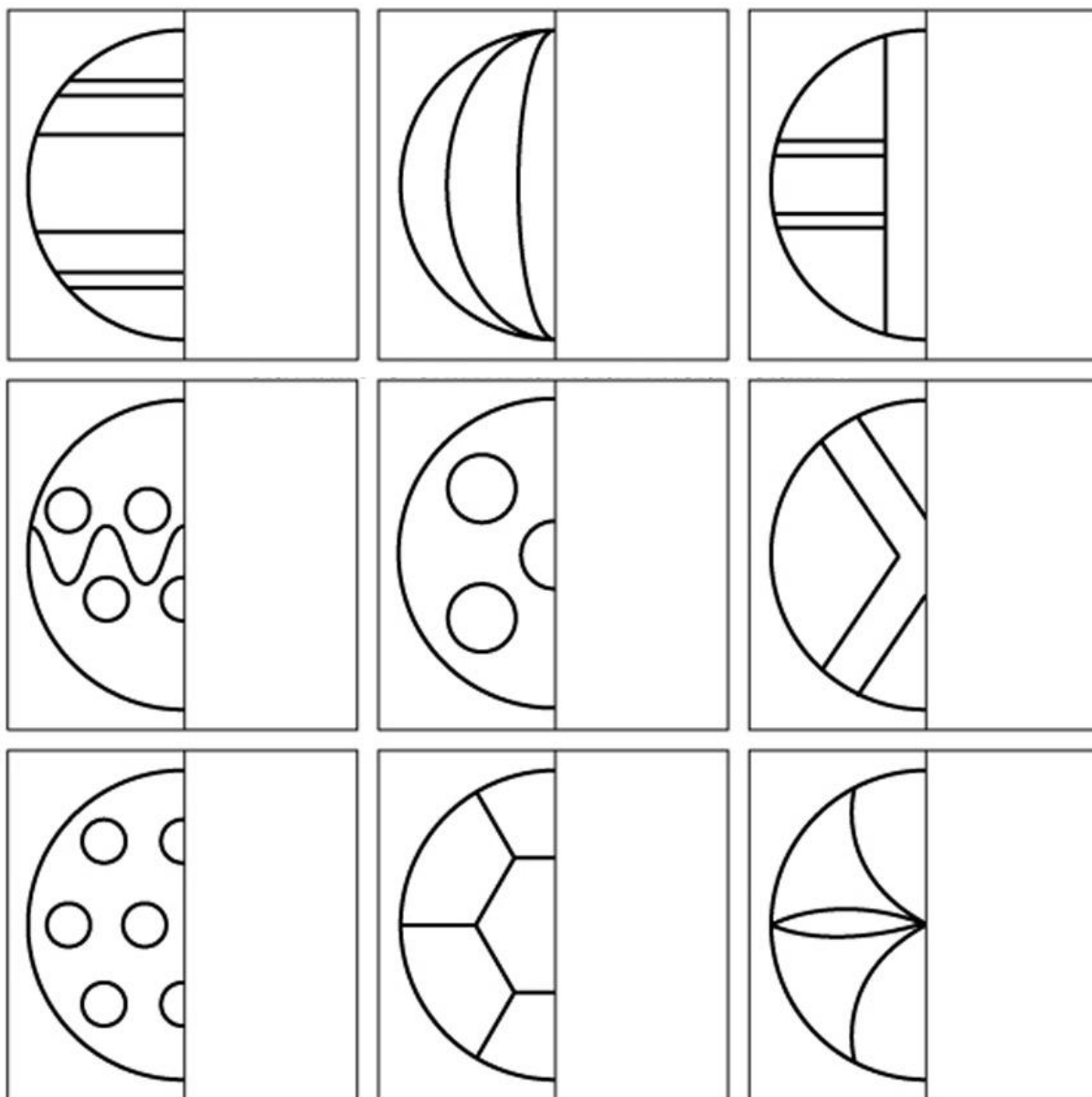
rukodelie-rukami.ru

Задание №2. Выполнение недостающей части рисунка в графическом редакторе paint

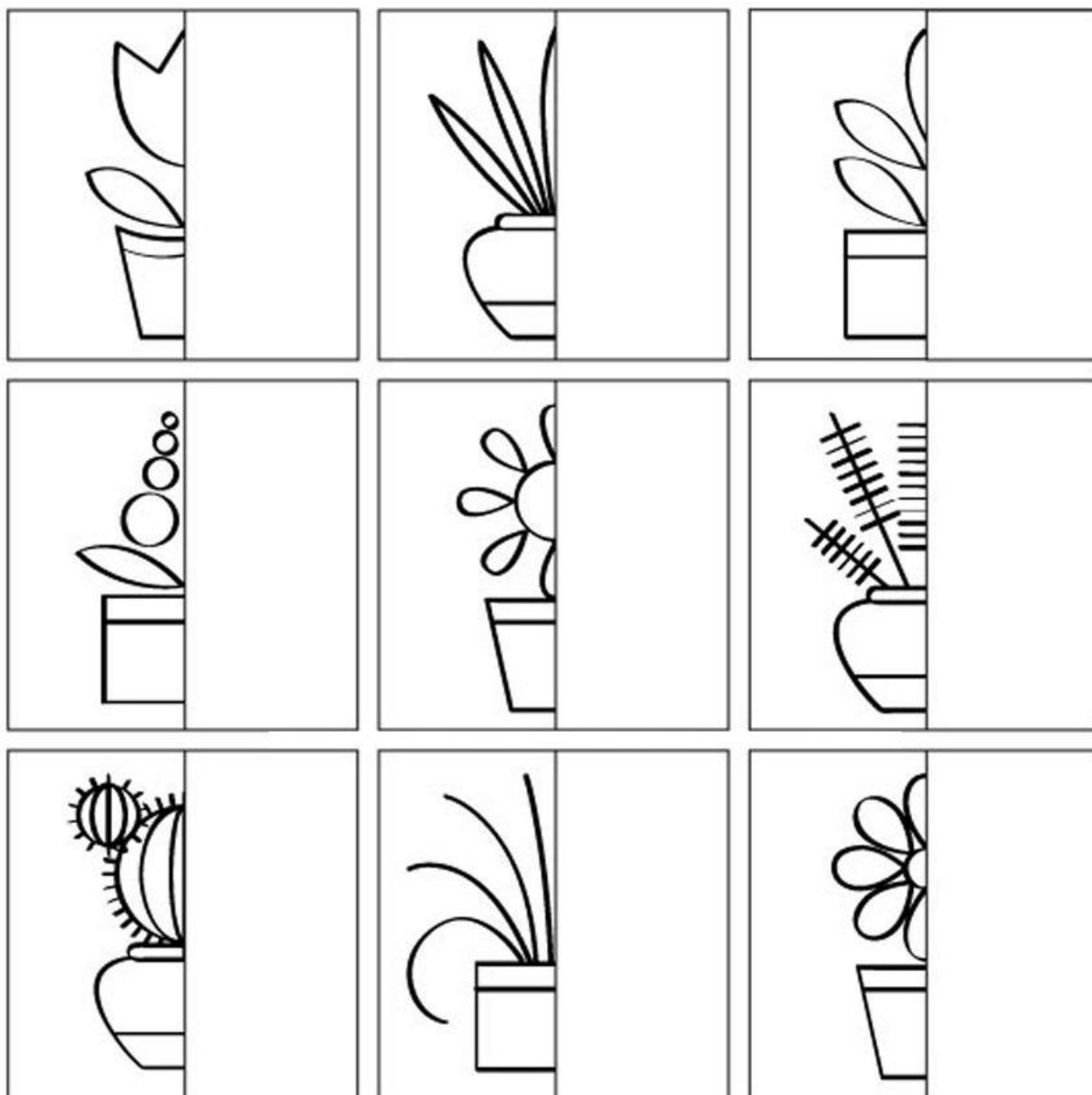
Инструкция:

1. Откройте самостоятельно выбранное задание(картинку) в графическом редакторе Paint
2. Выбери нужное изображение (изображение для работы определяет преподаватель) и дорисуйте недостающие элементы для получения целой картины
3. Раскрасьте получившееся изображение
4. Придумайте название изображения и напишите его сверху
5. Создайте веб-страницу с использованием редактора «Блокнот».
6. Разместить полученное изображение на веб-странице, используя придуманное название,
7. Оформите веб-страницу в деловом стиле.

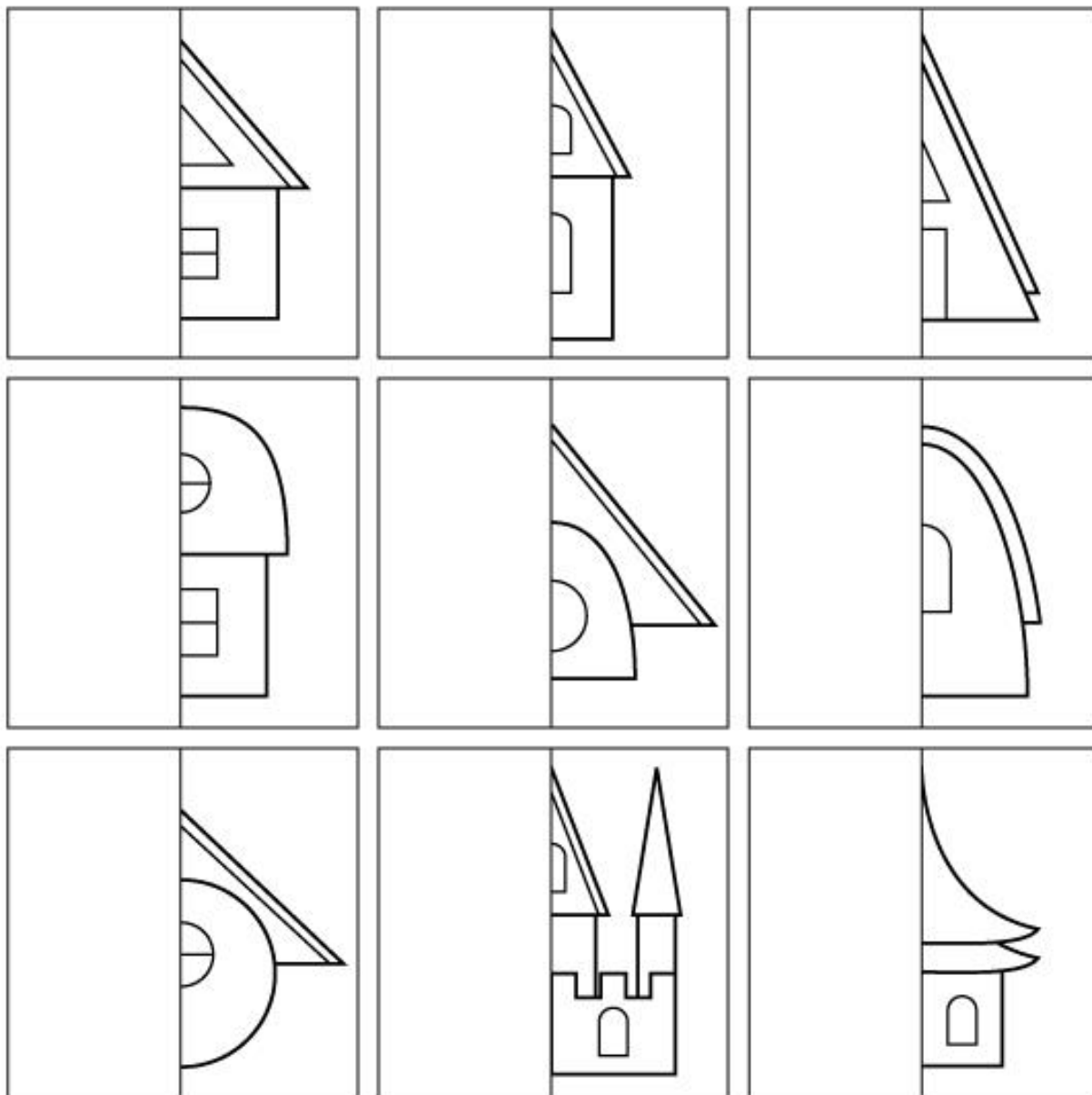
Вариант 1 “О спорт, - ты мир!” (низкий уровень сложности)



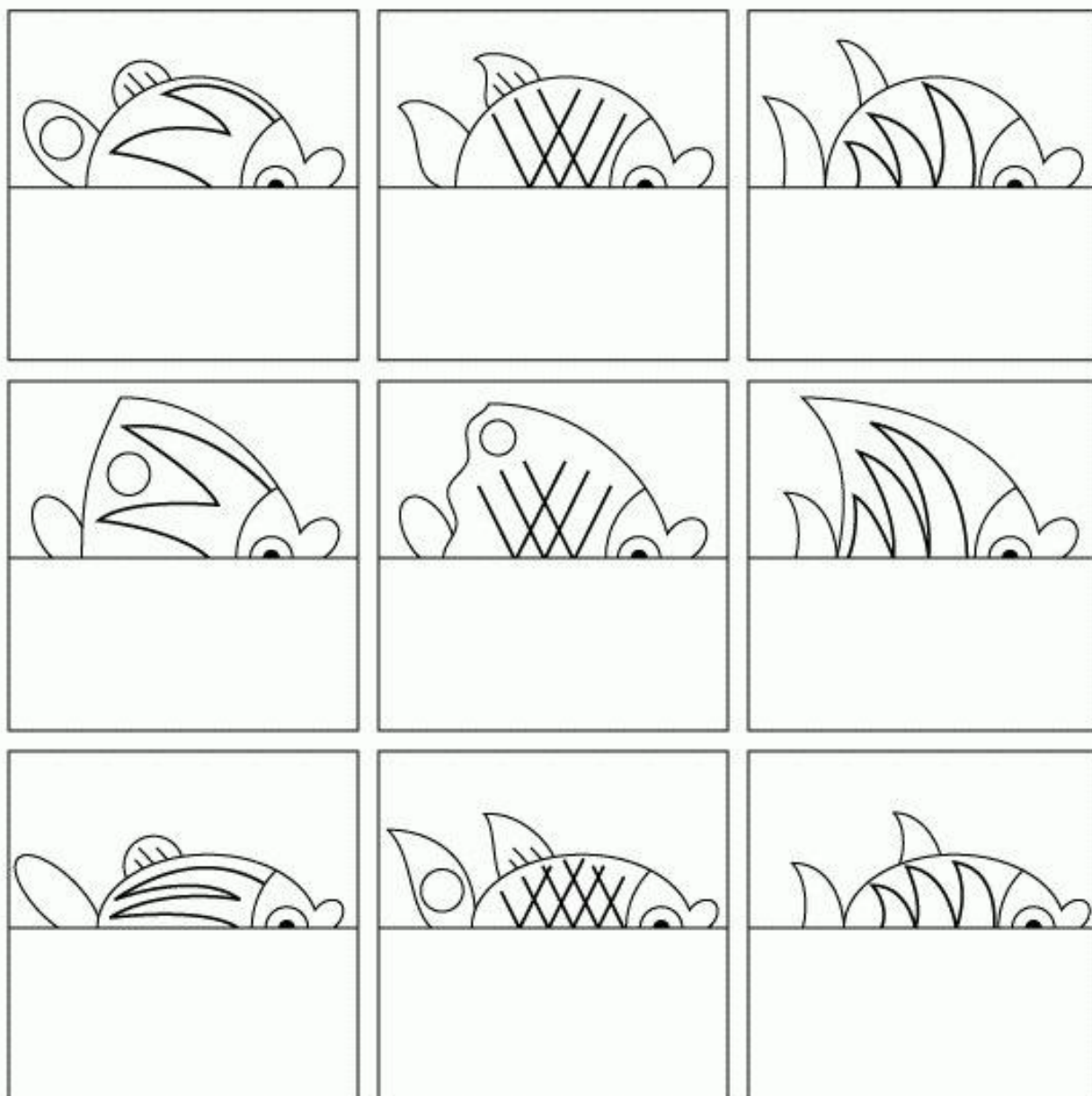
Вариант 2 “Мой зеленый уголок” (средний уровень сложности)



Вариант 3 “Тридцать девятое царство” (средний уровень сложности)



Вариант 4 “Подводный мир” (высокий уровень сложности)



ИТОГОВАЯ ОЦЕНКА ЭКЗАМЕНА (по модулю)

Баллы за задание экзамена (по модулю) рассчитываются по формуле:
(Задание 1) * 0,3 + (Задание 2) * 0,7 + (Материалы портфолио)*0,2
с применением правил математического округления.

Итогом экзамена является однозначное решение: вид профессиональной деятельности «Подготовка интерфейсной графики» освоен/не освоен.

Решение «освоен» принимается при получении итоговой оценки ≥ 3
(удовлетворительно)