

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")

РАССМОТРЕНО
на заседании Педагогического совета
протокол от «26» февраля 2024 г. № 4



**ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ
И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО МДК ПРОФЕССИОНАЛЬНОГО МОДУЛЯ
ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ
МОДУЛЕЙ**

по специальности

09.02.07 Информационные системы и программирование
квалификация «Программист»

форма обучения: очно-заочная

Вводится с 01.09.2024

Пермь 2024

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ.....	3
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ	4
3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ.....	6
4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО МДК.....	7
5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ МДК.....	

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (ФОС) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки специалистов среднего звена по профессиональному модулю ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ.

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 специалистов среднего звена по специальности СПО 09.02.07 Информационные системы и программирование;
- рабочей программы профессионального модуля ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ;
- учебного плана по специальности СПО 09.02.07 Информационные системы и программирование.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ МДК, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

Результатом в рамках освоения профессионального модуля ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ является овладение студентами вида профессиональной деятельности ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ, в том числе профессиональными (ПК) и общими (ОК) компетенциями:

Код	Наименование видов деятельности и профессиональных компетенций
ВД 2	Осуществление интеграции программных модулей
ПК 2.1.	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент
ПК 2.2.	Выполнять интеграцию модулей в программное обеспечение
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5.	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования
ВПК 2.6.	Выступать в роли бизнес-аналитика для представления решений, упрощающих и автоматизирующих рутинные офисные и бизнес-процессы
ВПК 2.7	Выступать в роли тренера для обучения пользователей применению прикладных программ
Код	Наименование общих компетенций
ОК 1	ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2	ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 3	ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 4	ОК 04. Эффективно взаимодействовать и работать в коллективе и команде
ОК 5	ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 6	ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
ОК 7	ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
ОК 8	ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках

В результате освоения профессионального модуля студент должен:

<p>иметь практический опыт</p>	<p>модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения</p>
<p>уметь</p>	<p>использовать выбранную систему контроля версий; использовать методы для получения кода с заданной функциональностью и степенью качества выступать в роли бизнес-аналитика для представления решений, упрощающих и автоматизирующих рутинные офисные и бизнес-процессы выступать в роли тренера для обучения пользователей применению прикладных программ</p>
<p>знать</p>	<p>модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения</p>

3. ОЦЕНКА УРОВНЕЙ ОСВОЕНИЯ МДК

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

В результате освоения профессионального модуля ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ ПРОГРАММНЫХ МОДУЛЕЙ 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.

4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

4.1. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ в форме экзамена МДК.02.01

Технология разработки программного обеспечения

1. Какие программы можно отнести к системному программному обеспечению:

Варианты ответа:

- 1) **операционные системы;**
- 2) прикладные программы;
- 3) игровые программы.

2. Какие программы можно отнести к системному ПО:

Варианты ответа:

- 1) **драйверы;**
- 2) текстовые редакторы;
- 3) электронные таблицы;
- 4) графические редакторы.

3. Специфические особенности ПО как продукта:

- 1) продажа по ценам ниже себестоимости (лицензирование);
- 2) **низкие материальные затраты при создании программ;**
- 3) **возможность создание программ небольшие коллективом или даже одним человеком;**
- 4) **разнообразие решаемых задач с помощью программных средств.**

4. Какие программы можно отнести к системному ПО:

Варианты ответа:

- 1) программа расчета заработной платы;
- 2) электронные таблицы;
- 3) **СУБД (системы управления базами данных).**

5. Какие программы нельзя отнести к системному ПО:

Варианты ответа:

- 1) **игровые программы;**
- 2) компиляторы языков программирования;
- 3) операционные системы;
- 4) системы управления базами данных.

6. Какие программы можно отнести к прикладному программному обеспечению:

Варианты ответа:

- 1) **электронные таблицы;**
- 2) таблицы решений;
- 3) СУБД (системы управления базами данных).

7. Какие программы можно отнести к прикладному ПО:

Варианты ответа:

- 1) **программа расчета заработной платы;**
- 2) диспетчер программ;
- 3) программа «Проводник» (Explorer).

8. Какие программы нельзя отнести к прикладному ПО:

Варианты ответа:

- 1) **компиляторы и (или) интерпретаторы;**
- 2) текстовые и (или) графические редакторы;
- 3) электронные таблицы.

9. Можно ли отнести операционную систему к программному обеспечению:

Варианты ответа:

- 1) да;
- 2) нет.

10. Можно ли отнести операционную систему к прикладному программному обеспечению:

Варианты ответа:

- 1) да;
- 2) нет.

11. Специфические особенности ПО как продукта:

Варианты ответа:

- 1) **низкие затраты при дублировании;**
- 2) универсальность;
- 3) простота эксплуатации;
- 4) наличие поддержки (сопровождения) со стороны разработчика.

12. Какие программы можно отнести к системному ПО:

Варианты ответа:

- 1) **утилиты;**
- 2) экономические программы;
- 3) статистические программы;
- 4) мультимедийные программы.

13. Этап, занимающий наибольшее время, в жизненном цикле программы:

Варианты ответа:

- 1) **сопровождение;**
- 2) проектирование;
- 3) тестирование;
- 4) программирование;
- 5) формулировка требований.

14. Этап, занимающий наибольшее время, при разработке программы:

Варианты ответа:

- 1) **тестирование;**
- 2) сопровождение;
- 3) проектирование;
- 4) программирование;
- 5) формулировка требований.

15. Первый этап в жизненном цикле программы:

Варианты ответа:

- 1) **формулирование требований;**
- 2) анализ требований;
- 3) проектирование;
- 4) автономное тестирование;
- 5) комплексное тестирование.

16. Один из необязательных этапов жизненного цикла программы:

Варианты ответа:

- 1) **оптимизация;**
- 2) проектирование;
- 3) тестирование;
- 4) программирование;
- 5) анализ требований.

17. Самый большой этап в жизненном цикле программы:

Варианты ответа:

- 1) **эксплуатация;**
- 2) изучение предметной области;

- 3) программирование;
- 4) тестирование;
- 5) корректировка ошибок.

18. Какой этап выполняется раньше:

Варианты ответа:

- 1) **отладка;**
- 2) тестирование.

19. Какой этап выполняется раньше:

Варианты ответа:

- 1) отладка;
- 2) оптимизация;
- 3) **программирование;**
- 4) тестирование.

20. Что выполняется раньше:

Варианты ответа:

- 1) **компиляция;**
- 2) отладка;
- 3) компоновка;
- 4) тестирование.

21. Что выполняется раньше:

Варианты ответа:

- 1) **проектирование;**
- 2) программирование;
- 3) отладка;
- 4) тестирование.

22. В стадии разработки программы не входит:

Варианты ответа:

- 1) **автоматизация программирования;**
- 2) постановка задачи;
- 3) составление спецификаций;
- 4) эскизный проект;
- 5) тестирование.

23. Самый важный критерий качества программы:

Варианты ответа:

- 1) **работоспособность;**
- 2) надежность;
- 3) эффективность;
- 4) быстрое действие;
- 5) простота эксплуатации.

24. Способы оценки качества:

Варианты ответа:

- 1) **сравнение с аналогами;**
- 2) наличие документации;
- 3) оптимизация программы;
- 4) структурирование алгоритма.

25. Существует ли связь между эффективностью и оптимизацией программы:

Варианты ответа:

- 1) **да;**
- 2) нет.

26. Наиболее важный критерий качества:

Варианты ответа:

- 1) **надежность;**

- 2) быстроедействие;
- 3) удобство в эксплуатации;
- 4) удобный интерфейс;
- 5) эффективность.

27. Способы оценки надежности:

Варианты ответа:

- 1) **тестирование;**
- 2) сравнение с аналогами;
- 3) трассировка;
- 4) оптимизация.

28. Повышает ли качество программ оптимизация:

Варианты ответа:

- 1) **да;**
- 2) нет.

29. Существует ли связь между надежностью и быстрымдействием:

Варианты ответа:

- 1) **нет;**
- 2) да.

30. В каких единицах можно измерить надежность:

Варианты ответа:

- 1) **отказов/час;**
- 2) км/час;
- 3) Кбайт/сек;
- 4) операций/сек.

31. В каких единицах можно измерить быстроедействие:

Варианты ответа:

- 1) отказов/час;
- 2) км/час;
- 3) Кбайт/сек;
- 4) **операций/сек.**

32. Что относится к этапу программирования:

Варианты ответа:

- 1) **написание кода программы;**
- 2) В) разработка интерфейса;
- 3) С) работоспособность;
- 4) анализ требований.

33. Последовательность этапов программирования:

Варианты ответа:

- 1) **компилирование, компоновка, отладка;**
- 2) В) компоновка, отладка, компилирование;
- 3) отладка, компилирование, компоновка;
- 4) компилирование, отладка, компоновка.

34) Инструментальные средства программирования:

Варианты ответа:

- 1) **компиляторы, интерпретаторы;**
- 2) СУБД (системы управления базами данных);
- 3) BIOS (базовая система ввода-вывода);
- 4) ОС (операционные системы).

35. На языке программирования составляется:

Варианты ответа:

- 1) **исходный код;**
- 2) исполняемый код;
- 3) объектный код;

4) алгоритм.

36. Правила, которым должна следовать программа это:

Варианты ответа:

- 1) **алгоритм;**
- 2) структура;
- 3) спецификация;
- 4) состав информации.

37. Можно ли внутри цикла поместить еще один цикл:

Варианты ответа:

- 1) **да;**
- 2) нет.

38. Можно ли внутри условного оператора поместить еще одно условие:

Варианты ответа:

- 1) **да;**
- 2) нет.

39. Можно ли одно большое (длинное) выражение разбить на несколько выражений:

Варианты ответа:

- 1) **да;**
- 2) нет.

40. Если имеется стандартная функция, нужно ли писать собственную:

Варианты ответа:

- 1) **нет;**
- 2) да.

41. Доступ, при котором записи файла читаются в физической последовательности, называется:

Варианты ответа:

- 1) **последовательным;**
- 2) прямым;
- 3) простым;
- 4) основным.

42. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:

Варианты ответа:

- 1) **прямым;**
- 2) последовательным;
- 3) простым;
- 4) основным.

43. Методы программирования (укажите НЕ верный ответ):

Варианты ответа:

- 1) **логическое;**
- 2) структурное;
- 3) модульное.

44. Что выполняется раньше:

Варианты ответа:

- 1) **разработка алгоритма;**
- 2) выбор языка программирования;
- 3) написание исходного кода;
- 4) компиляция.

45. Можно ли переменным присваивать произвольные идентификаторы:

Варианты ответа:

1) **да;**

2) **нет.**

46. Найдите НЕ правильное условие для создания имен:

Варианты ответа:

1) **имена могут содержать пробелы;**

2) длинное имя можно сократить;

3) из имени лучше выбрасывать гласные;

4) можно использовать большие буквы.

47. Какие символы не допускаются в именах переменных:

Варианты ответа:

1) **пробелы;**

2) цифры;

3) подчеркивание

48. Модно ли использовать имена, которые уже были использованы в другой программе (модуле):

Варианты ответа:

1) **да;**

2) **нет.**

49. Можно ли ставить знак подчеркивания в начале имени:

Варианты ответа:

1) **да, но не рекомендуется;**

2) да, без ограничений;

3) **нет.**

50. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:

Варианты ответа:

1) прямым указанием;

2) **венгерской нотацией;**

3) структурным программированием;

4) поляризацией.

51. Можно ли писать комментарии в отдельной строке:

Варианты ответа:

1) **да;**

2) **нет.**

52. Транслируются ли комментарии:

Варианты ответа:

1) да;

2) **нет.**

53. Наличие комментариев позволяет:

Варианты ответа:

1) **быстрее найти ошибки в программе;**

2) быстрее писать программы;

3) быстрее выполнять программы.

54. Наличие комментариев позволяет:

Варианты ответа:

1) **легче разобраться в программе;**

2) применять сложные структуры;

3) увеличить быстродействие.

55. Наличие комментариев позволяет:

Варианты ответа:

- 1) **улучшить читабельность программы;**
- 2) улучшить эксплуатацию программы;
- 3) повысить надежность программы.

56. Что определяет выбор языка программирования:

Варианты ответа:

- 1) **область приложения;**
- 2) знание языка;
- 3) наличие дополнительных библиотек.

57. Возможно ли комбинирование языков программирования в рамках одной задачи:

Варианты ответа:

- 1) **да;**
- 2) нет.

58. Допустимо ли комбинирование языков программирования в рамках одной задачи :

Варианты ответа:

- 1) **да;**
- 2) нет.

59. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:

Варианты ответа:

- 1) **для экономических задач;**
- 2) для системных задач;
- 3) для инженерных задач.

60. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:

Варианты ответа:

- 1) **для инженерных задач;**
- 2) для системных задач;
- 3) для экономических задач.

61. На каком этапе производится выбор языка программирования:

Варианты ответа:

- 1) **проектирование;**
- 2) программирование;
- 3) отладка;
- 4) тестирование.

62. Можно ли использовать комбинацию языков программирования в рамках одного проекта:

Варианты ответа:

- 1) **да;**
- 2) нет.

63. Для решения экономических задач характерно применение:

Варианты ответа:

- 1) **СУБД (систем управления базами данных);**
- 2) языков высокого уровня;
- 3) языков низкого уровня;
- 4) применение сложных математических расчетов.

64. Для решения инженерных задач характерно применение:

Варианты ответа:

- 1) САПР (систем автоматизированного проектирования);
- 2) СУБД (систем управления базами данных);
- 3) ОС (операционных систем).

65. Причины синтаксических ошибок:

Варианты ответа:

- 1) **плохое знание языка программирования;**
- 2) ошибки в исходных данных;
- 3) ошибки, допущенные на более ранних этапах;
- 4) неправильное применение процедуры тестирования.

66. Когда можно обнаружить синтаксические ошибки:

Варианты ответа:

- 1) **при компиляции;**
- 2) при отладке;
- 3) при тестировании;
- 4) на этапе проектирования;
- 5) при эксплуатации.

67. Ошибки компоновки заключаются в том, что:

Варианты ответа:

- 1) **указано внешнее имя, но не объявлено;**
- 2) неправильно использовано зарезервированное слово;
- 3) составлено неверное выражение;
- 4) указан неверный тип переменной.

68. Могут ли проявиться ошибки при изменении условий эксплуатации:

Варианты ответа:

- 1) **да;**
- 2) нет.

69. Могут ли проявиться ошибки при изменении в предметной области:

Варианты ответа:

- 1) **да;**
- 2) нет.

70. Возможно ли программирование с защитой от ошибок:

Варианты ответа:

- 1) **да;**
- 2) нет.

71. Есть ли недостатки программирования с защитой от ошибок:

Варианты ответа:

- 1) **да;**
- 2) нет.

72. Защитное программирование это:

Варианты ответа:

- 1) **встраивание в программу отладочных средств;**
- 2) создание задач защищенных от копирования;
- 3) разделение доступа в программе;
- 4) использование паролей;
- 5) оформление авторских прав на программу.

73. Вид ошибки с неправильным написанием служебных слов (операторов):

Варианты ответа:

- 1) **синтаксическая;**
- 2) семантическая;
- 3) логическая;
- 4) символьная.

74. Вид ошибки с неправильным использованием служебных слов (операторов):

Варианты ответа:

- 1) **семантическая;**
- 2) синтаксическая;
- 3) логическая;
- 4) символная.

75. Ошибки при написании программы бывают:

Варианты ответа:

- 1) **синтаксические;**
- 2) орфографические;
- 3) лексические;
- 4) фонетические;
- 5) морфологические.

76. Процедура поиска ошибки, когда известно, что она есть это:

Варианты ответа:

- 1) **отладка;**
- 2) тестирование;
- 3) компоновка;
- 4) транзакция;
- 5) трансляция.

77. Программа для просмотра значений переменных при выполнении программы:

Варианты ответа:

- 1) **отладчик;**
- 2) компилятор;
- 3) интерпретатор;
- 4) трассировка;
- 5) тестирование.

78. Отладка – это:

Варианты ответа:

- 1) **процедура поиска ошибок, когда известно, что ошибка есть;**
- 2) определение списка параметров;
- 3) правило вызова процедур (функций);
- 4) составление блок-схемы алгоритма.

79. Когда программист может проследить последовательность выполнения команд программы:

Варианты ответа:

- 1) **при трассировке;**
- 2) при тестировании;
- 3) при компиляции;
- 4) при выполнении программы;
- 5) при компоновке.

80. На каком этапе создания программы могут появиться синтаксические ошибки:

Варианты ответа:

- 1) **программирование;**
- 2) проектирование;
- 3) анализ требований;
- 4) тестирование.

81. Когда приступают к тестированию программы:

Варианты ответа:

- 1) **когда программа уже закончена;**
- 2) после постановки задачи;
- 3) на этапе программирования;
- 4) на этапе проектирования;
- 5) после составления спецификаций,

82. Тестирование бывает:

Варианты ответа:

- 1) **автономное;**
- 2) инструментальное;
- 3) визуальное;
- 4) алгоритмическое.

83. Тестирование бывает:

Варианты ответа:

- 1) **комплексное;**
- 2) инструментальное;
- 3) визуальное;
- 4) алгоритмическое.

84. Существует ли различие между отладкой и тестированием:

Варианты ответа:

- 1) **да;**
- 2) нет.

85. При комплексном тестировании проверяются:

Варианты ответа:

- 1) **согласованность работы отдельных частей программы;**
- 2) правильность работы отдельных частей программы;
- 3) быстродействие программы;
- 4) эффективность программы.

86. Чему нужно уделять больше времени, чтобы получить хорошую программу:

Варианты ответа:

- 1) **тестированию;**
- 2) программированию;
- 3) отладке;
- 4) проектированию.

87. Процесс исполнения программы с целью обнаружения ошибок:

Варианты ответа:

- 1) **тестирование;**
- 2) кодирование;
- 3) сопровождение;
- 4) проектирование.

88. Автономное тестирование это:

Варианты ответа:

- 1) **тестирование отдельных частей программы;**
- 2) инструментальное средство отладки;
- 3) составление блок-схем;
- 4) пошаговая проверка выполнения программы.

89. Трассировка это:

Варианты ответа:

- 1) **проверка пошагового выполнения программы;**
- 2) тестирование исходного кода;
- 3) отладка модуля;
- 4) составление блок-схемы алгоритма.

90. Локализация ошибки:

Варианты ответа:

- 1) **определение места возникновения ошибки;**
- 2) определение причин ошибки;
- 3) обнаружение причин ошибки;
- 4) исправление ошибки.

91. Назначение тестирования:

Варианты ответа:

- 1) **повышение надежности программы;**
- 2) обнаружение ошибок;
- 3) повышение эффективности программы;
- 4) улучшение эксплуатационных характеристик;
- 5) приведение программы к структурированному виду.

92. Назначение отладки:

Варианты ответа:

- 1) **поиск причин существующих ошибок;**
- 2) поиск возможных ошибок;
- 3) составление спецификаций;
- 4) разработка алгоритма.

93. Инструментальные средства отладки (НЕ правильный ответ):

Варианты ответа:

- 1) **компиляторы;**
- 2) отладчики;
- 3) трассировка.

94. Отладка программ это:

Варианты ответа:

- 1) **локализация и исправление ошибок;**
- 2) алгоритмизация программирования;
- 3) компиляция и компоновка.

95. Что выполняется раньше, автономная или комплексная отладка:

Варианты ответа:

- 1) **автономная;**
- 2) комплексная.

96. Что выполняется раньше, отладка или тестирование:

Варианты ответа:

- 1) **отладка;**
- 2) тестирование.

97. Что такое автоматизация программирования:

Варианты ответа:

- 1) **создание исходного кода программными средствами;**
- 2) создание исходного кода при помощи компилятора;
- 3) создание исходного кода без разработки алгоритма.

98. В чем сущность автоматизации программирования:

Варианты ответа:

- 1) **создание программы без написания ее текста;**
- 2) получение готовой программы без выполнения компоновки;
- 3) в отсутствии компиляции.

99. Возможна ли автоматизация программирования:

Варианты ответа:

- 1) **да;**
- 2) нет.

100. Создание исполняемого кода программы без написания исходного кода называется:

Варианты ответа:

- 1) **составлением спецификаций;**

- 2) отладкой;
- 3) проектированием.
- 4) **автоматизацией программирования;**

101. Одно из преимуществ автоматизации программирования:

Варианты ответа:

- 1) **наглядное программирование с визуальным контролем;**
- 2) получение стандартной программы;
- 3) создание программы с оптимальным кодом.

102. Один из методов автоматизации программирования:

Варианты ответа:

- 1) структурное программирование;
- 2) модульное программирование;
- 3) **визуальное программирование;**
- 4) объектно-ориентированное программирование.

103. Влияет ли автоматизация программирования на эффективность программы:

Варианты ответа:

- 1) **нет;**
- 2) да

104. Автоматизация программирования позволяет:

Варианты ответа:

- 1) повысить надежность программы;
- 2) **сократить время разработки программы;**
- 3) повысить быстродействие программы.

105. Позволяет ли автоматизация программирования всегда создавать эффективные программы:

Варианты ответа:

- 1) да.
- 2) **нет;**

106. Позволяет ли автоматизация программирования всегда создавать надежные программы:

Варианты ответа:

- A) **нет;**
- B) да.

107. Недостаток автоматизации программирования;

Варианты ответа:

- B) низкое быстродействие;
- A) **большой размер программы;**
- C) сложность программы.

108. Возможны ли ошибки при автоматизации программирования:

Варианты ответа:

- A) **да;**
- B) нет.

109. Нужно ли выполнять тестирование при автоматизации программирования:

Варианты ответа:

- A) **да;**
- B) нет.

110. Выполняется ли процедура компиляции при автоматизации программирования:

Варианты ответа:

- A) да;
- B) нет.

111. Что легко поддается автоматизации:

Варианты ответа:

- A) **интерфейс;**
- B) работа с файлами;
- C) сложные логические задачи;
- D) алгоритмизация.

112. Относится ли визуальное программирование к средствам автоматизации:

Варианты ответа:

- A) **да;**
- B) нет.

113. Нахождение наилучшего варианта из множества возможных:

Варианты ответа:

- A) **оптимизация;**
- B) тестирование;
- C) автоматизация;
- D) отладка;
- E) сопровождение.

114. Что такое оптимизация программ:

Варианты ответа:

- A) **улучшение работы существующей программы;**
- B) создание удобного интерфейса пользователя;
- C) разработка модульной конструкции программы;
- D) применение методов объектно-ориентированного программирования.

115. Критерии оптимизации:

Варианты ответа:

- A) **время выполнения или размер требуемой памяти;**
- B) размер программы и ее эффективность;
- C) независимость модулей;
- D) качество программы, ее надежность.

116. Критерии оптимизации:

Варианты ответа:

- A) **эффективность использования ресурсов;**
- B) структурирование алгоритма;
- C) структурирование программы.

117. Возможна ли оптимизация программ без участия программиста:

Варианты ответа:

- A) **да;**
- B) нет.

118. Возможна ли оптимизация циклов:

Варианты ответа:

- A) **да;**
- B) нет.

119. В чем заключается оптимизация условных выражений:

Варианты ответа:

- A) **в изменении порядка следования элементов выражения;**
- B) в использовании простых логических выражений;

С) в использовании сложных логических выражений;

Д) в использовании операций AND, OR и NOT.

120. Оптимизация циклов заключается в:

Варианты ответа:

А) уменьшении количества повторений тела цикла;

В) просмотре задачи с другой стороны;

С) упрощение задачи за счет включения логических операций.

121. Оптимизация программы это:

Варианты ответа:

А) модификация;

В) отладка;

С) повышение сложности программы;

Д) уменьшение сложности программы.

122. Критерии оптимизации программы:

Варианты ответа:

А) быстродействие или размер программы;

В) быстродействие и размер программы;

С) надежность или эффективность;

Д) надежность и эффективность.

123. Результат оптимизации программы:

Варианты ответа:

А) эффективность;

В) надежность;

С) машино-независимость;

Д) мобильность.

124. Сущность оптимизации циклов:

Варианты ответа:

А) сокращение количества повторений выполнения тела цикла;

В) сокращение тела цикла;

С) представление циклов в виде блок-схем;

Д) трассировка циклов;

Е) поиск ошибок в циклах.

125. В чем сущность модульного программирования:

Варианты ответа:

А) в разбиении программы на отдельные функционально независимые части;

В) в разбиении программы на отдельные равные части;

С) в разбиение программы на процедуры и функции;

126. Можно ли сочетать модульное и структурное программирование:

Варианты ответа:

А) да;

В) нет.

127. Может ли модуль включать несколько процедур или функций:

Варианты ответа:

А) да;

В) нет.

128. Рекомендуемые размеры модулей:

Варианты ответа:

А) небольшие;

В) большие;

С) равные;

Д) фиксированной длины.

129. В чем заключается независимость модуля:

Варианты ответа:

- A) в написании, отладке и тестировании независимо от остальных модулей;**
- B) в разработке и написании независимо от других модулей;
- C) в независимости от работы основной программы.

130. При модульном программировании желательно, чтобы модуль имел:

Варианты ответа:

- A) большой размер;
- B) небольшой размер;**
- C) фиксированный размер;
- D) любой размер.

131. Модульное программирование это:

Варианты ответа:

- A) разбиение программы на отдельные части;**
- B) структурирование;
- C) использование стандартных процедур и функций.

132. Можно ли использовать оператор GO TO в модульных программах:

Варианты ответа:

- A) можно;**
- B) нельзя.

133. Разрешается ли использование циклов при модульном программировании:

Варианты ответа:

- A) да;**
- B) нет.

134. Разрешается ли использование условных операторов при модульном программировании:

Варианты ответа:

- A) да;**
- B) нет.

135. Сократится ли размер программы, если ее написать в виде набора модулей:

Варианты ответа:

- A) нет;**
- B) да.

136. Достоинство модульного программирования:

Варианты ответа:

- A) создание программы по частям в произвольном порядке;**
- B) не требует компоновки;
- C) всегда дает эффективные программы;
- D) снижает количество ошибок.

137. Недостаток модульного программирования:

Варианты ответа:

- A) увеличивает трудоемкость программирования;
- B) усложняет процедуру комплексного тестирования;**
- C) снижает быстродействие программы;
- D) не позволяет выполнять оптимизацию программы.

138. Достоинство модульного программирования:

Варианты ответа:

- A) возможность приступить к тестированию до завершения написания всей программы;**

- В) не требует комплексного тестирования;
- С) уменьшает размер программы;
- Д) повышает надежность программы.

139. Допустимо ли использование оператора GO TO при структурном программировании:

1. Варианты ответа:

А) нет;

В) да.

140. Можно ли сочетать структурное программирование с модульным:

Варианты ответа:

А) можно;

В) нельзя;

С) только в особых случаях.

141. Любую ли программу можно привести к структурированному виду:

Варианты ответа:

А) любую;

В) не все;

С) нельзя.

142. Можно ли использовать оператор GO TO в структурированных программах:

Варианты ответа:

А) можно;

В) нельзя;

С) только в особых случаях.

143. Возможно, ли преобразовать неструктурированную программу к структурному виду:

Варианты ответа:

А) да;

В) нет.

144. Возможно ли программирование без оператора GO TO:

Варианты ответа:

А) да;

В) нет.

145. При структурном программировании задача выполняется:

Варианты ответа:

А) поэтапным разбиением на более легкие задачи;

В) без участия программиста;

С) объединением отдельных модулей программы.

146. Разрешается ли использование оператора GO TO при структурном программировании:

Варианты ответа:

А) нет;

В) да;

С) иногда.

147. Разрешается ли использование циклов при структурном программировании:

Варианты ответа:

А) да;

В) нет.

148. Разрешается ли использование оператора IF при структурном программировании:

Варианты ответа:

А) да;

В) нет.

149. Программирование без GO TO применяется. при:

Варианты ответа:

- A) **структурном программировании;**
- B) модульном программировании;
- C) объектно-ориентированном программировании;
- D) все ответы верные.

150. Достоинство структурного программирования:

Варианты ответа:

- A) **можно приступить к комплексному тестированию на раннем этапе разработки;**
- B) можно приступить к автономному тестированию на раннем этапе разработки;
- C) нет необходимости выполнять тестирование;
- D) можно пренебречь отладкой.

151. Достоинство структурного программирования:

Варианты ответа:

- A) **облегчает работу над большими и сложными проектами;**
- B) повышает быстродействие программы;
- C) снижает затраты на программирование.

152. Недостаток структурного программирования:

Варианты ответа:

- A) **увеличивает размер программы;**
- B) снижает эффективность;
- C) уменьшает количество ошибок;
- D) не требует отладки.

153. Повышает ли читабельность программ структурное кодирование:

Варианты ответа:

- A) **да;**
- B) нет.

154. Разрешается ли использование циклов при объектно-ориентированном программировании:

Варианты ответа:

- A) **да;**
- B) нет.

2. 155. Разрешается ли использование оператора IF при объектно-ориентированном программировании:

3. Варианты ответа:

- A) **да;**
- B) нет.

156. Предусматривает ли объектно-ориентированное программирование использование стандартных процедур и функций:

- A) **да;**
- B) нет.

157. Можно ли сочетать объектно-ориентированное и структурное программирование

Варианты ответа:

- A) **можно;**
- B) нельзя.

158) Можно ли сочетать объектно-ориентированное и модульное программирование:

Варианты ответа:

- A) **можно;**
- B) нельзя.

159. Что такое объект, в объектно-ориентированном программировании:

Варианты ответа:

- A) тип данных;**
- B) структура данных;
- C) событие;
- D) обработка событий;
- E) использование стандартных процедур.

160. Инкапсуляция это:

Варианты ответа:

- A) определение новых типов данных;
- B) определение новых структур данных;
- C) объединение переменных, процедур и функций в одно целое;**
- D) разделение переменных, процедур и функций;
- E) применение стандартных процедур и функций.

161. Наследование это:

Варианты ответа:

- A) передача свойств экземплярам;
- B) передача свойств предкам;
- C) передача свойств потомкам;**
- D) передача событий потомкам.

162. Полиморфизм это:

Варианты ответа:

- A) изменение поведения потомков, имеющих общих предков;**
- B) передача свойств по наследству;
- C) изменение поведения потомков на разные события;
- D) изменение поведения экземпляров, имеющих общих предков;

163. Три "кита" объектно-ориентированного метода программирования:

Варианты ответа:

- A) предки, родители, потомки;
- B) полиморфизм, инкапсуляция, наследование;**
- C) свойства, события, методы;
- D) визуальные, не визуальные компоненты и запросы.

164. Какое утверждение верно:

Варианты ответа:

- A) предки наследуют свойства родителей;
- B) родители наследуют свойства потомков;
- C) потомки не могут иметь общих предков;
- D) потомки наследуют свойства родителей.**

165. Может ли дочерний элемент иметь двух родителей:

Варианты ответа:

- A) да;
- B) нет;**
- C) только для визуальных элементов;
- D) если их свойства совпадают.

165. Могут ли два визуальных компонента иметь общего предка:

Варианты ответа:

- A) да;**
- B) нет;
- C) если их свойства совпадают;
- D) если их методы совпадают.

167. Есть ли различие между объектом и экземпляром:

Варианты ответа:

- A) да;**
- B) нет;
- C) если у них общий предок.

168. Есть ли различие в поведении объекта и экземпляра того же типа:

Варианты ответа:

- A) да;
- B) если у них есть общий предок;
- C) нет;**
- D) если у них нет общего предков.

169. Изменение свойств, приводит к изменению поведения экземпляра:

Варианты ответа:

- A) нет;
- B) только для визуальных;
- C) только НЕ для визуальных ;
- D) да .**

170. Можно ли свойствам присваивать значения:

Варианты ответа:

- A) да (всегда);
- B) не всегда;**
- C) нет.

171. Можно ли переопределять методы:

Варианты ответа:

- A) да;**
- B) нет.

172. Можно ли переопределять свойства:

Варианты ответа:

- A) да;
- B) нет.**

173. Могут ли два различных объекта реагировать на событие по-разному:

Варианты ответа:

- A) да;**
- B) нет.

174. Могут ли два экземпляра одного объекта реагировать на событие по-разному:

Варианты ответа:

- A) да;**
- B) нет.

175. Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

- A) сверху вниз;**
- A) снизу-вверх.

176. Какой этап проектирования может быть исключен:

Варианты ответа:

- A) эскизный проект;**
- B) технический проект;
- C) рабочий проект.

177. Какие этапы проектирования можно объединять:

Варианты ответа:

- A) технический и рабочий;**
- B) эскизный и рабочий;
- C) технический и эскизный.

178. Модульное программирование применимо при:

Варианты ответа:

А) проектировании сверху вниз;

В) проектирование снизу-вверх;

179. Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения это:

Варианты ответа:

А) проектирование;

В) анализ требований;

С) программирование;

Д) тестирование.

180. Составление спецификаций это:

Варианты ответа:

А) формализация задачи;

В) эскизный проект;

С) поиск алгоритма;

Д) отладка.

181. Этап разработки программы, на котором дается характеристика области применения программы:

Варианты ответа:

А) техническое задание;

В) эскизный проект;

С) технический проект;

Д) внедрение;

Е) рабочий проект.

182. Укажите правильную последовательность создания программы:

Варианты ответа:

А) формулирование задачи, анализ требований, проектирование, программирование;

В) анализ требований, проектирование, программирование, тестирование, отладка;

С) анализ требований, программирование, проектирование, тестирование;

Д) анализ требований, проектирование, программирование, модификация, трассировка;

Е) формулирование задачи, анализ требований, программирование, проектирование, отладка.

183. Уточнение структуры входных и выходных данных, разработка алгоритмов, определение элементов интерфейса входят в:

Варианты ответа:

А) технический проект;

В) рабочий проект;

С) эскизный проект.

184. Несуществующий метод проектирования:

Варианты ответа:

А) алгоритмическое;

В) нисходящее;

С) восходящее.

185. Метод проектирования:

Варианты ответа:

А) нисходящее;

В) алгоритмическое;

С) логическое;

Д) использование языков программирования;

Е) составление блок-схем.

186. Нисходящее проектирование это:

Варианты ответа:

- A) последовательное уточнение (детализация);**
- В) составление блок-схем;
- С) разделение программы на отдельные участки (блоки);
- Д) трассировка.

187. Признаки нисходящего программирования:

Варианты ответа:

- A) последовательная детализация;**
- В) наличие оптимизации;
- С) наличие тестирования;
- Д) автоматизация программирования.

188. Какой методикой пользуются при структурном программировании:

Варианты ответа:

- A) сверху вниз;**
- В) снизу-вверх.

189. Проектирование сверху вниз это:

Варианты ответа:

- A) последовательное разбиение общих задач на более мелкие;**
- В) составление из отдельных модулей большой программы.

190. Проектирование снизу-вверх это:

Варианты ответа:

- A) составление из отдельных модулей большой программы;**
- В) последовательное разбиение общих задач на более мелкие.

191. Модульное программирование применимо при:

Варианты ответа:

- А) проектировании сверху вниз;
- В) проектирование снизу-вверх;
- С) и в том, и другом случае;**
- Д) ни в коем случае.

192. Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

- A) сверху вниз;**
- В) снизу-вверх.

193. В чем заключается иерархический подход в решении задачи:

Варианты ответа:

- A) в последовательном разбиении задачи на более мелкие составные части;**
- В) в выделении основных и второстепенных элементов;
- С) в возможности параллельного выполнения отдельных частей задачи.

4. 194. Какой метод проектирования соответствует иерархическому подходу в решении задачи:

5. Варианты ответа:

- A) нисходящее (сверху вниз);**
- В) восходящее (снизу-вверх).

195. В каких единицах измеряются затраты на проектирование:

Варианты ответа:

- A) в человеко-днях;**
- В) в долларах;
- С) в тенге;
- Д) в килобайтах.

196. Зависит ли трудоемкость разработки от сложности алгоритма:

Варианты ответа:

- A) да;**
- В) нет.

197. Зависит ли трудоемкость разработки от количества программистов:

Варианты ответа:

А) да;

В) нет.

198. Зависит ли трудоемкость разработки от языка или системы программирования:

Варианты ответа:

А) да;

В) нет.

199. Зависит ли трудоемкость разработки от количества обрабатываемой информации:

Варианты ответа:

А) да;

В) нет.

200. Зависит ли трудоемкость разработки от вида информации:

Варианты ответа:

А) да;

В) нет.

201. Если вы приобрели программу законным путем, являетесь ли вы собственником программы:

Варианты ответа:

А) нет;

В) да.

202. Если вы приобрели программы законным путем, имеете ли вы право вносить в нее изменения:

Варианты ответа:

А) нет;

В) да

203. Если вы приобрели программы законным путем, имеете ли вы право продать ее:

Варианты ответа:

А) да;

В) нет.

204. Кому принадлежит право собственности на ПО:

Варианты ответа:

А) разработчику;

А) продавцу;

В) покупателю.

205. Кому принадлежит авторское право на ПО:

Варианты ответа:

А) разработчику;

С) продавцу;

Д) покупателю.

206. Что охраняется законом:

Варианты ответа:

А) структура базы данных;

В) содержание базы данных

4.2. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

в форме экзамена МДК.02.02
ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

ТЕСТ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1.	Технология — ASP.NET	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
2.	Технология NetCF — .NET Compact Framework 3.5	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
3.	Технология —Silverlight	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию</p>

		<p>(предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
4.	Технология VSTO — Visual Studio Tools for Office	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
5.	Технология — WinForms	<p>1) используются для разработки бизнес-приложений и различных инструментов для платформы Windows.</p> <p>2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows.</p> <p>3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio.</p> <p>4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p>
6.	Технология WPF Windows Presentation Foundation	<p>1) используются для разработки бизнес-приложений и различных инструментов для платформы Windows.</p> <p>2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows.</p>

		<p>3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio.</p> <p>4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p>
7.	Технология XNA	<p>1) используются для разработки бизнес-приложений и различных инструментов для платформы Windows.</p> <p>2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows.</p> <p>3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio.</p> <p>4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p>
8.	Элемент управления пользовательского интерфейса для Web-разработчика: ListView -	<p>1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами.</p> <p>2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения.</p> <p>3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным.</p> <p>4) поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
9.	Элемент управления пользовательского интерфейса для Web-разработчика: DataPager -	<p>1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами.</p> <p>2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения.</p> <p>3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает</p>

		<p>запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным.</p> <p>4) поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
10.	<p>Элемент управления пользовательского интерфейса для Web-разработчика: LinqDataSourceControl -</p>	<p>1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами.</p> <p>2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения.</p> <p>3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным.</p> <p>4) поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
11.	<p>Представление Source в Visual Studio 2008</p>	<p>1) дает вам полный доступ к редактированию кода XHTML данной страницы.</p> <p>2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого вызова, связанным с элементами управления в визуальном конструкторе.</p> <p>3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
12.	<p>Представление Design в Visual Studio 2008</p>	<p>1) дает вам полный доступ к редактированию кода XHTML данной страницы.</p> <p>2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого вызова, связанным с элементами управления в визуальном конструкторе.</p> <p>3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>

13.	Представление Split в Visual Studio 2008	<p>1) дает вам полный доступ к редактированию кода XHTML данной страницы.</p> <p>2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого вызова, связанным с элементами управления в визуальном конструкторе.</p> <p>3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
14.	Visual Studio Team System Development Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
15.	Visual Studio Team System Database Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы,</p>

		<p>сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
16.	Visual Studio Team System Architect Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
17.	Visual Studio Team System Test Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет</p>

		<p>вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) — предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
18.	Visual Studio Team Suite	<p>1) — для тех, кому нужно иметь все сразу (и у кого нет бюджетных ограничений) этот продукт содержит в себе абсолютно все.</p> <p>2) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>3) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>4) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p>
19.	Toolbox	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм.</p> <p>2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение.</p> <p>3) окно свойств.</p>

		4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.
20.	Solution Explorer	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм.</p> <p>2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение.</p> <p>3) окно свойств.</p> <p>4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.</p>
21.	Properties	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм.</p> <p>2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение.</p> <p>3) окно свойств.</p> <p>4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.</p>
22.	Object Browser	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм.</p> <p>2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение.</p> <p>3) окно свойств.</p> <p>4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.</p>
23.	ASP.NET Web Site	1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config.

		<p>2) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p> <p>4) — используется для создания Web-сайта, который содержит сервисы WCF.</p>
24.	ASP.NET Web Service	<p>1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config.</p> <p>2) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p> <p>4) — используется для создания Web-сайта, который содержит сервисы WCF.</p>
25.	Empty Web Site	<p>1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config.</p> <p>2) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p> <p>4) — используется для создания Web-сайта, который содержит сервисы WCF.</p>
26.	WCF Service	<p>1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config.</p> <p>2) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p> <p>4) — используется для создания Web-сайта, который содержит сервисы WCF.</p>

27.	ASP.NET Reports Web Site	<p>1) — создает сайт для формирования Web-отчетов (при помощи инструментов Microsoft Report Viewer).</p> <p>2) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config.</p> <p>3) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>4) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p>
28.	Расширение файла aspx	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
29.	Расширение файла asmx	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
30.	Расширение файла cs, vb	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
31.	Расширение файла config	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p>

		4) Представляет собой конфигурационный файл вашего Web-приложения.
32.	Page Preinit	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p> <p>2) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p> <p>5) это последнее событие, которое вызывается для страницы (после визуализации страницы).</p>
33.	Page Init—	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p> <p>2) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p>
34.	Page Load —	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p>

		<p>2) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p>
35.	Page Pre Render—	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p> <p>2) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p>
36.	Page Unload—	<p>1) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>2) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>3) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p> <p>4) это последнее событие, которое вызывается для страницы (после визуализации страницы).</p>
37.	Элемент управления Label	<p>1) для передачи пользователю текстовой информации.</p> <p>2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные.</p> <p>3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на</p>

		<p>странице некого действия (такого как сохранение, сброс или отмена).</p> <p>4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.</p>
38.	Элемент управления TextBox	<p>1) для передачи пользователю текстовой информации.</p> <p>2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные.</p> <p>3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некого действия (такого как сохранение, сброс или отмена).</p> <p>4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.</p>
39.	Элемент управления Button	<p>1) для передачи пользователю текстовой информации.</p> <p>2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные.</p> <p>3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некого действия (такого как сохранение, сброс или отмена).</p> <p>4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.</p>
40.	Элемент управления HyperLink	<p>1) для передачи пользователю текстовой информации.</p> <p>2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные.</p> <p>3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некого действия (такого как сохранение, сброс или отмена).</p> <p>4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.</p>

41.	Элемент управления DropDownList	<p>1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать.</p> <p>2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено).</p> <p>3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.</p> <p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства imageUrl).</p>
42.	Элемент управления CheckBox	<p>1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать.</p> <p>2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено).</p> <p>3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.</p> <p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства imageUrl).</p>
43.	Элемент управления RadioButton	<p>1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать.</p> <p>2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено).</p> <p>3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.</p> <p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять</p>

		изображение на вашей странице (при помощи свойства imageUrl).
44.	Элемент управления Image	<p>1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать.</p> <p>2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено).</p> <p>3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.</p> <p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства imageUrl).</p>
45.	Элемент управления Table	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
46.	Элемент управления Calendar	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера,</p>

		<p>который переходит с шага на шаг (или с представления на представление) и обратно.</p>
47.	Элемент управления MultiView	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
48.	Элемент управления Wizard	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
49.	Элемент управления RangeValidator	<p>1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений.</p> <p>2) Используйте этот элемент для проверки шаблона введенных пользователем данных.</p> <p>3) Используйте этот элемент для сравнения ввода пользователя.</p> <p>4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.</p>

50.	Элемент управления RegularExpressionValidator	<p>1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений.</p> <p>2) Используйте этот элемент для проверки шаблона введенных пользователем данных.</p> <p>3) Используйте этот элемент для сравнения ввода пользователя.</p> <p>4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.</p>
51.	Элемент управления CompareValidator	<p>1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений.</p> <p>2) Используйте этот элемент для проверки шаблона введенных пользователем данных.</p> <p>3) Используйте этот элемент для сравнения ввода пользователя.</p> <p>4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.</p>
52.	Элемент управления CustomValidator	<p>1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений.</p> <p>2) Используйте этот элемент для проверки шаблона введенных пользователем данных.</p> <p>3) Используйте этот элемент для сравнения ввода пользователя.</p> <p>4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.</p>
53.	Элемент управления Login	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p>

		<p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
54.	Элемент управления PasswordRecovery	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
55.	Элемент управления CreateUserWizard	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
56.	Элемент управления ChangePassword	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p>

		<p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
57.	Элемент управления GridView	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
58.	Элемент управления DataList	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
59.	Элемент управления DetailsView	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>

60.	Элемент управления Form View	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
61.	Элемент управления ListView	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
62.	Элемент управления Repeater	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
63.	Элемент управления DataPager	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время</p>

		<p>таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
64.	Элемент управления Data Source Controls	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
65.	Позиционирование окна формы на экране свойство - CenterParent	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства Form.Location.</p> <p>4) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.</p>
66.	Позиционирование окна формы на экране свойство - CenterScreen	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p>

		<p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.</p>
67.	Позиционирование окна формы на экране свойство - <code>Manual</code>	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.</p>
68.	Позиционирование окна формы на экране свойство - <code>WindowsDefaultBounds</code>	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.</p>
69.	Позиционирование окна формы на экране свойство - <code>WindowsDefaultLocation</code>	<p>1) Центрирует форму внутри текущего экрана дисплея.</p> <p>2) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>3) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.</p> <p>4) Позиционирует форму в положение по умолчанию Windows, размер формы определяется свойством <code>Form.Size</code> (это настройка по умолчанию).</p>
70.	Элемент управления <code>TableLayoutPanel</code>	<p>1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p>

		<p>2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p> <p>3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей.</p> <p>4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.</p>
71.	Элемент управления FlowLayoutPanel	<p>1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p> <p>2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p> <p>3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей.</p> <p>4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.</p>
72.	Элемент управления SplitContainer	<p>1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p> <p>2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p> <p>3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей.</p>

		<p>4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.</p>
73.	Элемент управления ToolStripContainer	<p>1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p> <p>2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p> <p>3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей.</p> <p>4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.</p>
74.	Элемент управления MenuStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
75.	Элемент управления ToolStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
76.	Элемент управления StatusStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p>

		4) Элемент управления для представления данных с иерархическими связями.
77.	Элемент управления TreeView	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
78.	Элемент управления ImageList	<p>1) Работают как провайдер изображений для других элементов управления.</p> <p>2) Для отображения данных в табличном формате.</p> <p>3) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p> <p>4) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p>
79.	Элемент управления DataGridView	<p>1) Работают как провайдер изображений для других элементов управления.</p> <p>2) Для отображения данных в табличном формате.</p> <p>3) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров.</p> <p>4) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.</p>
80.	Базовый класс WPF: visual—	<p>1) этот класс является основным модулем визуализации в WPF.</p> <p>2) этот класс реализует модель многопоточности в WPF.</p> <p>3) это базовый класс для элементов управления в WPF.</p>

		<p>4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).</p>
81.	Базовый класс WPF: DispatcherObject —	<p>1) этот класс является основным модулем визуализации в WPF.</p> <p>2) этот класс реализует модель многопоточности в WPF.</p> <p>3) это базовый класс для элементов управления в WPF.</p> <p>4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).</p>
82.	Базовый класс WPF: Control —	<p>1) этот класс является основным модулем визуализации в WPF.</p> <p>2) этот класс реализует модель многопоточности в WPF.</p> <p>3) это базовый класс для элементов управления в WPF.</p> <p>4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).</p>
83.	Базовый класс WPF: Application—	<p>1) этот класс является основным модулем визуализации в WPF.</p> <p>2) этот класс реализует модель многопоточности в WPF.</p> <p>3) это базовый класс для элементов управления в WPF.</p> <p>4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).</p>
84.	Кнопка Swap в WPF конструкторе	<p>1) меняет местами панель XAML и панель конструирования;</p> <p>2) разделяет панели по вертикали;</p>

		<p>3) разделяет панели по горизонтали;</p> <p>4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).</p>
85.	Кнопка Vertical Split в WPF конструкторе	<p>1) меняет местами панель XAML и панель конструирования;</p> <p>2) разделяет панели по вертикали;</p> <p>3) разделяет панели по горизонтали;</p> <p>4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).</p>
86.	Кнопка Horizontal Split в WPF конструкторе	<p>1) меняет местами панель XAML и панель конструирования;</p> <p>2) разделяет панели по вертикали;</p> <p>3) разделяет панели по горизонтали;</p> <p>4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).</p>
87.	Кнопка Collapse/Expand Panel в WPF конструкторе	<p>1) меняет местами панель XAML и панель конструирования;</p> <p>2) разделяет панели по вертикали;</p> <p>3) разделяет панели по горизонтали;</p> <p>4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).</p>
88.	Панели компоновки WPF Canvas	<p>1) Контейнерный элемент управления без встроенной логики компоновки.</p> <p>2) Панель, которая позволяет выполнять закрепление своих дочерних элементов.</p> <p>3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам.</p> <p>4) Контейнерный элемент управления, который реализует для своих дочерних элементов</p>

		управления размещение в виде вертикальной или горизонтальной стопки.
89.	Панели компоновки WPF DockPanel	<p>1) Контейнерный элемент управления без встроенной логики компоновки.</p> <p>2) Панель, которая позволяет выполнять закрепление своих дочерних элементов.</p> <p>3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам.</p> <p>4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.</p>
90.	Панели компоновки WPF Grid	<p>1) Контейнерный элемент управления без встроенной логики компоновки</p> <p>2) Панель, которая позволяет выполнять закрепление своих дочерних элементов</p> <p>3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам.</p> <p>4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.</p>
91.	Панели компоновки WPF StackPanel	<p>1) Контейнерный элемент управления без встроенной логики компоновки</p> <p>2) Панель, которая позволяет выполнять закрепление своих дочерних элементов</p> <p>3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам.</p> <p>4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.</p>
92.	Элементы управления источниками данных: ObjectDataSource	<p>1) Предоставляет другие классы как источники данных.</p>

		<p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
93.	Элементы управления источниками данных: SqlDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
94.	Элементы управления источниками данных: AccessDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
95.	Элементы управления источниками данных: XmlDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p>

		<p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
96.	Элементы управления источниками данных: SiteMapDataSource	<p>1) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>2) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>3) Предоставляет XML-файл как источник данных.</p> <p>4) Специальный источник данных, который предоставляет карту сайта ASP NET как источник данных.</p>
97.	Web-сервис	<p>1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные.</p> <p>2) используется для представления и описания данных независимым от платформы образом.</p> <p>3) обеспечивает средство локализации в Интернете.</p> <p>4) служит для описания содержимого Web-сервиса и его Web-методов.</p>
98.	XML (Extensible Markup Language, расширяемый язык разметки)	<p>1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные.</p> <p>2) используется для представления и описания данных независимым от платформы образом.</p> <p>3) обеспечивает средство локализации в Интернете.</p> <p>4) служит для описания содержимого Web-сервиса и его Web-методов.</p>
99.	URI (Uniform Resource Identifier)	<p>1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные.</p>

		<p>2) используется для представления и описания данных независимым от платформы образом.</p> <p>3) обеспечивает средство локализации в Интернете.</p> <p>4) служит для описания содержимого Web-сервиса и его Web-методов.</p>
100.	WSDL (Web Service Description Language, язык описания Web-сервиса)	<p>1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные.</p> <p>2) используется для представления и описания данных независимым от платформы образом.</p> <p>3) обеспечивает средство локализации в Интернете.</p> <p>4) служит для описания содержимого Web-сервиса и его Web-методов.</p>

4.3. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ в форме экзамена МДК.02.03 МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Вопросы к экзамену составлены на основе рабочей программы МДК и охватывают ее наиболее актуальные разделы и темы. Контрольно-оценочные материалы целостно отражают объем проверяемых теоретических знаний и практических умений. Билет состоит из 2 вопросов (см. Вопросы для подготовки) и 1 практического задания (см. Задачи).

В критерии оценки уровня подготовки обучающегося входят: уровень освоения обучающимся материала, предусмотренного учебной программой по дисциплине; умения обучающегося использовать теоретические знания при выполнении практических задач; обоснованность, четкость, краткость изложения ответа.

ВОПРОСЫ ДЛЯ ПОДГОТОВКИ

1. Понятие решения. Множество решений, оптимальное решение. Показатель эффективности решения.
2. Математические модели, принципы их построения, виды моделей.
3. Понятие целевой функции.
4. Общий вид и основная задача линейного программирования.
5. Какие задачи линейного программирования можно решить геометрическим методом?
6. В чем заключается признак оптимальности в симплекс методе?
7. Транспортная задача. Методы нахождения начального решения

- транспортной задачи.
8. Общий вид задач нелинейного программирования.
 9. Что означает термин «Транспортный тариф» в решении транспортных задач.
 10. Основные понятия динамического программирования.
 11. Сформулируйте принцип оптимальности Беллмана.
 12. Методы хранения графов в памяти ЭВМ.
 13. Какие бывают графы.
 14. Что такое степень вершины графа.
 15. Какой граф называется деревом.
 16. Как строятся матрицы инцидентности и матрицы смежности.
 17. Системы массового обслуживания: понятия, примеры, модели.
 18. Какой процесс называется марковским.
 19. Какие типы неопределенностей в СМО встречаются.
 20. Виды систем массового обслуживания.
 21. Метод имитационного моделирования. Достоинства и недостатки данного метода.
 22. Метод Монте-Карло.
 23. Основные понятия теории марковских процессов: случайный процесс, марковский процесс, граф состояний, поток событий, вероятность состояния.
 24. Уравнения Колмогорова, финальные вероятности состояний.
 25. Схема гибели и размножения. Метод имитационного моделирования.
 26. Единичный жребий и формы его организации.
 27. Общая постановка задачи нахождения эмпирических формул.
 28. Величина достоверности аппроксимации.
 29. Типы регрессионных зависимостей.
 30. Понятие прогноза. Количественные методы прогнозирования: скользящие средние, экспоненциальное сглаживание, проектирование тренда.
 31. Качественные методы прогноза.
 32. Предмет и задачи теории игр.
 33. Основные понятия теории игр: игра, игроки, партия, выигрыш, проигрыш, ход, личные и случайные ходы, стратегические игры, стратегия, оптимальная стратегия.
 34. Антагонистические матричные игры: чистые и смешанные стратегии.
 35. Методы решения конечных игр.
 36. Область применимости теории принятия решений. Принятие решений в условиях определенности, в условиях риска, в условиях неопределенности.
 37. Нормализация значений полей.
 38. Требования, предъявляемые к исходным данным при построении дерева решений.
 39. Обучающая выборка. Использование обучающей выборки.
 40. Критерии принятия решений в условиях неопределенности. Дерево решений.

ЗАДАЧИ

1. Составить математическую модель следующей задачи. Предположим, что для производства продукции вида А и В можно использовать материал трех сортов. При этом на изготовление единицы изделия А расходуется a_1 кг. первого сорта, a_2 кг. второго сорта и a_3 кг. третьего сорта. На изготовление продукции вида В расходуется b_1 кг. первого сорта, b_2 кг. второго сорта и b_3 кг. третьего сорта. На складе фабрики имеется всего материала первого сорта C_1 кг, второго сорта – C_2 кг, третьего – C_3 кг. От реализации единицы продукции вида А фабрика имеет прибыль m тысяч рублей, а от реализации вида В прибыль составляет n тысяч рублей. Определить максимальную прибыль от реализации всей продукции.

$$\begin{array}{lll} a_1=19 & b_1=26 & C_1=868 \\ a_2=16 & b_2=17 & C_2=638 \\ a_3=19 & b_3=8 & C_3=853 \\ & & m=5 \\ & & n=4 \end{array}$$

Решение задачи произвести, используя ПО компьютера.

2. Составить математическую модель следующей задачи. Предположим, что для производства продукции вида А и В можно использовать материал трех сортов. При этом на изготовление единицы изделия А расходуется a_1 кг. первого сорта, a_2 кг. второго сорта и a_3 кг. третьего сорта. На изготовление продукции вида В расходуется b_1 кг. первого сорта, b_2 кг. второго сорта и b_3 кг. третьего сорта. На складе фабрики имеется всего материала первого сорта C_1 кг, второго сорта – C_2 кг, третьего – C_3 кг. От реализации единицы продукции вида А фабрика имеет прибыль m тысяч рублей, а от реализации вида В прибыль составляет n тысяч рублей. Определить максимальную прибыль от реализации всей продукции.

$$\begin{array}{lll} a_1=14 & b_1=40 & C_1=1200 \\ a_2=15 & b_2=27 & C_2=993 \\ a_3=20 & b_3=4 & C_3=1097 \\ & & m=5 \\ & & n=13 \end{array}$$

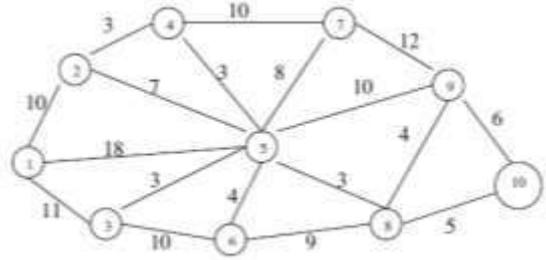
Решение задачи произвести, используя ПО компьютера.

3. Составить математическую модель следующей задачи. Предположим, что для производства продукции вида А и В можно использовать материал трех сортов. При этом на изготовление единицы изделия А расходуется a_1 кг. первого сорта, a_2 кг. второго сорта и a_3 кг. третьего сорта. На изготовление продукции вида В расходуется b_1 кг. первого сорта, b_2 кг. второго сорта и b_3 кг. третьего сорта. На складе фабрики имеется всего материала первого сорта C_1 кг, второго сорта – C_2 кг, третьего – C_3 кг. От реализации единицы продукции вида А фабрика имеет прибыль m тысяч рублей, а от реализации вида В прибыль составляет n тысяч рублей. Определить максимальную прибыль от реализации всей продукции.

$$\begin{array}{lll} a_1=9 & b_1=27 & C_1=606 \\ a_2=15 & b_2=15 & C_2=802 \\ a_3=15 & b_3=3 & C_3=840 \\ & & m=11 \\ & & n=6 \end{array}$$

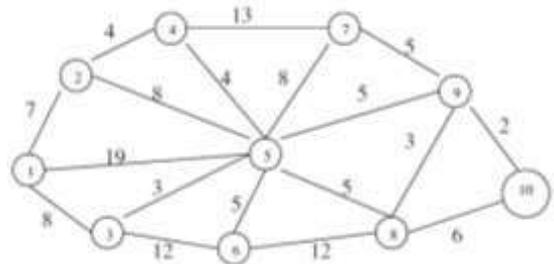
Решение задачи произвести, используя ПО компьютера.

4. По данной схеме, соединяющей 10 точек, найти кратчайшее расстояние от точки 1 до точки 10.



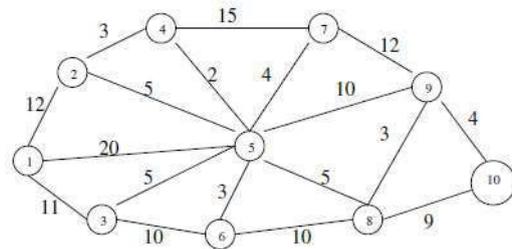
Алгоритм решения записать в виде блок-схемы.

5. По данной схеме, соединяющей 10 точек, найти кратчайшее расстояние от точки 1 до точки 10.



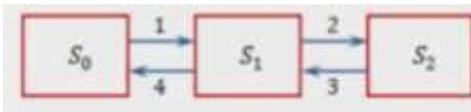
Алгоритм решения записать в виде блок-схемы.

6. По данной схеме, соединяющей 10 точек, найти кратчайшее расстояние от точки 1 до точки 10.



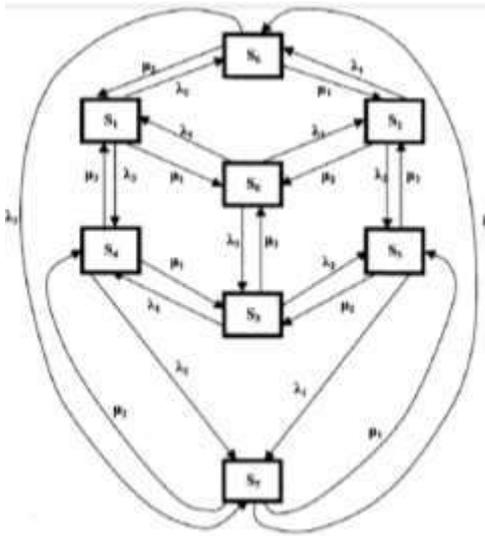
Алгоритм решения записать в виде блок-схемы.

7. Процесс гибели и размножения представлен графом. Найти предельные вероятности состояний.

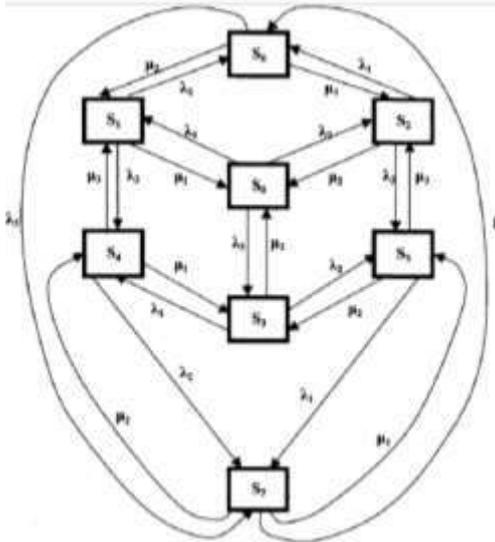


8. Техническое устройство состоит из трех узлов и в любой момент времени может находиться в одном из восьми состояний. Численные значения интенсивности потоков событий равны $\mu_1=1, \lambda_2=2, \lambda_3=2, \mu_1=4, \mu_2=4, \mu_3=4$.

Решение задачи произвести используя ПО компьютера

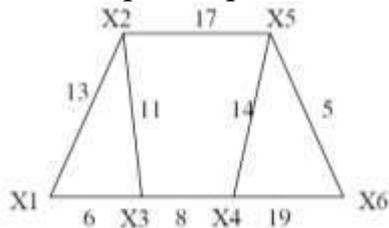


9. Техническое устройство состоит из трех узлов и в любой момент времени может находиться в одном из восьми состояний. Численные значения интенсивности потоков событий равны $\lambda_1=2, \lambda_2=1, \lambda_3=1, \mu_1=4, \mu_2=2, \mu_3=2$.
Решение задачи произвести используя ПО компьютера

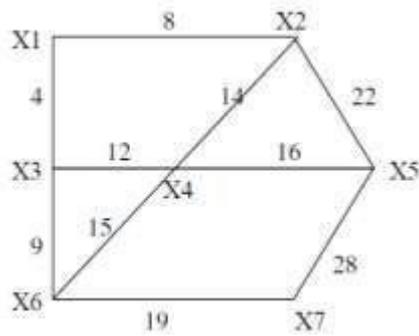


10. АТС имеет 5 линий связи. Поток заявок имеет интенсивность 2 вызова в минуту, а время каждого разговора составляет в среднем 3 минуты. Определить вероятность отказа и вероятность того, что ни одна линия связи не будет занята.
Алгоритм решения задачи представить в виде блок-схемы.

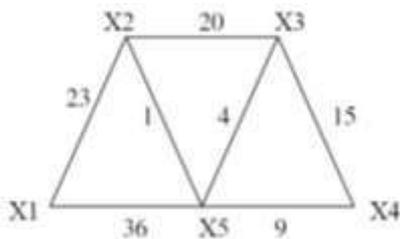
11. Составить матрицы инцидентности и смежности. Найти минимальный остов дерева, кратчайший путь от точки X1 до остальных точек.



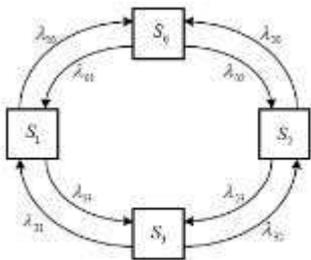
12. Составить матрицы инцидентности и смежности. Найти минимальный остов дерева, кратчайший путь от точки X1 до остальных точек.



13. Составить матрицы инцидентности и смежности. Найти минимальный остов дерева, кратчайший путь от точки X1 до остальных точек.



14. Техническое устройство S состоит из двух узлов, каждый из которых в случайный момент времени может выйти из строя (отказаться), после чего мгновенно начинается ремонт узла, продолжающийся заранее неизвестное случайное время. Возможные состояния системы можно перечислить: S_0 – оба узла исправны; S_1 – первый узел ремонтируется, второй исправен; S_2 – второй узел ремонтируется, первый исправен; S_3 – оба узла ремонтируются. Найти предельные вероятности для системы S при $\lambda_{01}=1, \lambda_{02}=2, \lambda_{10}=2, \lambda_{13}=2, \lambda_{20}=3, \lambda_{23}=1, \lambda_{31}=3, \lambda_{32}=2$.



15. Решить задачу графическим методом на минимум и на максимум

$$x - 2y \rightarrow \min, \max$$

$$\begin{cases} 5x + 3y \geq 30, \\ x - y \leq 3, \\ -3x + 5y \leq 15, \\ x \geq 0, \quad y \geq 0. \end{cases}$$

При создании графиков использовать ПО.

16. Решить графическим методом ЗЛП, заданную указанной математической моделью.

$$F = 2x_1 - x_2 \rightarrow \max,$$

$$\begin{cases} x_1 \leq 3, \\ x_1 \geq -1, \\ -2x_1 - 3x_2 \leq 6, \\ -x_1 + 2x_2 \leq 6. \end{cases}$$

При создании графиков использовать ПО.

17. Решите графически следующие задачи линейного программирования

$$F = x_1 + 2x_2 \rightarrow \max,$$

$$\begin{cases} x_1 + 2x_2 \leq 6, \\ 2x_1 + x_2 \leq 8, \\ x_2 \leq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

При создании графиков использовать ПО.

18. Решить задачу линейного программирования симплекс-методом.

$$f = 2X_1 + X_2 - 2X_3 \rightarrow \min$$
$$\begin{cases} X_1 + X_2 - X_3 \geq 8, \\ X_1 - X_2 + 2X_3 \geq 2, \\ -2X_1 - 8X_2 + 3X_3 \geq 1, \\ X_i \geq 0 (i = 1, 2, 3). \end{cases}$$

19. Решить задачу линейного программирования симплексным методом:

$$F = -3x_1 + x_2 + 4x_3 \rightarrow \max,$$

$$\begin{cases} -x_2 + x_3 + x_4 = 1 \\ -5x_1 + x_2 + x_3 = 2 \\ -8x_1 + x_2 + 2x_3 - x_5 = 3 \end{cases}$$

$$x_1 \dots x_5 \geq 0.$$

20. Составить математическую модель транспортной задачи и решить её методом потенциалов. Завод имеет 3 цеха А, В, С и 4 склада №1,2,3,4. Цех А производит 30 тыс.штук изделий, цех В – 40 тыс. штук изделий, С – 20 тыс. штук изделий. Пропускная способность склада №1 - 20 тыс. штук изделий, №2 - 30 тыс. штук изделий, №3 – 30 тыс.штук, №4 – 10 тыс. штук. Стоимость перевозки из цеха А соответственно в склады №1,2,3,4 1 тыс. штук изделий составляет 20, 30, 3, 4 р., из цеха В 1 тыс. – соответственно 3, 20, 5, 1 р., а из цеха С – соответственно 4, 30, 2, 6 р. Составить такой план перевозок изделий, при котором расходы на перевозку 90 тыс. изделий были бы наименьшими.

4.4. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ в форме зачета МДК.02.04 ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ БИЗНЕСА

Зачет проводится в форме защиты рефератов.

ТЕМЫ РЕФЕРАТОВ

1. Разработка программного комплекса для обработки сведений по учету программно-аппаратного обеспечения предприятия.
2. Разработка программного комплекса для обработки сведений по учету работы печатного издательства.
3. Разработка программного комплекса для обработки сведений по работе медицинского учреждения.
4. Разработка программного комплекса для обработки сведений по учету работы отдела кадров предприятия.
5. Разработка программного комплекса для обработки сведений по учету работы гольф-клуба.
6. Разработка информационно-справочной системы для обработки сведений по отпуску льготных лекарств в аптеке.
7. Разработка программного комплекса для обработки сведений по учету аудиторного фонда института.
8. Разработка программного комплекса для обработки сведений по восхождению в альпинистском клубе.
9. Разработка программного комплекса для обработки сведений по учету самолетных рейсов.
10. Разработка программного комплекса для обработки сведений по учету работы оптового склада.
11. Разработка приложения для покупки, аренды и обслуживания автомобиля.
12. Разработка программного комплекса для обработки сведений по учету работы библиотеки.
13. Разработка программного комплекса для обработки сведений по учету работы страховой компании.
14. Разработка приложения учета кассет в частной коллекции.
15. Разработка системы тестирования знаний студентов по дисциплине технологии программирования.
16. Разработка системы анализа спортивных достижений в клубных матчах по футболу.
17. Разработка программного комплекса электронного автомобильного музея.
18. Разработка информационно-справочной системы расписания поездов.
19. Разработка элемента информационной системы станции тех. обслуживания.
20. Разработка справочной информационной системы книжного магазина.

21. Разработка информационно-справочной системы "Телефонный справочник".
22. Разработка информационно-справочной системы "Железнодорожные билеты".
23. Разработка программы автоматизации учета книг в домашней библиотеке.
24. Разработка приложения учета продаж железнодорожных билетов.
25. Разработка программного комплекса для обработки базы данных турагентства.
26. Разработка приложения ведения счетов по пластиковым картам.
27. Разработка программного комплекса электронного вызова сотрудников технической поддержки.
28. Разработка программного комплекса учёта физических лиц в Налоговой инспекции.
29. Разработка приложения автоматизации рабочего места администратора гостиницы.
30. Разработка программного комплекса подбора и расчета комплектующих компьютера.

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ МДК

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
Раздел модуля 1. Анализ и проектирование программных решений		
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	<p>Оценка «отлично» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры.</p> <p>Оценка «хорошо» - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры.</p> <p>Оценка «удовлетворительно» - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен/диф.зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p>
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
Раздел модуля 2. Технологии тестирования программных модулей		

<p>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Оценка «отлично» - выполнена отладка модуля с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка «хорошо» - выполнена отладка модуля сохранены и представлены результаты отладки.</p> <p>Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.4 Выполнять тестирование программных модулей</p>	<p>Оценка «отлично» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия, сделан вывод о достаточности тестового пакета.</p> <p>Оценка «хорошо» - выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия.</p> <p>Оценка «удовлетворительно» - выполнено тестирование модуля и оформлены результаты тестирования. выполнено функциональное тестирование, выполнена и представлена оценка тестового покрытия с некоторыми погрешностями.</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Оценка «отлично» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации</p>

	<p>Оценка «хорошо» - определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p> <p>Оценка «удовлетворительно» - определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</p>	<p>методами рефакторинга.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
Раздел модуля 3. Технологии разработки мобильных приложений		
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.	Оценка «отлично» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке	Экзамен/д.зачет в форме собеседования: практическое задание по созданию модуля для заданного

	<p>работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.</p> <p>Оценка «хорошо» - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие выполняемых функций спецификации с незначительными отклонениями.</p> <p>Оценка «удовлетворительно» - разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.</p>	<p>мобильного устройства на основе спецификации</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
Раздел модуля 4. Системное программирование		
<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно-ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/д.зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 1.3 Выполнять отладку программных модулей с</p>	<p>Оценка «отлично» - выполнена отладка модуля с пояснением особенностей</p>	<p>Экзамен/д.зачет в форме собеседования:</p>

использованием специализированных программных средств	<p>отладочных классов; сохранены и представлены результаты отладки.</p> <p>Оценка «хорошо» - выполнена отладка модуля сохранены и представлены результаты отладки.</p> <p>Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам</p>	<ul style="list-style-type: none"> • обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач 	<p>Экспертное наблюдение за выполнением работ</p>
<p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности</p>	<p>- использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач</p>	
<p>ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях</p>	<p>- демонстрация ответственности за принятые решения</p> <p>- обоснованность самоанализа и коррекция результатов собственной работы;</p>	
<p>ОК 04. Эффективно взаимодействовать и работать в коллективе и команде</p>	<p>- 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>ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p>	<p>Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей</p>	

<p>ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения</p>	<p>- соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик, - соблюдение стандартов антикоррупционного поведения</p>	
<p>ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях</p>	<p>- эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности</p>	
<p>ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности</p>	<p>- эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.</p>	
<p>ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках</p>	<p>- эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту;</p>	

Критерии оценивания знаний и умений обучающихся, сформированности общих и профессиональных компетенций

3.1. 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.

3.2. 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.

3.3. 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

3.4. 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.

3.5 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