

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

РАССМОТРЕНО
на заседании Педагогического совета
протокол от «16» февраля 2026 г.
№ 4

УТВЕРЖДАЮ:
Директор АНПОО СЦК при ЗУИЭП"


А.С. Волков
«16» февраля 2026 г.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ
И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ОП.06 Основы алгоритмизации и программирования**

по специальности
09.02.11 Разработка и управление программным обеспечением
квалификация «Программист»

форма обучения: заочная

форма промежуточной аттестации по дисциплине: экзамен

Пермь 2026

СОДЕРЖАНИЕ

1. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ	3
2. КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ.....	6
3. КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	10
4. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ	18

1. ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

1.1. Общая характеристика оценочных материалов.

Оценочные материалы (ОМ) разработан с целью установления соответствия образовательных достижений студентов требованиям программы подготовки специалистов среднего звена по дисциплине ОП.06 Основы алгоритмизации и программирования 09.02.11 Разработка и управление программным обеспечением.

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

Характеристика текущего контроля

Целью текущего контроля является контроль запланированных по дисциплине знаний и умений. Текущий контроль по дисциплине проводится в следующих формах: тестирование, практическое занятие и других формах.

Текущий контроль проводится в соответствии с рабочей программой по дисциплине и структурных элементов электронного курса в СДО. Периодичность проведения текущего контроля зависит от сроков выполнения заданий обучающимися в пределах установленного учебным планом семестра.

Текущий контроль проводится только с помощью электронного обучения и дистанционных образовательных технологий. Текущий контроль включает в себя оценку выполнения практической работы и тестирования по каждому разделу тематического плана учебной дисциплины. Порядок проведения текущего контроля определяется оценочными средствами.

Характеристика промежуточной аттестации

Целью контроля промежуточной аттестации является контроль освоения запланированных по дисциплине знаний и умений. Форма промежуточной аттестации: экзамен. Промежуточная аттестация с помощью электронного обучения и дистанционных образовательных технологий.

Контрольно-измерительные материалы промежуточной аттестации состоят из:

1) теоретической части, которая выполняется письменно при проведении промежуточной аттестации обучающихся с применением дистанционных образовательных технологий, представляет собой ответ на теоретический вопрос по программе курса;

2) тестовой части, которая выполняется письменно и включает в себя:

- задание типа А (базовый уровень) - 10 тестовых вопросов с 4 вариантами ответов;
- задание типа В (задания повышенной сложности) - 2 задания с 4 вариантами ответов;
- задание типа С (практико-ориентированное комплексное задание) - 1 задание с 4 вариантами ответов.

Первая часть (А) теста включает 10 заданий базового уровня сложности. Данные задания направлены на выявление сформированности полученных в ходе изучения дисциплины базовых знаний, соответствующих заданным стандартам компетенциям. Каждое тестовое задание данной части содержит 4 варианта ответов, один или несколько из которых является правильным.

Вторая часть (В) теста состоит из 2 заданий повышенного уровня сложности. Данные задания направлены на выявление усвоения основных алгоритмов профессиональной деятельности, соответствующих заданным стандартам компетенциям.

Практическая часть (С) состоит из 1 практико-ориентированного задания повышенного уровня сложности (задачи, ситуации).

Для каждого варианта теста подготовлены эталоны ответов для проверки результатов теста.

Данные задания направлены на выявление уровня сформированности опыта практической деятельности, соответствующего заданным стандартам компетенциям, и

выполняются с учетом смоделированной ситуации.

1.2. Требования к результатам освоения учебной дисциплины

Результатом освоения учебной дисциплины являются сформированные умения и приобретенные знания:

Код ПК, ОК	Умения	Знания
ОК 01	– распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части	– актуальный профессиональный и социальный контекст, в котором приходится работать и жить
ОК 02	– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации	– номенклатура информационных источников, применяемых в профессиональной деятельности
ОК 03	– применять современную научную профессиональную терминологию	– современная научная и профессиональная терминология
ОК 04	– взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности	– психологические особенности личности
ОК 05	– грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке	– правила оформления документов
ОК 06	– демонстрировать осознанное поведение	– традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений
ОК 07	– соблюдать нормы экологической безопасности	– правила экологической безопасности при ведении профессиональной деятельности
ОК 08	– пользоваться средствами профилактики перенапряжения, характерными для данной специальности	– средства профилактики перенапряжения
ОК 09	– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и	– правила построения простых и сложных предложений на профессиональные темы

	бытовые), понимать тексты на базовые профессиональные темы	
ПК 2.2	– разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий – применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей – анализировать требования и определять функциональность модуля – создавать интерфейсы для взаимодействия с другими модулями и системами – обеспечивать безопасность, производительность и масштабируемость при разработке модулей – оптимизировать проектируемые модули для повышения их эффективности и качества – работать с системой контроля версий – улучшать производительность модулей, выявляя и устраняя узкие места – проводить анализ и мониторинг производительности приложений – применять инструменты для рефакторинга и оптимизации программного кода	– язык программирования, основные конструкции, синтаксис – паттерны проектирования – структуры данных – принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP – работа с инструментальным программным обеспечением – методы оптимизации кода и алгоритмов – эффективные алгоритмы и структуры данных для повышения производительности – многопоточность в программных модулях – методы оптимизации сетевых протоколов для ускорения обмена данными – кэширование данных – управление памятью – техники повышения производительности программного обеспечения
ПК 2.4	– анализировать требования к программному обеспечению и составлять планы тестирования. – создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям. – выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования. – анализировать результаты тестирования и документировать найденные ошибки.	– принципы и методы тестирования программного обеспечения. – основы программирования и архитектуры программного обеспечения. – основы баз данных и SQL-запросов. – инструменты для автоматизации тестирования – основы разработки и отладки программного обеспечения на

– разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении. – выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования – использовать системы контроля дефектов ПО – составлять отчет о выполнении тестирования ПО	- разных языках программирования – понятие дефекта программного обеспечения – критерии качества ПО – виды и типы тестирования ПО – техники ручного тестирования – техники автоматизированного тестирования – жизненный цикл дефекта ПО – принципы работы в системе контроля дефектов – основные понятия о качестве ПО
--	---

2. КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Практические задания

1. В одномерном массиве, состоящем из n вещественных элементов, вычислить: произведение положительных элементов массива; сумму элементов массива, расположенных до минимального элемента.
2. В одномерном массиве, состоящем из n вещественных элементов, вычислить: сумму отрицательных элементов массива. Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.
3. В одномерном массиве, состоящем из n вещественных элементов, вычислить: номер минимального по модулю элемента массива. Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные 0, считать положительными).
4. В одномерном массиве, состоящем из n целых элементов, вычислить сумму элементов массива, расположенных после последнего элемента, равного нулю. Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.
5. В одномерном массиве, состоящем из n вещественных элементов, вычислить: количество элементов массива, больших C ; произведение элементов массива, расположенных после максимального элемента.
6. Дана строка. Вывести строку, содержащую те же символы, но расположенные в обратном порядке.
7. Дана строка S и число N . Преобразовать строку S в строку длины N следующим образом: если длина строки S больше N , то отбросить первые символы, если длина строки S меньше N , то в ее начало добавить символы "." (точка).

8. Даны строки S1, S2 и символ C. После каждого вхождения символа C в строку S1 вставить строку S2.
9. Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов в строке.
10. Дана строка-предложение на русском языке. Подсчитать количество содержащихся в строке знаков препинания.

2.2. Тестовые задания

Вопрос 1. Экземпляр класса носит название

- а) функция
- б) **объект**
- в) метод
- г) процедура

Вопрос 2. К необязательным элементам синтаксиса описания метода следует отнести

- а) **атрибуты**
- б) имя метода
- в) спецификаторы
- г) модификаторы

Вопрос 3. Переменные или константы, которые получают значения, передаваемые методу при вызове, носят название

- а) контейнеры
- б) методы
- в) **параметры**
- г) атрибуты

Вопрос 4. При передаче параметра по значению метод получает

- а) ссылки на место в памяти
- б) модули параметров
- в) **копии параметров**
- г) указатели

Вопрос 5. Соотношение между членами последовательности, в котором каждый следующий член выражается через несколько предыдущих, носит название

- а) импликационное
- б) **рекуррентное**
- в) детерминированное
- г) итерационное

Вопрос 6. К типам рекурсии следует отнести

- а) **прямую рекурсию**
- б) возвратную рекурсию
- в) косвенную рекурсию
- г) циклическую рекурсию

Вопрос 7. Объект, генерирующий информацию о «необычном программном происшествии», носит название

- а) **исключение**
- б) событие
- в) контейнер
- г) ошибка

Вопрос 8. К стандартным исключениям следует отнести

- а) инверсию операторов
- б) **деление на нуль**

- в) попадание индекса вне диапазона определения
- г) синтаксические ошибки

Вопрос 9. В каком пространстве имен определен класс Exception?

- а) **System**
- б) Main
- в) Detect
- г) Base

Вопрос 10. Чтобы указать, что некоторое выражение должно быть проконтролировано на предмет переполнения, используется ключевое слово

- а) set
- б) **checked**
- в) mode
- г) try

Вопрос 11. Параметром оператора throw является

- а) **объект**
- б) метод
- в) свойство
- г) значение

Вопрос 12. Механизм обработки исключительных ситуаций носит название

- а) мониторинг
- б) **обработка исключений**
- в) контроль ошибок
- г) отладка

Вопрос 13. Блок кода, который может вызвать исключение, размещается в блоке

- а) finally
- б) catch
- в) **try**
- г) throw

Вопрос 14. Блок обработки исключений располагается в блоке

- а) try
- б) finally
- в) **catch**
- г) using

Вопрос 15. Блок, выполняющийся всегда после try и catch, называется

- а) **finally**
- б) end
- в) close
- г) finish

Вопрос 16. Оператор, создающий новое исключение, обозначается ключевым словом

- а) catch
- б) try
- в) finally
- г) **throw**

Вопрос 17. Базовый класс для всех исключений в .NET называется

- а) Error
- б) Fault
- в) **Exception**
- г) BaseException

Вопрос 18. Свойство, возвращающее текстовое описание исключения, называется

- а) code
- б) type

в) **Message**

г) Info

Вопрос 19. Свойство, содержащее информацию о стеке вызовов при исключении, называется

а) trace

б) log

в) **StackTrace**

г) history

Вопрос 20. Механизм передачи параметров в метод без указания их типов называется

а) типизация

б) **varargs**

в) полиморфизм

г) обобщение

Вопрос 21. Способ передачи параметров, при котором метод получает ссылку на переменную, называется

а) по значению

б) **по ссылке**

в) по копированию

г) по указателю

Вопрос 22. Механизм автоматического преобразования типов при передаче параметров называется

а) кастинг

б) **приведение типов**

в) конвертация

г) трансформация

Вопрос 23. Метод, имеющий то же имя, что и другой метод в том же классе, называется

а) **перегруженным**

б) переопределенным

в) перекрытым

г) замещенным

Вопрос 24. Свойство метода, определяющее его доступность из других классов, называется

а) тип

б) **модификатор доступа**

в) спецификатор

г) атрибут

Вопрос 25. Метод, который может быть переопределен в производном классе, должен быть объявлен с модификатором

а) private

б) **virtual**

в) static

г) sealed

Вопрос 26. Метод, переопределяющий виртуальный метод базового класса, должен быть объявлен с модификатором

а) override

б) **override**

в) virtual

г) new

Вопрос 27. Метод, который должен быть обязательно реализован в производных классах, объявляется с модификатором

а) virtual

б) **abstract**

в) override

г) sealed

Вопрос 28. Метод, который нельзя переопределить в производных классах, объявляется с модификатором

а) virtual

б) abstract

в) **sealed**

г) override

Вопрос 29. Механизм, позволяющий методу использовать различные реализации в зависимости от типа объекта, называется

а) инкапсуляция

б) наследование

в) **полиморфизм**

г) абстракция

Вопрос 30. Тип параметра, позволяющий методу принимать любое количество аргументов, называется

а) array

б) **params**

в) dynamic

г) generic

3. КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

3.1. Теоретические вопросы промежуточной аттестации

1. Понятие алгоритма. Свойства алгоритма. Способы описания алгоритма. Виды алгоритмов.
2. Язык программирования как способ описания алгоритмов.
3. История развитие языков программирования. Области применения языков программирования. Стандарты.
4. Основные этапы решения задач на компьютере.
5. Компиляторы и интерпретаторы.
6. Структурный и объектный подходы к программированию.
7. Технология разработки ПО NET FRAMEWORK.
8. Среда разработки Visual Studio.Net. Установка и создание программы в среде Visual Studio.Net на языке C#.
9. Назначение и область применения языка C#.
10. Структура программы в языке C#.
11. Ввод и вывод данных. Операторы присваивания. Составной оператор
12. Данные и типы данных в C#. Простые типы данных.
13. Переменные: определение, правила именования.
14. Константы (литералы): определение, виды и правила записи в программе.
15. Операции и выражения. Правила формирования и вычисления выражений.
16. Математические функции (класс Math).
17. Разветвляющиеся алгоритмы и их описание на языке C#. Условный оператор. Оператор выбора. Операторы перехода: break, goto, return.
18. Циклы: цикл с параметром, циклы с предусловием и постусловием. Принудительный выход из цикла: операторы break и continue. Бесконечные циклы.

19. Массивы в языке C#: определение, виды. Объявление одномерного массива. Варианты инициализации.
20. Ввод и вывод одномерных массивов.
21. Оператор цикла `foreach`. Создание и обработка одномерного массива.
22. Понятие о процедурах и функциях. Функции, определяемые пользователем. Общее определение функции в языке C#. Вызов функции. Возвращение значений. Выход из функции.
23. Параметры функции. Передача параметров по значению и по ссылке. Модификатор `out`.
24. Массив параметров и ключевое слово `params`. Массив в качестве параметра.
25. Область видимости (контекст) переменных. Рекурсивные функции.
26. Строки. Стандартные методы для работы со строками.
27. Понятие потока. Классы и библиотеки .NET для работы с потоками. Понятие файла. Виды доступа к файлам. Объект `FileStream`.
28. Классы `StreamWriter` и `StreamReader`.
29. Байтовый поток и методы работы с ним.
30. Символьный поток.
31. Двоичные потоки.
32. Стандартные методы для работы с вводом выводом. Переопределение потоков
33. Основные операции при работе с файлами. Способы работы с файлами. Обработка файлов.
34. Основные принципы объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция.
35. Понятие класса и объекта. Характеристики объекта: поля, свойства, методы, события.
36. Классы и объекты. Описание класса. Доступ к классам и объектам. Модификаторы доступа к элементам класса: `public`, `private`, `protected`, `internal`.
37. Метод: понятие, правила записи. Вызов метода. Передача параметров по значению. Правило триединого соответствия параметров и аргументов: по количеству, типам и по порядку следования. Создание методов, возвращающих значения. Способы размещения методов. Перегрузка методов.
38. Конструкторы и деструкторы в языке C#. Разработка программ с использованием конструкторов.
39. Инкапсуляция как управление доступом к данным. Свойства класса: понятие, виды, правила записи.
40. Наследование и полиморфизм. Иерархия классов. Синтаксис наследования.
41. Скрытие и перекрытие методов. Ключевые слова `virtual`, `override`.
42. Вызов методов базового класса («родителя»): ключевое слово `base`. Особенности использования конструкторов в иерархически связанных между собой классах.
43. Восходящее и нисходящее приведение.
44. Наследование в интерфейсах. Сходства и различия интерфейсов, абстрактных классов и обычных классов.
45. Стандартные интерфейсы .NET: `Comparable`, `ICloneable`, `IEnumerable`.
46. Назначение и синтаксис структуры. Элементы тела структуры. Сходства и различия структур и классов.
47. Тип данных `enum`: объявление и особенности использования. Операции с перечислениями.
48. Понятие коллекции. Классы, возможности, правила работы, встроенные методы.
49. Делегаты: понятие, правила описания. Примеры и особенности использования делегатов.
50. Обработка события: автоматическое создание обработчиков.

3.2 Тестовая часть промежуточной аттестации

Тестовые задания типа А (Выберите один верный ответ)

Вопрос 1. Что такое конструктор класса?

- а) **Специальный метод для создания объектов**
- б) Обычный метод класса
- в) Свойство класса
- г) Событие класса

Вопрос 2. Какой модификатор доступа обеспечивает самый строгий уровень инкапсуляции?

- а) public
- б) protected
- в) internal
- г) **private**

Вопрос 3. Что такое инкапсуляция?

- а) Наследование свойств от базового класса
- б) **Соккрытие деталей реализации**
- в) Создание новых классов
- г) Переопределение методов

Вопрос 4. Какой оператор используется для приведения типов?

- а) as
- б) is
- в) **cast**
- г) typeof

Вопрос 5. Что такое абстрактный класс?

- а) Класс без методов
- б) **Класс, который нельзя создать напрямую**
- в) Класс без свойств
- г) Класс только для наследования

Вопрос 6. Какой модификатор используется для создания статического метода?

- а) virtual
- б) override
- в) **static**
- г) abstract

Вопрос 7. Что такое полиморфизм?

- а) Создание новых классов
- б) **Способность объектов одного класса использовать методы других классов**
- в) Соккрытие данных
- г) Наследование свойств

Вопрос 8. Какой оператор используется для проверки типа объекта?

- а) cast
- б) **is**
- в) as
- г) typeof

Вопрос 9. Что такое интерфейс?

- а) **Набор методов без реализации**
- б) Класс с реализацией
- в) Абстрактный класс
- г) Структура данных

Вопрос 10. Какой модификатор используется для реализации интерфейса?

- а) implements
- б) **interface**

- в) extends
- г) inherits

Вопрос 11. Что такое делегат?

- а) **Тип, который может ссылаться на метод**
- б) Специальный метод
- в) Событие
- г) Свойство

Вопрос 12. Какой оператор используется для обработки исключений?

- а) try-finally
- б) catch-throw
- в) **try-catch-finally**
- г) try-throw

Вопрос 13. Что такое событие?

- а) Специальный метод
- б) **Механизм уведомления о происходящих действиях**
- в) Делегат
- г) Свойство класса

Вопрос 14. Какой модификатор используется для создания свойства?

- а) get-set
- б) **property**
- в) field
- г) attribute

Вопрос 15. Что такое обобщение (generics)?

- а) Создание массивов
- б) **Использование параметризованных типов**
- в) Приведение типов
- г) Наследование классов

Вопрос 16. Какой модификатор используется для создания индексатора?

- а) index
- б) **this**
- в) indexer
- г) get-set

Вопрос 17. Что такое sealed-класс?

- а) Абстрактный класс
- б) **Класс, который нельзя наследовать**
- в) Статический класс
- г) Интерфейс

Вопрос 18. Какой модификатор используется для создания константы?

- а) const
- б) **readonly**
- в) static
- г) constant

Вопрос 19. Что такое атрибут?

- а) **Специальная метка для добавления метаданных**
- б) Свойство класса
- в) Метод класса
- г) Событие

Вопрос 20. Какой модификатор используется для создания частичного класса?

- а) partial
- б) **partial**
- в) split
- г) fragment

Вопрос 21. Что такое коллекция?

- а) **Структура данных для хранения множества элементов**
- б) Массив фиксированного размера
- в) Список методов
- г) Набор свойств

Вопрос 22. Какой интерфейс реализует перечислимые коллекции?

- а) IEnumerable
- б) **IEnumerable<T>**
- в) ICollection
- г) IList

Вопрос 23. Что такое лямбда-выражение?

- а) **Сокращенная форма записи делегата**
- б) Специальный метод
- в) Анонимный метод
- г) Событие

Вопрос 24. Какой оператор используется для создания анонимного типа?

- а) new
- б) **var**
- в) anonymous
- г) dynamic

Вопрос 25. Что такое LINQ?

- а) **Технология запросов к данным**
- б) Язык программирования
- в) Фреймворк
- г) Библиотека классов

Вопрос 26. Какой оператор используется для объединения коллекций?

- а) join
- б) **union**
- в) merge
- г) combine

Вопрос 27. Что такое асинхронное программирование?

- а) **Программирование с использованием async/await**
- б) Последовательное выполнение задач
- в) Синхронное выполнение методов
- г) Однопоточное выполнение

Вопрос 28. Какой оператор используется для асинхронного ожидания?

- а) wait
- б) **await**
- в) async
- г) delay

Вопрос 29. Что такое Task в C#?

- а) **Объект, представляющий асинхронную операцию**
- б) Синхронный метод
- в) Обычный класс
- г) Интерфейс

Вопрос 30. Какой метод используется для запуска задачи в фоновом режиме?

- а) Run
- б) Execute
- в) **Start**
- г) Begin

Тестовые задания типа В (Установите соответствие между элементами)

Задание 1. Сопоставление модификаторов доступа и их описания
Установите соответствие между модификатором доступа и его описанием:

Модификатор Описание

1. public	А. Доступ только внутри сборки
2. private	Б. Доступ из любого места
3. protected	В. Доступ только внутри класса
4. internal	Г. Доступ из производного класса

Ответ: 1-Б, 2-В, 3-Г, 4-А

Задание 2. Сопоставление типов исключений и их примеров
Установите соответствие между типом исключения и примером его возникновения:

Тип исключения	Пример
1. ArithmeticException	А. Выход за пределы массива
2. NullReferenceException	Б. Деление на ноль
3. IndexOutOfRangeException	В. Обращение к null-объекту
4. InvalidCastException	Г. Неверное приведение типов

Ответ: 1-Б, 2-В, 3-А, 4-Г

Задание 3. Сопоставление концепций ООП и их определений
Установите соответствие между концепцией ООП и её определением:

Концепция	Определение
1. Наследование	А. Соккрытие деталей реализации
2. Инкапсуляция	Б. Создание новых классов на основе существующих
3. Полиморфизм	В. Множественная реализация методов
4. Абстракция	Г. Выделение общих характеристик

Ответ: 1-Б, 2-А, 3-В, 4-Г

Задание 4. Сопоставление типов коллекций и их особенностей
Установите соответствие между типом коллекции и её характеристикой:

Коллекция	Характеристика
1. List	А. Упорядоченная коллекция с индексами
2. Dictionary	Б. Коллекция пар “ключ-значение”
3. HashSet	В. Неупорядоченная коллекция уникальных элементов
4. Queue	Г. Коллекция по принципу FIFO

Ответ: 1-А, 2-Б, 3-В, 4-Г

Задание 5. Сопоставление операторов LINQ и их функций
Установите соответствие между оператором LINQ и его назначением:

Оператор	Функция
1. Where	А. Фильтрация элементов
2. Select	Б. Преобразование элементов
3. OrderBy	В. Сортировка по возрастанию
4. GroupBy	Г. Группировка элементов

Ответ: 1-А, 2-Б, 3-В, 4-Г

Задание 6. Сопоставление типов параметров и их особенностей

Установите соответствие между типом параметра и его характеристикой:

Тип параметра	Характеристика
1. ref	А. Передача по ссылке
2. out	Б. Выходной параметр
3. params	В. Массив параметров
4. value	Г. Передача по значению

Ответ: 1-А, 2-Б, 3-В, 4-Г

Задание 7. Сопоставление типов делегатов и их применения

Установите соответствие между типом делегата и его назначением:

Делегат	Назначение
1. Action	А. Метод без возвращаемого значения
2. Func	Б. Метод с возвращаемым значением
3. Predicate	В. Метод-предикат
4. EventHandler	Г. Обработчик событий

Ответ: 1-А, 2-Б, 3-В, 4-Г

Задание 8. Сопоставление типов циклов и их особенностей

Установите соответствие между типом цикла и его характеристикой:

Цикл	Характеристика
1. for	А. Цикл с предусловием
2. while	Б. Цикл со счетчиком
3. do-while	В. Цикл с постусловием
4. foreach	Г. Цикл по коллекции

Ответ: 1-Б, 2-А, 3-В, 4-Г

Тестовые задания типа С (Выбор правильного варианта ответа из предложенного списка)

Задание 1

Задача: Вам нужно создать программу, которая будет хранить информацию о книгах в библиотеке. Какой тип коллекции лучше использовать для хранения списка книг?

- а) Массив фиксированного размера
- б) Стек
- в) **Список (List)**
- г) Очередь

Задание 2

Задача: При разработке программы возникла ошибка “Деление на ноль”. Как правильно обработать эту ситуацию?

- а) Игнорировать ошибку
- б) Использовать try-catch только в конце программы
- в) **Обработать ошибку в блоке try-catch там, где возможно деление**
- г) Завершить программу при возникновении ошибки

Задание 3

Задача: Необходимо создать программу для расчета стоимости покупок. Какой модификатор доступа лучше использовать для метода расчета суммы?

- а) private
- б) protected
- в) internal
- г) **public**

Задание 4

Задача: Нужно создать класс “Прямоугольник”. Какой метод лучше использовать для вычисления площади?

- а) Статический метод без параметров
- б) Метод с внешними переменными
- в) **Метод с параметрами ширины и высоты**
- г) Свойство без параметров

Задание 5

Задача: Требуется создать программу для работы с разными типами фигур. Какой механизм ООП лучше использовать?

- а) Только интерфейсы
- б) Только наследование
- в) **Абстрактный класс с виртуальными методами**
- г) Отдельные классы без связей

Задание 6

Задача: Необходимо сохранить данные пользователя. Какой тип данных лучше использовать для хранения пароля?

- а) Строка (string)
- б) **Защищенная строка (SecureString)**
- в) Числовой тип
- г) Массив символов

Задание 7

Задача: Нужно создать программу для работы с файлами. Какой метод лучше использовать для чтения файла?

- а) Чтение всего файла сразу
- б) **Построчное чтение**
- в) Чтение в обратном порядке
- г) Случайный доступ без проверки

Задание 8

Задача: Требуется создать программу для подсчета количества слов в тексте. Какой тип параметра лучше использовать для метода подсчета?

- а) ref-параметр
- б) out-параметр
- в) **value-параметр**
- г) params-параметр

Задание 9

Задача: Необходимо создать программу для работы с базой данных. Какой паттерн лучше использовать для подключения?

- а) Observer
- б) **Singleton**

- в) Factory
- г) Proxy

Задание 10

Задача: Требуется создать программу для сортировки списка чисел. Какой алгоритм сортировки лучше использовать для небольшого массива?

- а) Быстрая сортировка
- б) **Сортировка вставками**
- в) Слияние
- г) Пирамидальная сортировка

4. ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ

Результаты обучения	Показатели освоения компетенций	Методы оценки
Знать: - Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. - номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; - создание модулей программного обеспечения на различных языках программирования - Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. - Подпрограммы, составление библиотек подпрограмм - Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного	Владение профессиональной терминологией Умение работать с информационными источниками Использование основных алгоритмических конструкций Разработка модулей программного обеспечения на языке программирования, используя структуры данных, Разработка модулей программного обеспечения, используя принципы объектно-ориентированного программирования Решение ситуационных задач Отладка и тестирование программного обеспечения	Экспертное наблюдение выполнения практических работ и видов работ по практике Тестирование Контрольная работа Опрос

программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения - отладки программного обеспечения на уровне программных модулей - тестирования программного обеспечения Уметь: - распознавать задачу, анализировать и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи - Разрабатывать алгоритмы для конкретных задач - определять задачи для поиска информации; определять необходимые источники информации; применять средства информационных технологий для решения профессиональных задач; - Использовать программы для графического отображения алгоритмов - разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий - выполнять тестирование программного обеспечения		
--	--	--

Критериями оценки преподавателем *результатов практических заданий* обучающегося являются:

- степень овладения обучающимся учебного материала;

- правильность выполнения задания;
- оформление работ в соответствии с требованиями;
- способность обучающегося использовать теоретический запас знаний на практике;
- способность обучающегося использовать электронные образовательные ресурсы, осуществлять поиск нужной информации, осваивать её и применять на практике;
- наличие навыков выявления проблемы, формирования решений, критической оценки собственных решений;
- обоснованность изложения ответа;
- навыки анализа и предъявления вариантов действий в ситуационных задачах;
- навыки формирования собственной позиции, её оценки и аргументирования;
- при выполнении заданий, носящих творческий характер, на оценку влияет неординарность подхода к решению поставленной задачи и наличие собственного мнения.

Критерии оценивания практических работ и практических заданий

- оценка «отлично» ставится, если обучающийся самостоятельно выполнил все этапы решения задач на компьютере; работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы.

- оценка «хорошо» ставится, если работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с компьютером в рамках поставленной задачи; правильно выполнена большая часть работы (свыше 85 %); работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

- оценка «удовлетворительно» ставится, если работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы на компьютере, требуемыми для решения поставленной задачи.

- оценка «неудовлетворительно» ставится, если допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями работы на компьютере или значительная часть работы выполнена не самостоятельно.

Контрольное задание в форме тестов

Процент результативности (правильных ответов)	Оценка уровня подготовки	
	балл (отметка)	вербальный аналог
100 - 90	5	отлично
89 - 80	4	хорошо
79 - 70	3	удовлетворительно
Менее 70	2	неудовлетворительно