

Министерство просвещения РФ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Глазовский государственный инженерно-педагогический университет
имени В.Г. Короленко»

Утверждена
на заседании ученого совета университета

«21» апреля 2025 г. протокол № 9
Приказ № 45 от 21 апреля 2025 г.

Ректор Я.А. Чиговская-Назарова

**АДАПТИРОВАННАЯ РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
(для лиц с нарушениями функций опорно-двигательного аппарата)**

ЯЗЫК ПРОГРАММИРОВАНИЯ C#

Уровень основной профессиональной образовательной программы	Бакалавриат
Направление подготовки	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника
Форма обучения	Очная
Семестр(ы)	3

Глазов 2025

1. Цель и задачи изучения дисциплины

1.1. Цель и задачи изучения дисциплины

Целью изучения дисциплины является формирование способности использовать основные концептуальные положения объектно-ориентированного и визуального направлений программирования, методы, способы и средства разработки программ в рамках дисциплины «Язык программирования C#».

Задачи изучения дисциплины:

- сформировать умения использовать основные концептуальных положения объектно-ориентированного и визуального направлений программирования;
- сформировать умения применять методы, способы и средства разработки программ в рамках объектно-ориентированного и визуального направлений программирования.

Программа адаптирована для лиц с нарушениями опорно-двигательного аппарата (ОДА) с учетом их психофизического развития, индивидуальных возможностей и необходимых специальных условий обучения.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными индикаторами достижения компетенций

Код компетенции	ПК-1
Формулировка компетенции	Способен разрабатывать и отлаживать программный код
Индикатор достижения компетенции	ИПК 1.1 Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования ИПК 1.2 Умеет: применять выбранные языки программирования для написания программного кода ИПК 1.3 Владеет: методами создания программного кода в соответствии с техническим заданием, его отладки и оформления в соответствии с установленными требованиями

1.3. Воспитательная работа

Направление воспитательной работы	Типы задач	Формы работы
формирование у обучающихся осознания социальной значимости своей будущей профессии, мотивации к осуществлению профессиональной деятельности	производственно-технологический	включение в социокультурную среду путем формирования у студентов практических умений и навыков в рамках профессиональной деятельности
научно-исследовательская работа обучающихся	производственно-технологический	Исследовательская деятельность студентов (публикация статей, выступление с докладом)

1.4. Место дисциплины в структуре образовательной программы

Дисциплина «Язык программирования C#» относится к обязательной части учебного плана.

1.5. Особенности реализации дисциплины

Дисциплина реализуется на русском языке.

Для лиц с нарушениями функций ОДА используется электронное обучение, дистанционные технологии. Для поддержки курса используется сайт: <http://moodle.ggpi.org>.

2. Объем дисциплины

Вид учебной работы по семестрам	Всего, зачетных единиц	Академ. часы	Из них в форме практической подготовки
Общая трудоемкость дисциплины	4	144	
СЕМЕСТР 3			
Контактная работа с преподавателем:			
Аудиторные занятия (всего)		72	
Занятия лекционного типа		16	
Лабораторные работы		-	
Занятия семинарского типа		-	
Практические занятия		48	
КСР		8	
Самостоятельная работа обучающихся		72	
Вид промежуточной аттестации: Зачет с оценкой		0	

3. Содержание дисциплины

3.1. Разделы дисциплины и виды занятий (тематический план занятий)

№ п/п	Разделы и темы дисциплины Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в академических часах)						
		всего	ауд	лекц	пр	лаб	КСР	СРС
Семестр 3								
1.	Основы объектно-ориентированного программирования	12	8	2	6	–	–	8
2.	Интегрированная среда разработки	12	8	2	6	–	–	8
3.	Этапы разработки приложения	12	8	2	6	–	–	8
4.	Создание классов	12	8	2	6	–	–	8
5.	Иерархия классов	12	10	2	6	–	2	10
6.	Визуальное событийно-управляемое программирование	16	10	2	6	–	2	10
7.	Разработка оконного приложения	16	10	2	6	–	2	10
8.	Шаблоны проектирования	16	10	2	6	–	2	10
Итого – по дисциплине		144	72	16	48	–	8	72

3.2. Занятия лекционного типа

Для лиц с нарушениями функций ОДА лекция сопровождается текстом с увеличенным шрифтом или усиливающей звуковой аппаратурой.

Занятия, при возможности, проводятся в мультимедийной аудитории, где имеется возможность подкрепления основных положений лекционного материала необходимым иллюстративным материалом (письменная презентация ключевых вопросов, являющихся темой обсуждения во время беседы; использование необходимых электронных видеоматериалов для иллюстрирования вопросов и контекста обсуждаемой проблемы, и т.п.). Есть возможность предоставлять необходимый учебный материал электронно для последующей самостоятельной работы с ним.

При объяснении материала мысли излагаются четко и лаконично (в простые предложения), информация подается в виде небольших логически и по смыслу законченных фрагментов.

СЕМЕСТР 3

Лекция 1

Тема. Основы объектно-ориентированного программирования

Жизненные циклы объектов. Состояние, событие, действие. Правила переходов в состояния. Таблица переходов в состояния. Жизненные циклы связей, конкурирующие связи. Модель взаимодействия объектов. Модели процессов, процессы и потоки данных. Модель доступа к объектам, диаграммы потоков данных действий. Рабочие продукты объектно-ориентированного анализа.

Лекция 2

Тема. Интегрированная среда разработки

Требования к аппаратным и программным средствам интегрированной среды разработки. Интерфейс среды разработки: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.

Лекция 3

Тема. Этапы разработки приложения

Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.

Лекция 4

Тема. Создание классов

Понятия метод, свойство, событие. Понятия инкапсуляция, наследование, полиморфизм. Проектирование класса. Графическое представление класса. Члены класса: поля, конструкторы, методы. Модификаторы доступа.

Лекция 5

Тема. Иерархия классов

Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Перегрузка методов.

Лекция 6

Тема. Визуальное событийно-управляемое программирование

Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Дополнительные элементы управления. Свойства компонентов

(элементов управления). Виды свойств. Синтаксис определения свойств. Категория свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий. Процедуры, определённые пользователем: синтаксис, передача аргументов. Вызов событий.

Лекция 7

Тема. Разработка оконного приложения

Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание процедур обработки событий. Компиляция и запуск приложения.

Лекция 8

Тема. Шаблоны проектирования

Классификация шаблонов проектирования. Шаблоны генерации. Структурные шаблоны. Шаблоны поведения. Шаблон модель, представление, контроллер.

3.3. Занятия семинарского типа

Учебным планом не предусмотрены

3.4. Практические занятия

Для лиц с нарушениями функций ОДА материал в электронном виде можно найти по адресу: <http://moodle.ggpi.org>.

Выполнение практических работ проводятся в микрогруппах или парами, в которых присутствует смешанный состав обучающихся: в паре – один обычный обучающийся и один обучающийся с двигательным нарушением; микрогруппа включает одного обучающегося с двигательным нарушением и несколько обычных обучающихся.

В ходе практического занятия используются следующие методы:

- опора на определенные и точные понятия;
- использование для иллюстрации конкретных примеров;
- применение вопросов для мониторинга понимания;
- разделение изучаемого материала на небольшие логические блоки;
- увеличение доли конкретного материала и соблюдение принципа от простого к сложному при объяснении материала.

СЕМЕСТР 3

Практическое занятие 1

Тема. Основы объектно-ориентированного программирования

Перечень заданий.

Настройка среды и параметров проекта.

Проектирование объектно-ориентированного приложения.

Создание интерфейса пользователя.

Программирование приложения.

Тестирование, отладка приложения.

Создание документации.

Выполнение проекта.

Управление выполнением программы.

Используя общий диалог, написать программу, позволяющую сохранять указанный пользователем текст в файле и считывать сохраненный текст из файла.

Создать проект, позволяющий добавлять записи в простой список и удалять их.

Разработать класс, объекты которого обладают методами, позволяющими заполнить одномерный массив случайными неповторяющимися числами, вывести его на экран и упорядочить.

Практическое занятие 2

Тема. Основы объектно-ориентированного программирования

Перечень заданий.

Код окна приложения.

Код главного программного модуля.

Код классов приложения.

Файлы ресурсов.

Автоматическая генерация кода средой разработки.

Файлы сборки и исполняемые файлы.

Практическое занятие 3

Тема. Основы объектно-ориентированного программирования

Перечень заданий.

Формат XML в файлах проекта.

Добавление элементов в проект.

Построить график функции в выбранной пользователем системе координат.

Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.

Практическое занятие 4

Тема. Интегрированная среда разработки

Перечень заданий.

Тестирование корректности ввода данных.

Автоматизация тестирования.

Составление тестовых наборов данных.

Программа для генерации тестовых наборов данных.

Разработать класс, объекты которого обладают методами, позволяющими вывести на экран в любую его точку правильный многоугольник произвольного размера с произвольным количеством вершин.

Разработать класс, объекты которого обладают методами, позволяющими вывести на экран график функции. График должен выводиться в прямоугольную область указанного размера.

Объект должен иметь свойство, определяющее величину единичного отрезка.

Практическое занятие 5

Тема. Интегрированная среда разработки

Перечень заданий.

Основные элементы управления интегрированной среды разработки, их состав и назначение.

Дополнительные элементы управления.

Свойства компонентов.

Виды свойств.

Синтаксис определения свойств.

Практическое занятие 6

Тема. Интегрированная среда разработки

Перечень заданий.

Категория свойств.

Назначение свойств и их влияние на результат.

Управление объектом через свойства.

Составить программу, генерирующую цвет по трем его составляющим (красный, зеленый, синий).

Написать программу, моделирующую движение броуновской частицы.

Практическое занятие 7

Тема. Этапы разработки приложения

Перечень заданий.

Программирование приложения.

Тестирование, отладка приложения.

Создание документации.

Обзор основных компонентов.

Компонент кнопка.

Компонент надпись.

Компонент поле ввода.

Компонент флажок.

Компонент переключатель.

Компонент поле со списком.

Контейнер компонентов - объединение элементов.

Компонент полоса прокрутки.

Компонент список с флажками.

Компонент главное меню.

Создать проект, реализующий часы с «будильником».

Разработать класс, объекты которого фиксируют время своего существования.

Практическое занятие 8

Тема. Этапы разработки приложения

Перечень заданий.

Диалоговые окна открытия и сохранения файлов.

Диалоговое окно установки шрифта.

Диалоговое окно выбора основного и фоновых цветов.

Компонент панель инструментов.

Компонент строка состояния.

Набор вкладок.

Практическое занятие 9

Тема. Этапы разработки приложения

Перечень заданий.

Класс исключений.

Стандартные механизмы обработки исключений.

Блок try ... catch ... finally.

Примеры обработки исключений.

Разработать проект, моделирующий арифметический калькулятор. Калькулятор должен выполнять следующие операции: сложение, вычитание, умножение, деление, возведение в квадрат и извлечение квадратного корня как для целых, так и для вещественных чисел.

Проект содержит три текстовых окна. Используя буфер обмена, организовать обмен информацией между приложениями.

Практическое занятие 10

Тема. Создание классов

Перечень заданий.
Организация доступа к закрытым полям класса.
Понятие интерфейса класса.
Абстрактные классы.
Способы создания объектов - экземпляров классов.
Массивы объектов.
Списки объектов.

Практическое занятие 11

Тема. Создание классов

Перечень заданий.
Передача параметров по ссылке и по значению.
Модификаторы `ref` и `out`.
Область видимости переменных и других идентификаторов.
Виртуальные методы.

Практическое занятие 12

Тема. Создание классов

Перечень заданий.
Перегрузка методов.
Вызовы методов.
Статические методы.
Передача массива в качестве параметров, модификатор `params`.

Практическое занятие 13

Тема. Иерархия классов

Перечень заданий.
Интерфейс класса и его реализация.
Физическое представление иерархии классов в заголовочных файлах.
Разработать класс, объекты которого обладают методами, позволяющими вывести на экран анимацию вращающегося шарика. Координата точки – оси вращения, скорость вращения, радиус вращения и направление вращения задаются.
Используя объектно-ориентированный подход, написать программу, моделирующую движение броуновских частиц.

Практическое занятие 14

Тема. Иерархия классов

Перечень заданий.
Создание библиотеки классов.
Перегрузка методов.
Разработать компонент «Таймер», выводящий время, прошедшее с момента запуска таймера. Компонент должен иметь метод, возвращающий текущее время, обрабатывать событие, связанное с включением и выключением таймера и метод, обнуляющий таймер.

Практическое занятие 15

Тема. Иерархия классов

Перечень заданий.
Графическое представление иерархии классов.
Описать класс, объекты которого связываются с текстовыми окнами, хранят введенную пользователем в окно информацию, обладают методом сортировки символов в текстовых

окнах в алфавитном порядке и методом восстанавливающим текст в окне после сортировки.

Практическое занятие 16

Тема. Визуальное событийно-управляемое программирование

Перечень заданий.

Синтаксис определения свойств.

Категория свойств.

Назначение свойств и их влияние на результат.

Управление объектом через свойства.

Практическое занятие 17

Тема. Визуальное событийно-управляемое программирование

Перечень заданий.

Добавление формы в проект.

Свойства и события формы. Обработка событий формы.

Практическое занятие 18

Тема. Визуальное событийно-управляемое программирование

Перечень заданий.

Передача фокусам между формами.

Сворачивание, сокращение и закрытие окна приложения.

Практическое занятие 19

Тема. Разработка оконного приложения

Перечень заданий.

Разработка функциональной схемы работы приложения.

Создание процедур обработки событий.

Компиляция и запуск приложения.

Разработать компонент, представляющий собой окно, в котором можно рисовать окружности. Компонент должен иметь свойство, определяющее цвет «графического пера».

Создать проект, моделирующий работу секундомера. Секундомер должен иметь циферблат с минутной и секундной стрелками, метку и одну командную кнопку.

Практическое занятие 20

Тема. Разработка оконного приложения

Перечень заданий.

Окна вывода сообщений.

Окна ввода данных.

Диалоговые окна открытия и сохранения файлов.

Использование масок для списка обрабатываемых файлов.

Диалоговые окна установки шрифта и задания цвета компонента.

Создание пользовательских диалоговых окон.

События от мыши и клавиатуры.

Событие Click.

Добавление обработчика.

Практическое занятие 21

Тема. Разработка оконного приложения

Перечень заданий.

Аргументы, передаваемые в параметры обработчика.

Разработать компонент, предлагающий интерфейс для ввода данных и вывода решения квадратного уравнения.
Разработайте проект, имитирующий работу простейшего графического редактора.

Практическое занятие 22

Тема. Шаблоны проектирования

Перечень заданий.

Реализуйте на языке C# пример применения паттерна Стратегия по учебнику «Паттерны проектирования».

Компонент главное меню.

Диалоговые окна открытия и сохранения файлов.

Диалоговое окно установки шрифта.

Диалоговое окно выбора основного и фоновых цветов.

Компонент панель инструментов.

Компонент строка состояния.

Набор вкладок.

Шаблон модель, представление, контроллер.

Практическое занятие 23

Тема. Шаблоны проектирования

Перечень заданий.

Реализуйте на языке C# пример применения паттерна Наблюдатель по учебнику «Паттерны проектирования».

Разработка диаграммы состояний.

Диаграмма прецедентов для оконного приложения.

Практическое занятие 24

Тема. Шаблоны проектирования

Перечень заданий.

Разработать компонент, предлагающий интерфейс для отображения одномерного массива и имеющий методы для заполнения, очистки и сортировки.

Разработать компонент «Переключатель», имеющий пять стационарных состояний.

Компонент должен иметь свойство для установки состояния и обрабатывать событие, связанное с выбором другого состояния.

3.5. Лабораторные работы

Учебным планом не предусмотрены

3.6. Контроль самостоятельной работы

Для лиц с нарушениями функций ОДА материал в электронном виде можно найти по адресу: <http://moodle.ggpi.org>.

Для лиц с нарушениями функций опорно-двигательного аппарата учебно-методическое обеспечение для контроля самостоятельной работы обучающихся по дисциплине предъявляется (по выбору обучающегося): устно, письменно на бумаге или на компьютере, в форме тестирования, электронных тренажеров и т.п.

Конкретные формы и виды самостоятельной работы обучающихся с нарушениями функций ОДА устанавливаются преподавателем с учетом индивидуальных психофизических особенностей. При необходимости обучающимся предоставляется дополнительное время для консультаций и выполнения заданий.

Самостоятельная работа включает следующие виды деятельности: работа с книгой и другими источниками информации, планы-конспекты; реферативные

(воспроизводящие), реконструктивно-вариативные, эвристические, творческие самостоятельные работы; проектные работы; дистанционные технологии.

Уделяется внимание индивидуальной работе. Под индивидуальной работой подразумевается две формы взаимодействия с преподавателем: индивидуальная учебная работа (консультации), т.е. дополнительное разъяснение учебного материала и углубленное изучение материала с теми обучающимися, которые в этом заинтересованы, и индивидуальная воспитательная работа. Индивидуальные консультации по предмету становятся важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся с нарушениями функций ОДА.

СЕМЕСТР 3

Контроль самостоятельной работы 1

Тема. Иерархия классов

Перечень заданий.

Создать консольное приложение простейший калькулятор.

Создать консольное приложение, которое выводит справочную информацию.

Оформить исходный код с помощью комментариев.

Создать проект, позволяющий сохранять текст, набранный в текстовом окне в файле данных, очищать текстовое окно и загружать в него сохраненный текст.

Имеется десять флажков. Создать проект, позволяющий сохранять состояние флажков, обнулять и восстанавливать сохраненное состояние.

Организовать анимацию процесса движения шарика по окружности.

Разработать класс, объекты которого обладают методами, позволяющими вывести на экран график функции. График должен выводиться в прямоугольную область указанного размера. Объект должен иметь свойство, определяющее величину единичного отрезка.

Контроль самостоятельной работы 2

Тема. Визуальное событийно-управляемое программирование

Перечень заданий.

Программа должна выводить в консоль натуральные числа от 1 до N, где N - константа, заданная в программе.

Создать простое оконное приложение с обработчиком событий от кнопки.

В приложении 5 кнопок, при щелчке по каждой кнопке в окне на компоненте pictureBox рисуется одна фигура: линия, прямоугольник, треугольник, эллипс, дуга.

Контроль самостоятельной работы 3

Тема. Разработка оконного приложения

Перечень заданий.

Создать проект, позволяющий построить треугольник по координатам вершин.

Построить на экране правильный N-угольник. Количество вершин и радиус окружности, на которой они расположены, указываются пользователем.

Составить программу, моделирующую работу светофора.

Решить задачу о десяти флажках с помощью циклов и указателей на экземпляры класса CheckBox.

В окне приложения разместите в столбик 5 кнопок с надписями button1, ..., button5. С помощью цикла for each выведите номера кнопок. Цикл выполняется в обработчике 1-й кнопке. Объясните полученный результат.

Контроль самостоятельной работы 4

Тема. Шаблоны проектирования

Перечень заданий.

Создать проект, позволяющий построить треугольник по координатам вершин.
Построить на экране правильный N-угольник. Количество вершин и радиус окружности, на которой они расположены, указываются пользователем.
Составить программу, моделирующую работу светофора.
Решить задачу о десяти флажках с помощью циклов и указателей на экземпляры класс CheckBox.
В окне приложения разместите в столбик 5 кнопок с надписями button1, ..., button5. С помощью цикла for each выведите номера кнопок. Цикл выполняется в обработчике 1-й кнопке. Объясните полученный результат.

3.7. Самостоятельная работа студентов

Рекомендуемые формы самостоятельной работы студентов: перечислить не менее 3 форм работы, используемые для реализации дисциплины. Формы работы можно взять из указаний «Методические рекомендации по организации образовательного процесса при освоении дисциплины».

4. Фонд оценочных средств

Формы текущего контроля, промежуточной аттестации и послитоговый контроль для лиц с нарушениями функций ОДА устанавливаются с учетом их психофизиологических особенностей. При необходимости все виды аттестации проходит в несколько этапов.

Текущий контроль результатов обучения осуществляется преподавателем в процессе проведения практических занятий и лабораторных работ, а также выполнения индивидуальных работ и домашних заданий, или в режиме тренировочного тестирования в целях получения информации о выполнении обучаемым требуемых действий в процессе учебной деятельности; правильности выполнения требуемых действий; соответствии формы действия данному этапу усвоения учебного материала; формировании действия с должной мерой обобщения, освоения и т.д.

Формы и сроки проведения промежуточного контроля определяются преподавателем с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.).

Для лиц с нарушениями функций опорно-двигательного аппарата формами текущего контроля, промежуточной аттестации и послитогового контроля используются (в зависимости от индивидуальных особенностей и потребностей):

- устный ответ;
- письменный ответ;
- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

При проведении всех форм контроля учитываются психофизическое развитие и ограничения здоровья. Время выполнения заданий для лиц с нарушениями функций ОДА может быть увеличено, но не более чем на 30 минут.

Для лиц с нарушениями опорно-двигательного аппарата материалы ко всем видам аттестации предъявляться (в зависимости от индивидуальных особенностей и потребностей):

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

Рекомендуемые формы контроля и оценки результатов обучения лиц с нарушением функций ОДА:

- письменная проверка с использованием специальных технических средств (альтернативных средств ввода, управления компьютером и др.): контрольные, графические работы, тестирование, домашние задания, эссе, письменные коллоквиумы, отчеты и др.;
- устная проверка, с использованием специальных технических средств (средств коммуникаций): дискуссии, тренинги, круглые столы, собеседования, устные коллоквиумы и др.;
- с использованием компьютера и специального ПО (альтернативных средств ввода и управления компьютером и др.): работа с электронными образовательными ресурсами, тестирование, рефераты, курсовые проекты, графические работы, дистанционные формы предпочтительнее обучающимся, ограниченным в передвижении и др.

ФОС включает оценочные средства текущего, промежуточного и итогового контроля (Приложение 1).

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

5.1. Основная литература

1. Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 245 с. — ISBN 978-5-4497-0862-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/146386.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
2. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2020. — 369 с. — (Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/450868> (дата обращения: 31.03.2025).

5.2. Дополнительная литература

1. Агапов, В. П. Основы программирования на языке C# : учебное пособие / В. П. Агапов. — Москва : Московский государственный строительный университет, ЭБС АСВ, 2012. — 128 с. — ISBN 978-5-7264-0576-6. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/16366.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
2. Васильев, В. Н. Основы программирования на языке C# : учебное пособие / В. Н. Васильев. — Волгоград : Волгоградский институт бизнеса, 2010. — 70 с. — ISBN 978-5-9061-7234-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/11341.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
3. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2020. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/450999> (дата обращения: 31.03.2025).

4. Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 192 с. — (Высшее образование). — ISBN 978-5-534-12338-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/451467> (дата обращения: 31.03.2025).
5. Котов, О. М. Язык C#. Краткое описание и введение в технологии программирования : учебное пособие / О. М. Котов. — Екатеринбург : Уральский федеральный университет, ЭБС АСВ, 2014. — 208 с. — ISBN 978-5-7996-1094-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/68524.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей
6. Туральчук, К. А. Параллельное программирование с помощью языка C# : учебное пособие / К. А. Туральчук. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 189 с. — ISBN 978-5-4497-2476-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/133969.html> (дата обращения: 31.03.2025). — Режим доступа: для авторизир. пользователей

1. Обучающиеся с нарушениями функций опорно-двигательного аппарата обеспечены печатными и электронными ресурсами в форме, адаптированной к ограниченным возможностям здоровья и восприятия информации:

- в печатной форме
- в форме электронного документа
- в форме аудиофайла

2. Каждому обучающемуся с нарушениями функций ОДА обеспечен доступ к библиотечным ресурсам и сети Интернет и предоставлен не менее чем одним учебным, методическим и (или) электронным изданием в форме, адаптированной к ограничениям здоровья.

3. Для обучения лиц с нарушениями функций ОДА комплектация библиотечного фонда осуществляется электронными изданиями основной и дополнительной литературы по дисциплинам.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине

6.1 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

1. <https://sites.google.com/site/ifizmat/prog> – Программирование, сайт преподавателя
2. <https://metanit.com/cpp/tutorial/> – Руководство по языку программирования C++.
3. <http://citforum.ru/programming/> – Программирование.
4. <https://www.onlinegdb.com> – Online GDB is online compiler and debugger for C/C++. You can compile, run and debug code with gdb online. Using gcc/g++ as compiler and gdb as debugger. Currently C, C++, C#, Java, Python 3 languages are supported.
5. <https://sites.google.com/view/ifizmat/devcsharp> – Язык программирования C#, сайт преподавателя

6.2. Перечень необходимых профессиональных баз данных и информационных справочных систем

Электронная библиотечная система «IPR SMART». Режим доступа: <http://www.iprbookshop.ru>

Электронная библиотечная система «Юрайт». Режим доступа: <https://urait.ru>

Электронно-библиотечная система «Лань» (раздел «Сетевая электронная библиотека педагогических вузов»). Режим доступа: <https://e.lanbook.com>

7. Методические указания и учебно-методическое обеспечение для обучающихся по освоению дисциплины

Дисциплина реализуется в соответствии с указаниями «Методические рекомендации по организации образовательного процесса при освоении дисциплины», размещенными в ЭИОС университета (eios.ggpi.org).

Методические рекомендации для работы с инвалидами и лицами с ОВЗ размещены в ЭИОС университета (eios.ggpi.org).

8. Материально-техническая база, программное обеспечение, необходимое для осуществления образовательного процесса по дисциплине

Учебный корпус 1, аудитории(я) 235, 219.

Полный перечень материально-технической базы и программного обеспечения размещены в ЭИОС университета (eios.ggpi.org).

Образовательная среда организации, организация рабочих мест обучающихся, технические и программные средства общего и специального назначения соответствуют Методическим рекомендациям по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащённости образовательного процесса (утв. Министерством образования и науки РФ 8 апреля 2014 г. N АК-44/05вн), а именно:

- наличие компьютерной техники, адаптированной для инвалидов со специальным программным обеспечением, альтернативных устройств ввода информации и других технических средств приема-передачи учебной информации в доступных формах для студентов с нарушениями опорно-двигательного аппарата;

- для студентов с нарушениями функций опорно-двигательного аппарата используются альтернативные устройств ввода информации (при необходимости);

- используются специальные возможности операционной системы Windows, такие как экранная клавиатура, с помощью которой можно вводить текст, настройка действий Windows при вводе с помощью клавиатуры или мыши.

Для студентов с нарушениями функций опорно-двигательного аппарата предусмотрено расположение рабочих мест в первых рядах у окна и в среднем ряду.

9. Рейтинг-план оценки успеваемости студентов

Дисциплина / Семестр	Объем аудиторной работы				Перечень контрольных мероприятий	Максимально е кол-во баллов	Поощрение	Штрафы	Итоговая форма отчета (мин. балл)
	лк	Прак	Лаб.	КСР					
Язык программирования C# / 3	16	48	—	8	1. Контроль посещаемости лекций 2. Контроль посещаемости практических занятий и КСР 3. Работа на практическом занятии и занятии КСР <u>Контрольные мероприятия</u> 1. Тестирование 2. Контрольная работа <u>Компенсационные мероприятия</u> Создание программного продукта по теме, определяемой преподавателем	16 56 140 5 5 10	не применяются	не применяются	Допуск к зачету – 50%, «автомат» при зачете – 90%
ВСЕГО						232			

Лист регистрации изменений и дополнений к РПД
(фиксируются изменения и дополнения перед началом учебного года,
при необходимости внесения изменений на следующий год –
оформляется новый лист изменений)

№ п.п.	Содержание изменения	Дата, номер протокола заседания кафедры. Подпись заведующего кафедрой	Дата, номер протокола заседания совета факультета. Подпись декана факультета
1.			
2.			
3.			
4.			
5.			
6.			

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ ЯЗЫК ПРОГРАММИРОВАНИЯ C#

1. Фонд оценочных средств для текущего контроля успеваемости, промежуточной аттестации и послитогового контроля по дисциплине

1.1. Настоящий Фонд оценочных средств(ФОС) по дисциплине «Язык программирования C#» является неотъемлемым приложением к рабочей программе дисциплины «Язык программирования C#» (РПД). На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

1.2. Оценивание всех видов контроля(текущего, промежуточного, послитогового) осуществляется по 5-ти балльной шкале.

1.3. Результаты оценивания текущего контроля учитываются в рейтинге.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными индикаторами достижения компетенций

Код компетенции	ПК-1
Формулировка компетенции	Способен разрабатывать и отлаживать программный код
Индикатор достижения компетенции	ИПК 1.1 Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования ИПК 1.2 Умеет: применять выбранные языки программирования для написания программного кода ИПК 1.3 Владеет: методами создания программного кода в соответствии с техническим заданием, его отладки и оформления в соответствии с установленными требованиями

3. Содержание оценочных средств текущего контроля и критерии их оценивания

3.1. Текущий контроль осуществляется преподавателем дисциплины при проведении занятий в следующих формах: тестирование, контрольная работа.

3.2. Формы текущего контроля и критерии их оценивания.

Форма контроля 1 - Типовые тестовые задания

Проверяемые компетенции и индикаторы достижения компетенций: ПК-1: ИПК-1.1, ИПК 1.2, ИПК-1.3.

Время выполнения заданий: 25 минут

Критерии оценивания:

- верные ответы на 90% вопросов – «отлично»;
- верные ответы на 70% вопросов – «хорошо»;
- верные ответы на 50% вопросов – «удовлетворительно»;
- меньше 50% ответов на вопросы – «неудовлетворительно».

1. Visual C# основан на языке...
 - а) Паскаль;
 - б) Ада;
 - в) Фортран;
 - г) Си++.
2. Что такое Visual Studio?
 - а) компилятор;
 - б) интегрированная среда разработки программ;
 - в) машинный язык;
 - г) операционная система.
3. Где взять кнопку, которую мы хотим установить на форму?
 - а) в главном меню;
 - б) в окне свойств;
 - в) на панели элементов;
 - г) в обозревателе объектов.
4. Что произойдет в результате выполнения кода: `MessageBox.Show("Test MessageBox");`
 - а) этот текст появится в заголовке окна;
 - б) этот текст появится на метке в окне формы;
 - в) произойдет ошибка в программе;
 - г) этот текст выйдет в сообщении в отдельном окне.
5. Какое свойство кнопки отвечает за надпись на кнопке?
 - а) Location;
 - б) Text;
 - в) Visible;
 - г) Name.
6. Установите соответствие между операторами языка C# и назначением

1 ++	а) Исключающее ИЛИ
2 --	б) Логическое И
3 ^	в) Декремент
4 &&	г) Инкремент
7. Установите соответствие между значением и типом данных в языке C#

1 -123	а) unsigned char
2 1	б) int
3 true	в) double
4 4.669	г) bool

Форма контроля 2 – Типовая контрольная работа

Проверяемые компетенции и индикаторы достижения компетенций: ПК-1: ИПК-1.1, ИПК-1.2, ИПК-1.3.

Время выполнения заданий: 45 минут

Критерии оценивания:

Обучающимся предлагается выполнить серию из трех заданий.

Для получения оценки «удовлетворительно» необходимо выполнить 1 задание.

Для получения оценки «хорошо» необходимо выполнить 2 задания.

Для получения оценки «отлично» необходимо выполнить все задания.

1. Создать проект, позволяющий построить треугольник по координатам вершин.
2. Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.
3. Организовать анимацию процесса движения шарика по окружности.

3.3 Методические указания по проведению процедуры текущего контроля

1. Текущий контроль проводится на протяжении всего семестра.
2. Сбор, обработка и оценивание результатов текущего контроля проводятся преподавателем, ведущим дисциплину.
3. Предъявление результатов оценивания осуществляется в течение недели после проведения контрольного мероприятия.
4. Результаты текущего контроля учитываются в рейтинге по дисциплине.
5. Все материалы, полученные от обучающихся в ходе текущего контроля (контрольная работа, диктант, тест, организация дискуссии, круглого стола, доклад, реферат, отчет по лабораторной работе, отчет по педагогической практике и т.п.), должны храниться в течение текущего семестра на кафедрах.
6. Считать, что положительные результаты текущего контроля свидетельствуют об успешном процессе формирования указанных компетенций индикаторов достижения компетенций (этапов формирования компетенций).

4. Содержание оценочных средств промежуточной аттестации и критерии их оценивания

- 4.1. Промежуточная аттестация проводится в виде: зачета с оценкой (3 сем.).
- 4.2. Содержание оценочного средства. Проверяемые компетенции и индикаторы достижения компетенций: ПК-1: ИПК-1.1, ИПК 1.2, ИПК-1.3.

Примерные вопросы и задания к зачету

Теоретические вопросы

1. Язык программирования высокого уровня общего назначения Visual C#. Общая характеристика: назначение и области применения. Виды приложений C#. Инструменты разработки. Возможности кроссплатформенной разработки. Стиль кодирования. Правила оформления кода.
2. Интегрированная среда разработки Visual Studio. Требования к аппаратным и программным средствам интегрированной среды разработки. Интерфейс среды разработки: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. Компиляция и запуск приложения.
3. Разработка оконного приложения. Структура проекта в Visual C#. Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Создание функций обработки событий. Понятие точки входа программы.
4. Типы данных, переменные и константы в Visual C#. Синтаксис объявления, использование, размещение в памяти.
5. Операторы и компоненты ввода—вывода данных. Назначение, принципы использования, основные свойства.

6. Передача параметров по ссылке и по значению. Модификаторы `ref` и `out`. Область видимости переменных и других идентификаторов. Передача массива в качестве параметров, модификатор `params`.
7. Основные понятия объектно-ориентированного программирования. Жизненные циклы объектов. Состояние, событие, поведение. Правила переходов в состояния. Таблица переходов в состояния. Жизненные циклы связей, конкурирующие связи. Модель взаимодействия объектов. Модели процессов, процессы и потоки данных. Модель доступа к объектам. Объектно-ориентированный анализа.
8. Иерархия классов. Классы объектно-ориентированного языка программирования: виды, назначение, свойства, методы, события. Объявление класса, свойств и методов экземпляра класса. Наследование. Полиморфизм.
9. Инкапсуляция. Организация доступа к закрытым полям класса. Понятие интерфейса класса. Абстрактные классы. Статические классы. Способы создания объектов — экземпляров классов. Массивы объектов. Списки объектов. Виртуальные методы. Перегрузка методов. Вызовы методов. Статические методы.
10. Этапы разработки приложения. Проектирование объектно-ориентированного приложения. Разработка диаграммы состояний. Диаграмма прецедентов для оконного приложения. Создание интерфейса пользователя. Программирование приложения. Тестирование, отладка приложения. Создание документации.
11. Разработка программ на основе методологии объектно-ориентированного программирования. Создание классов. Объявление методов, свойств. Конструктор и деструктор. Обработчики событий. Реализация инкапсуляции, наследования, полиморфизма. Проектирование класса. Графическое представление класса. Члены класса: поля, конструкторы, методы. Модификаторы доступа.
12. Основные и дополнительные компоненты: кнопка, надпись, поле ввода, флажок, переключатель, поле со списком, полоса прокрутки, список с флажками, главное меню. Контейнер компонентов – объединение элементов. Компоненты панель инструментов и строка состояния. Набор вкладок. Отображаемые и неотображаемые компоненты. Свойства компонентов – элементов управления. Виды свойств. Синтаксис определения свойств. Категории свойств. Назначение свойств и их влияние на результат. Управление объектом через свойства. События элементов управления, их сущность и назначение. Создание компонентов пользователя.
13. Визуальное событийно-управляемое программирование. Типы событий. Создание функций на основе событий. Функции, определённые пользователем: синтаксис, передача аргументов. Вызов событий. События от мыши и клавиатуры. Событие `Click`. Добавление обработчика. Аргументы, передаваемые в параметры обработчика.
14. Форма и модуль как взаимосвязанные компоненты приложения. Модуль формы, структура модуля, заготовка модуля формы. Добавление формы в проект. Свойства и события формы. Обработка событий формы. Передача фокуса между формами. Сворачивание, сокрытие и закрытие окна приложения.
15. Возможности `Visual C#` по организации и использованию диалогов, диалоговых окон. Диалоговые окна открытия и сохранения файлов, установки шрифта, выбора основного и фоновых цветов. Использование масок для списка обрабатываемых файлов. Окна вывода сообщений и ввода данных. Создание пользовательских диалоговых окон.
16. Класс исключений. Стандартные механизмы обработки исключений. Блок `try ... catch ... finally`. Примеры обработки исключений.

17. Тестирование и отладка приложений. Тестирование корректности ввода данных. Автоматизация тестирования. Составление тестовых наборов данных. Программа для генерации тестовых наборов данных.
18. Шаблоны проектирования. Классификация шаблонов проектирования. Шаблоны генерации. Структурные шаблоны. Шаблоны поведения. Шаблон модель, представление, контроллер.

Практика (одна из задач)

1. Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.
2. Создать проект, позволяющий сохранять текст, набранный в текстовом окне в файле данных, очищать текстовое окно и загружать в него сохраненный текст.
3. Создать проект, позволяющий построить треугольник по координатам вершин.
4. Имеется десять флажков. Создать проект, позволяющий сохранять состояние флажков, обнулять и восстанавливать сохраненное состояние.
5. Организовать анимацию процесса движения шарика по окружности.
6. Создать проект, реализующий часы с «будильником».
7. Проект содержит три текстовых окна. Используя буфер обмена, организовать обмен информацией между приложениями.
8. Создать проект, позволяющий добавлять записи в простой список и удалять их.
9. Построить на экране правильный N-угольник. Количество вершин и радиус окружности, на которой они расположены, указываются пользователем.
10. Построить график функции в выбранной пользователем системе координат.
11. Составить программу, генерирующую цвет по трем его составляющим (красный, зеленый, синий).
12. Составить программу, моделирующую работу светофора.
13. Написать программу, моделирующую движение броуновской частицы.
14. На форме имеется десять командных кнопок (пронумерованных от 1 до 10) и графический контейнер. При нажатии на одну из кнопок в контейнере должно появиться соответствующее римское число.
15. Разработать программу в Visual C#, позволяющую заполнять двумерный массив с указанным пользователем количеством строк и столбцов и находить среднее арифметическое в каждой строке и в каждом столбце.
16. Разработать проект, моделирующий арифметический калькулятор. Калькулятор должен выполнять следующие операции: сложение, вычитание, умножение, деление, возведение в квадрат и извлечение квадратного корня как для целых, так и для вещественных чисел.
17. Создать проект, моделирующий работу секундомера. Секундомер должен иметь циферблат с минутной и секундной стрелками, метку и одну командную кнопку.
18. Разработайте проект, имитирующий работу простейшего графического редактора.
19. Разработать компонент «Переключатель», имеющий пять стационарных состояний. Компонент должен иметь свойство для установки состояния и обрабатывать событие, связанное с выбором другого состояния.

20. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран анимацию вращающегося шарика. Координата точки – оси вращения, скорость вращения, радиус вращения и направление вращения задаются.
21. Программа должна выводить в консоль натуральные числа от 1 до N, где N – константа, заданная в программе.
22. Вывести в консоли числа от 1 до N. Задать N через параметры командной строки. Выполнить компиляцию программы с помощью утилиты csc.exe.
23. Создать простое оконное приложение с обработчиком событий от кнопки. Используйте 2 кнопки Button Start и Stop, 1 надпись Label. При щелчке по 1-й кнопке выводится надпись «Start», а при щелчке по второй - надпись «Stop». В отчете нарисуйте эскиз оконного интерфейса приложения и запишите текст обработчиков кнопок.
24. В приложении 5 кнопок, при щелчке по каждой кнопке в окне на компоненте pictureBox рисуется одна фигура: линия, прямоугольник, треугольник, эллипс, дуга.
25. Создать приложение виртуальной реальности в среде Unity 3D с применением скриптов на языке C#.
26. На основе примера приложения создать мобильное приложение виртуальной реальности в среде Unity 3D с применением языка C#.
27. Разработать класс, объекты которого фиксируют время своего существования.
28. Программа загружает выбранный рисунок. Доработать программу, дополнив галереей миниатюр.
29. Разработать компонент, предлагающий интерфейс для отображения одномерного массива и имеющий методы для заполнения, очистки и сортировки.
30. Разработать и протестировать экспортируемый компонент по следующей задаче. Создать проект, позволяющий добавлять записи в простой список и удалять их.
31. Разработать и протестировать экспортируемый компонент по следующей задаче. Разработать класс, объекты которого обладают методами, позволяющими заполнить одномерный массив случайными неповторяющимися числами, вывести его на экран и упорядочить.
32. Модифицировать компонент однострочного поля ввода. Проект inttextbox.
33. Используя объектно-ориентированный подход, написать программу, моделирующую движение броуновских частиц.
34. Разработать компонент «Таймер», выводящий время, прошедшее с момента запуска таймера. Компонент должен иметь метод, возвращающий текущее время, обрабатывать событие, связанное с включением и выключением таймера и метод, обнуляющий таймер.
35. Описать класс, объекты которого связываются с текстовыми окнами, хранят введенную пользователем в окно информацию, обладают методом сортировки символов в текстовых окнах в алфавитном порядке и методом восстанавливающим текст в окне после сортировки.
36. Разработать компонент, представляющий собой окно, в котором можно рисовать окружности. Компонент должен иметь свойство, определяющее цвет «графического пера».
37. Разработать компонент, предлагающий интерфейс для ввода данных и вывода решения квадратного уравнения.
38. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран в любую его точку правильный многоугольник произвольного размера с

произвольным количеством вершин. Разработать экспортируемый компонент. Создать приложение для тестирования компонента.

39. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран график функции. График должен выводиться в прямоугольную область указанного размера. Объект должен иметь свойство, определяющее величину единичного отрезка. Разработать экспортируемый компонент. Создать приложение для тестирования компонента.

40. Разработать класс, объекты которого изображают сектор. Объект должен иметь свойство, определяющее центральный угол, цвета заливки и контура. На основе этого класса, создать класс, объекты которого изображают круговую диаграмму. Разработать экспортируемый компонент. Создать приложение для тестирования компонента.

41. Решить следующую задачу с помощью циклов и указателей на экземпляры класс CheckBox. Имеется десять флажков. Создать проект, позволяющий сохранять состояние флажков, обнулять и восстанавливать сохраненное состояние.

42. В окне приложения разместите в столбик 5 кнопок с надписями button1, ..., button5. С помощью цикла foreach выведите номера кнопок. Цикл выполняется в обработчике 1-й кнопки. Объясните полученный результат.

4.3. Критерии оценивания

Зачет с оценкой выставляется с учетом рейтинга. Если обучающийся набрал недостаточное количество баллов или хочет повысить оценку, то обучающийся сдает зачет с оценкой.

Шкала оценивания для зачета с оценкой:

Уровни освоения индикаторов в достижении компетенций	Содержательное описание уровня	Основные признаки выделения уровня	Академическая оценка	% освоения (рейтинговая оценка)
Повышенный (высокий)	Творческая деятельность	Включает нижестоящий уровень. Умение самостоятельно принимать решение, решать проблему/задачу теоретического или прикладного характера на основе изученных методов, приемов, технологий.	Отлично	90-100
Базовый	Продуктивная деятельность	Включает нижестоящий уровень. Способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику применения	Хорошо	70-89

Удовлетворительный	Репродуктивная деятельность	Изложение в пределах задач курса теоретического и практического материала	Удовлетворительно	50-69
Недостаточный	Отсутствие признаков удовлетворительного уровня		Неудовлетворительно	менее 50

4.4. Методические указания по проведению процедуры промежуточной аттестации

1. Сроки проведения процедуры оценивания: на последнем занятии по предмету. Если обучающийся по результатам рейтинговой системы не набирает нужное количество баллов или желает повысить оценку, то сдает экзамен/зачет согласно требованиям.
2. Сбор, обработка и оценивание результатов промежуточной аттестации проводится преподавателем, ведущим дисциплину.
3. Предъявление результатов оценивания осуществляется: по окончании ответа студента и фиксируется в зачетной книжке и экзаменационной ведомости.
4. При наличии письменных ответов обучающихся, полученных в ходе экзаменационной сессии, материалы хранятся в течение месяца после завершения сессии на кафедрах.
5. Порядок выполнения и защиты курсовой работы регламентирован «Положением о курсовой работе ФГБОУ ВО «Глазовский государственный инженерно-педагогический университет имени В.Г. Короленко».
6. Считать, что положительные результаты промежуточного контроля свидетельствуют об успешном процессе формирования указанных компетенций и индикаторов достижения компетенций (этапов формирования компетенций).

5. Содержание оценочных средств для проверки сформированности компетенций и индикаторов достижения компетенций (поситоговый контроль) и критерии их оценивания

Задания для проверки компетенции и индикаторов достижения компетенции: ПК-1: ИПК-1.1, ИПК-1.2, ИПК-1.3.

Код компетенции	ПК-1
Формулировка компетенции	Способен разрабатывать и отлаживать программный код
Индикатор достижения компетенции	ИПК 1.1 Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования ИПК 1.2 Умеет: применять выбранные языки программирования для написания программного кода ИПК 1.3 Владеет: методами создания программного кода в соответствии с техническим заданием, его отладки и оформления в соответствии с установленными требованиями

Время выполнения заданий: не более 30 минут

Практическое задание. Исправьте ошибки в программном коде, допущенные при решении задачи: присвоить элементам массива псевдослучайные числа из отрезка [A, B].

```
using System;
```

```
namespace test01
{
```

```

class Program
{
    const int SizeArray = 20;
    const int LowRange = 23;
    const int HiRange = 55;
    static void Main(string[] args)
    {
        Random randomizer = new Random();
        int[] intArray = new int[SizeArray];
        Console.WriteLine("Test int array random: ");
        for (int i = 0; i<SizeArray; ++i) {
            intArray[i] = randomizer.NextDouble(LowRange,
HiRange);
        }
        foreach (char item in intArray) {
            Console.Write($"{item} ");
        }
        Console.WriteLine();
    }
}

```

Ключ к практическому заданию: Правильное решение задачи.
using System;

```

namespace test01
{
    class Program
    {
        const int SizeArray = 20;
        const int LowRange = 23;
        const int HiRange = 55;
        static void Main(string[] args)
        {
            Random randomizer = new Random();
            int[] intArray = new int[SizeArray];
            Console.WriteLine("Test int array random: ");
            for (int i = 0; i<SizeArray; ++i) {
                intArray[i] = randomizer.Next(LowRange, HiRange);
            }
            foreach (var item in intArray) {
                Console.Write($"{item} ");
            }
            Console.WriteLine();
        }
    }
}

```

}

Критерии оценивания:

Каждый индикатор достижения компетенции оценивается в 10 баллов:

- Тестовое задание оценивается в 10 баллов (ответ на вопрос теста стоит 0 или 2 балла);
- Задания на соответствие оцениваются в 10 баллов (каждое оценивается 0-5 баллов)
 - 5 баллов – полностью правильно найденные соответствия;
 - 4 балла – три правильных соответствия;
 - 3 балла – два правильных соответствия;
 - 2 балла – одно правильно соответствие;
 - 1 балл – отсутствие правильных соответствий;
 - 0 баллов – не приступал к выполнению задания;
- Каждое практическое задание оценивается в 10 баллов:
 - 10 баллов - студент правильно выполнил предложенные задания на основе изученной теории, методов, приемов, технологий;
 - 8 баллов - студент способен применять полученные теоретические знания в практической деятельности, решать типичные задачи на основе воспроизведения стандартных алгоритмов, при выполнении заданий допускает незначительные ошибки;
 - 6 баллов - при выполнении задания допущены грубые ошибки;
 - 0 баллов - студент не выполнил задание.

Оценка зависит от процента выполнения всех заданий.

Шкала оценивания сформированности компетенции (ий) и индикатора (ов) достижения компетенции (ий)

Уровни освоения индикатора (ов) достижений компетенций	Основные признаки выделения уровня	Академическая оценка	% выполнения всех заданий
Повышенный (высокий)	Включает нижестоящий уровень. Умение самостоятельно принимать решение, решать проблему/задачу теоретического или прикладного характера на основе изученных методов, приемов, технологий.	Отлично	90-100
Базовый	Включает нижестоящий уровень. Способность собирать, систематизировать, анализировать и грамотно использовать информацию из самостоятельно найденных теоретических источников и иллюстрировать ими теоретические положения или обосновывать практику применения	Хорошо	70-89
Удовлетворительный	Изложение в пределах задач курса теоретического и практического контролируемого материала	Удовлетворительно	50-69
Недостаточный	Отсутствие признаков удовлетворительного уровня	Неудовлетворительно	менее 50