


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ИНКЛЮЗИВНОГО ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО  
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Факультет Прикладная математика и информатика  
Кафедра Информационных технологий и прикладной математики

«Утверждаю»  
Зав. кафедрой   
«26» августа 2019 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ДИСЦИПЛИНЕ**

**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

образовательная программа направления подготовки  
09.03.03 Прикладная информатика  
Блок Б1.В.14 «Дисциплины (модули)», часть, формируемая участниками  
образовательных отношений

**Профиль подготовки**

Прикладная информатика в биоинформационных технологиях

Квалификация (степень) выпускника  
Бакалавр

Форма обучения очная

Курс 4 семестр 7,8

Москва  
2019

Составитель / составители: МГГЭУ, доцент кафедры информационных технологий и прикладной математики

  
\_\_\_\_\_

подпись

место работы, занимаемая должность

Белоглазов А.А.  
Ф.И.О.

«22» августа 2019 г.  
Дата

Рецензент: МГГЭУ, профессор кафедры информационных технологий и прикладной математики

  
\_\_\_\_\_

подпись

место работы, занимаемая должность

Истомина Т.В.  
Ф.И.О.

«23» августа 2019 г.  
Дата

Согласовано:

*Представитель работодателя или объединения работодателей*

Генеральный директор, АО «Микропроцессорные системы», к.т.н.

(должность, место работы)

  
\_\_\_\_\_

подпись

Демидов Л.Н.  
Ф.И.О.

«26» августа 2019 г.  
Дата

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры Информационных технологий и прикладной математики (протокол № 1 от «26» августа 2019 г.)

/Зав. кафедрой ИТиПМ/  Петрунина Е.В. «26» августа 2019 г.

подпись

Ф.И.О.

Дата

Дополнения и изменения, внесенные в фонд оценочных средств, утверждены на заседании кафедры \_\_\_\_\_,

протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой \_\_\_\_\_ / Ф.И.О/

Дополнения и изменения, внесенные в фонд оценочных средств, утверждены на заседании кафедры \_\_\_\_\_,

протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой \_\_\_\_\_ / Ф.И.О/

Дополнения и изменения, внесенные в фонд оценочных средств, утверждены на заседании кафедры \_\_\_\_\_,

протокол № \_\_\_\_ от « \_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Заведующий кафедрой \_\_\_\_\_ / Ф.И.О/

## Содержание

1. Паспорт фонда оценочных средств.....
2. Перечень оценочных средств.....
3. Описание показателей и критериев оценивания компетенций.....
4. Методические материалы, определяющие процедуры оценивания результатов обучения, характеризующих этапы формирования компетенций.....
5. Материалы для проведения текущего контроля и промежуточной аттестации.....
- ...

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Объектно-ориентированное программирование»

Оценочные средства составляются в соответствии с рабочей программой дисциплины и представляют собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.), предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные средства используются при проведении текущего контроля успеваемости и промежуточной аттестации.

Таблица 1 - Перечень компетенций, формируемых в процессе освоения дисциплины

Код компетенции	Наименование результата обучения
ПК-2	<p>Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение</p> <p>ПК-2.1. Знает программные шаблоны; метрики и риски тестирования; базовые понятия качества программного продукта и качества процесса разработки программного обеспечения; основные концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования); функциональные характеристики применения программного обеспечения.</p> <p>ПК-2.2. Умеет реализовывать программные продукты на языках программирования высокого уровня; описывать архитектуру программного средства включая выделение: функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов; применять соответствующие программные или аппаратные архитектурные решения; использовать модели данных; анализировать и оценивать архитектуру на предмет атрибутов качества.</p> <p>ПК-2.3. Владеет навыками планирования процесса разработки программного продукта; навыками задания функциональных рамок подсистем; навыками определения наиболее значимых критериев качества программного продукта.</p>
ПК-3	<p>Способен проектировать ИС в соответствии с профилем подготовки по видам обеспечения</p> <p>ПК-3.1. Знает методы разработки, анализа и проектирования ПО; функциональное и техническое проектирование; паттерны проектирования; номенклатуру инструментальных средств, поддерживающих создание программного обеспечения; техники распределения задач на разработку между исполнителями.</p> <p>ПК-3.2. Умеет работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем; использовать нотации для построения функциональной и процессной моделей исследуемой предметной области; проектировать компоненты программных средств.</p> <p>ПК-3.3. Владеет навыками определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта; навыками проектирования прикладных программных продуктов, в том числе клиент-серверных приложений.</p>
ПК-8	Способен программировать приложения и создавать программные прототипы решения прикладных задач

	<p>ПК-8.1. Знает основные языки программирования приложений; теоретические и методические основы технологии программирования, анализа и применения алгоритмических и программных решений; методы и приёмы разработки программных прототипов решения прикладных задач.</p> <p>ПК-8.2. Умеет реализовывать программные продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p> <p>ПК-8.3. Владеет навыками разработки приложений и программных прототипов.</p>
--	---

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл.2).

Таблица 2 - Формирование компетенций в процессе изучения дисциплины:

Код компетенции	Уровень освоения компетенций	Индикаторы достижения компетенций	Вид учебных занятий <sup>1</sup> , работы, формы и методы обучения, способствующие формированию и развитию компетенций <sup>2</sup>	Контролируемые разделы и темы дисциплины <sup>3</sup>	Оценочные средства, используемые для оценки уровня сформированности компетенции <sup>4</sup>
ПК-2		<i>Знает</i>			
	Недостаточный уровень	ПК-2.1. Не знает программных шаблонов; основных концепций и атрибутов качества программного обеспечения (надежности, безопасности, удобства использования); функциональных характеристик применения программного обеспечения.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET.	Текущий контроль – устный опрос.

<sup>1</sup> Лекционные занятия, практические занятия, лабораторные занятия, самостоятельная работа...

<sup>2</sup> Необходимо указать активные и интерактивные методы обучения (например, интерактивная лекция, работа в малых группах, методы мозгового штурма и т.д.), способствующие развитию у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств.

<sup>3</sup> Наименование темы (раздела) берется из рабочей программы дисциплины.

<sup>4</sup> Оценочное средство должно выбираться с учетом запланированных результатов освоения дисциплины, например:

«Знать» – собеседование, коллоквиум, тест...

«Уметь», «Владеть» – индивидуальный или групповой проект, кейс-задача, деловая (ролевая)

игра, портфолио...

				<p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
Базовый уровень	<p>ПК-2.1. Студент усвоил основное содержание материала дисциплины, но имеет пробелы в применении на практике. Имеет несистематизированные знания о программных шаблонах; основных концепциях и атрибутах качества программного обеспечения (надежности, безопасности, удобства использования); функциональных характеристик применения программного обеспечения.</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.</p> <p>Тема 6. Проектирование ПО.</p> <p>Тема 7. Методы разработки ПО.</p> <p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	<p>Текущий контроль – устный опрос.</p>	

	Средний уровень	ПК-2.1 Студент способен выделять главные положения в изученном материале. Знает программные шаблоны; основные концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования); функциональные характеристики применения программного обеспечения.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос.
	Высокий уровень	ПК-2.1. Студент знает, понимает, выделяет главные положения в изученном материале и способен дать краткую характеристику основным идеям проработанного материала дисциплины. Знает программные шаблоны; основные концепции и атрибуты качества	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных	Текущий контроль – устный опрос, тестирование.



		программного обеспечения (надежности, безопасности, удобства использования); функциональные характеристики применения программного обеспечения.		запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	
		<i>Умеет</i>			
Базовый уровень	ПК-2.2. Студент затрудняется описывать архитектуру программного средства; использовать модели данных; анализировать и оценивать архитектуру на предмет атрибутов качества.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное	Текущий контроль – устный опрос.	

				<p>программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
Средний уровень	<p>ПК-2.2. Студент умеет описывать архитектуру программного средства; использовать модели данных; анализировать и оценивать архитектуру на предмет атрибутов качеств, но допускает незначительные ошибки</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	<p>Текущий контроль – устный опрос, тестирование.</p>	

	Высокий уровень	ПК-2.2. Студент умеет самостоятельно описывать архитектуру программного средства; использовать модели данных; анализировать и оценивать архитектуру на предмет атрибутов качества.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.</p> <p>Тема 6. Проектирование ПО.</p> <p>Тема 7. Методы разработки ПО.</p> <p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	Текущий контроль – устный опрос.
		<i>Владеет</i>			
	Базовый уровень	ПК-2.3. Студент владеет основными навыками определения наиболее значимых критериев качества программного продукта.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и</p>	Текущий контроль – устный опрос.

				<p>события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.          Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
Средний уровень	<p>ПК-2.3. Студент владеет знаниями всего изученного материала, владеет навыками определения наиболее значимых критериев качества программного продукта, но допускает незначительные ошибки.</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#          Тема 2. Объекты и типы          Тема 3. Наследование          Тема 4. Обобщения, массивы и кортежи. Коллекции.          Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное</p>	<p>Текущий контроль – устный опрос.</p>	

				программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	
	Высокий уровень	ПК-2.3. Студент владеет знаниями всего изученного материала, владеет навыками определения наиболее значимых критериев качества программного продукта.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос.
ПК-3		Знает			

	Недостаточный уровень	ПК-3.1. Не способен проектировать ИС в соответствии с профилем подготовки по видам обеспечения. Не знает основных принципов, методов и средств разработки ПО на языках высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос.
	Базовый уровень	ПК-3.1. Студент усвоил основное содержание материала дисциплины, но имеет несистематизированные знания о принципах, методах и средствах разработки ПО на языках высокого уровня (C#, C++);	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных	Текущий контроль – устный опрос, тестирование.

				<p>запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.          Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
	Средний уровень	ПК-3.1. Студент знает основные принципы, методы и средства разработки ПО на языках высокого уровня (C#, C++)	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#          Тема 2. Объекты и типы          Тема 3. Наследование          Тема 4. Обобщения, массивы и кортежи. Коллекции.          Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.</p>	Текущий контроль – устный опрос, тестирование.

				Тема 12. Программирование с использованием пакета Math.Net Numerics.	
Высокий уровень	ПК-3.1. Студент показывает глубокое знание и понимание основных принципов, методов и средств разработки ПО на языках высокого уровня (C#, C++)	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации		Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – Решение разноуровневых задач (заданий), устный опрос.
		<i>Умеет</i>			



	<p>Базовый уровень</p>	<p>ПК-3.2. Студент испытывает затруднения при применении программных продуктов для решения прикладных задач на языках высокого уровня (C#, C++)</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#  Тема 2. Объекты и типы  Тема 3. Наследование  Тема 4. Обобщения, массивы и кортежи. Коллекции.  Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.  Тема 6. Проектирование ПО.  Тема 7. Методы разработки ПО.  Тема 8. Визуальное программирование графических интерфейсов пользователя.  Тема 9. Доступ к данным в .NET.  Тема 10. Основы разработки Web-приложений в .NET.  Тема 11. Асинхронное программирование.  Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	<p>Текущий контроль – устный опрос.</p>
	<p>Средний уровень</p>	<p>ПК-3.2. Студент умеет реализовывать программные продукты для решения прикладных задач и разрабатывать ПО на языках высокого уровня (C#, C++), но допускает ошибки.</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#  Тема 2. Объекты и типы  Тема 3. Наследование  Тема 4. Обобщения, массивы и кортежи. Коллекции.  Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных</p>	<p>Текущий контроль – устный опрос.</p>

				<p>запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.          Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
	Высокий уровень	ПК-3.2. Студент умеет самостоятельно реализовывать программные продукты для решения прикладных задач и разрабатывать ПО на языках высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#          Тема 2. Объекты и типы          Тема 3. Наследование          Тема 4. Обобщения, массивы и кортежи. Коллекции.          Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.</p>	Текущий контроль – устный опрос.

				Тема 12. Программирование с использованием пакета Math.Net Numerics.	
		<i>Владеет</i>			
Базовый уровень	ПК-3.3. Студент владеет базовыми приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта Интегрированной Среды Разработки (ИСР) MS VisualStudio	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос.	

	Средний уровень	ПК-3.3. Студент владеет приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта Интегрированной Среды Разработки (ИСР) MS VisualStudio; навыками задания функциональных рамок подсистем	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы С# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос, тестирование.
	Высокий уровень	ПК-3.3. Студент владеет концептуально-понятийным аппаратом Владеет приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта Интегрированной Среды Разработки (ИСР) MS VisualStudio; навыками задания функциональных рамок подсистем	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы С# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных	Текущий контроль – Решение разноуровневых задач (заданий), устный опрос.

		рамок администрирования подсистем		запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	
<i>ПК-8</i>		<i>Знает</i>			
	Недостаточный уровень	ПК-8. Способен программировать приложения и создавать программные прототипы решения прикладных задач. Не знает методы и приёмы разработки программных прототипов решения прикладных задач с использованием языков высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное	Текущий контроль – устный опрос.

				программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	
Базовый уровень	ПК-8.1. Студент показывает поверхностное знание и понимание о методах и приёмах разработки программных прототипов решения прикладных задач с использованием языков высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	Текущий контроль – устный опрос.	
Средний уровень	ПК-8.1. Студент знает методы и приёмы разработки программных прототипов решения прикладных задач с использованием языков высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и	Текущий контроль – устный опрос.	

				<p>кортежи. Коллекции.          Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.          Тема 10. Основы разработки Web-приложений в .NET.          Тема 11. Асинхронное программирование.          Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
Высокий уровень	<p>ПК-8.1. Студент показывает глубокое знание и понимание о методах и приёмах разработки программных прототипов решения прикладных задач с использованием языков высокого уровня (C#, C++).</p>	<p>Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#          Тема 2. Объекты и типы          Тема 3. Наследование          Тема 4. Обобщения, массивы и кортежи. Коллекции.          Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.          Тема 6. Проектирование ПО.          Тема 7. Методы разработки ПО.          Тема 8. Визуальное программирование графических интерфейсов пользователя.          Тема 9. Доступ к данным в .NET.</p>	<p>Текущий контроль – устный опрос.</p>	

				<p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
		<i>Умеет</i>			
Базовый уровень	ПК-8.2. испытывает затруднения при описании архитектуры приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов с использованием языков высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.</p> <p>Тема 6. Проектирование ПО.</p> <p>Тема 7. Методы разработки ПО.</p> <p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	Текущий контроль – устный опрос.	
Средний уровень	ПК-8.2. Студент умеет описывать архитектуру приложений включая	Лекционные и практические занятия,	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура</p>	Текущий контроль – устный опрос.	



		выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов с использованием языков высокого уровня (C#, C++).	самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	.NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.	
Высокий уровень	ПК-8.2. Студент умеет самостоятельно описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов с использованием языков высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C# Тема 2. Объекты и типы Тема 3. Наследование Тема 4. Обобщения, массивы и кортежи. Коллекции. Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО.	Текущий контроль – устный опрос.	

				<p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
		<i>Владеет</i>			
Базовый уровень	ПК-8.3. Студент на базовом уровне владеет навыками разработки приложений и в ПО на языках высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.</p> <p>Тема 6. Проектирование ПО.</p> <p>Тема 7. Методы разработки ПО.</p> <p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net</p>	Текущий контроль – устный опрос.	

				Numerics.	
Средний уровень	ПК-8.3. Студент на среднем уровне владеет навыками разработки приложений и в ПО на языках высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации		<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.</p> <p>Тема 6. Проектирование ПО.</p> <p>Тема 7. Методы разработки ПО.</p> <p>Тема 8. Визуальное программирование графических интерфейсов пользователя.</p> <p>Тема 9. Доступ к данным в .NET.</p> <p>Тема 10. Основы разработки Web-приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	Текущий контроль – устный опрос, тестирование.
Высокий уровень	ПК-8.3. Студент на высоком уровне владеет навыками разработки приложений и в ПО на языках высокого уровня (C#, C++).	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации		<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p> <p>Тема 2. Объекты и типы</p> <p>Тема 3. Наследование</p> <p>Тема 4. Обобщения, массивы и кортежи. Коллекции.</p> <p>Тема 5. Операции и приведения. Делегаты, лямбда-выражения и</p>	Текущий контроль – Решение разноуровневых задач (заданий), устный опрос.

				<p>события. Язык интегрированных запросов. Ошибки и исключения. Тема 6. Проектирование ПО. Тема 7. Методы разработки ПО. Тема 8. Визуальное программирование графических интерфейсов пользователя. Тема 9. Доступ к данным в .NET. Тема 10. Основы разработки Web-приложений в .NET. Тема 11. Асинхронное программирование. Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
--	--	--	--	--	--

## 2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ<sup>5</sup>

Таблица 3

№	Наименование оценочного средства	Характеристика оценочного средства	Представление оценочного средства в ФОС
1	Устный опрос	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися.	Вопросы по темам/разделам дисциплины
2	Решение разноуровневых задач (заданий)	Различают задачи и задания: а) репродуктивного уровня, позволяющие оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, узнавание объектов изучения в рамках определенного раздела дисциплины; б) реконструктивного уровня, позволяющие оценивать и диагностировать умения синтезировать, анализировать, обобщать фактический и теоретический материал с формулированием конкретных выводов, установлением причинно-следственных связей; в) творческого уровня, позволяющие оценивать и диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения.	Комплект разноуровневых задач (заданий)
3	Тестирование	Средство, позволяющее оценить уровень знаний обучающегося путем выбора им одного из нескольких вариантов ответов на поставленный вопрос. Возможно использование тестовых вопросов, предусматривающих ввод обучающимся короткого и однозначного ответа на поставленный вопрос.	Тестовые задания
4	Зачет		Вопросы к зачету
5	Зачет с оценкой		Вопросы к зачету с оценкой

<sup>5</sup> Указываются оценочные средства, применяемые в ходе реализации рабочей программы данной дисциплины.

### 3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценивание результатов обучения по дисциплине «Объектно-ориентированное программирование» осуществляется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся.

Предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины) и промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом).

Показатели и критерии оценивания компетенций, формируемых в процессе освоения данной дисциплины, описаны в табл. 4.

Таблица 4.

Код компетенции	Уровень освоения компетенции	Индикаторы достижения компетенции	Критерии оценивания результатов обучения
ПК-2, ПК-3, ПК-8		Знает	
	Недостаточный уровень Оценка «незачтено», «неудовлетворительно»	ПК-2.1. ПК-3.1. ПК-8.1.	<i>Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины</i>
	Базовый уровень Оценка, «зачтено», «удовлетворительно»	ПК-2.1. ПК-3.1. ПК-8.1.	<i>Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении</i>
	Средний уровень Оценка «зачтено», «хорошо»	ПК-2.1. ПК-3.1. ПК-8.1.	<i>Знает основную часть материала курса, способен применить изученный материал на практике, испытывает незначительные затруднения в решении задач</i>
	Высокий уровень Оценка «зачтено», «отлично»	ПК-2.1. ПК-3.1. ПК-8.1.	<i>Показывает глубокое знание и понимание материала, способен применить изученный материал на практике</i>
		Умеет	
	Базовый уровень	ПК-2.2. ПК-3.2. ПК-8.2.	<i>Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач</i>
	Средний уровень	ПК-2.2. ПК-3.2. ПК-8.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач</i>
	Высокий уровень	ПК-2.2. ПК-3.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, показывает глубокое знание и понимание материала, способен решить задачу</i>

		<i>ПК-8.2.</i>	<i>при изменении формулировки</i>
		Владеет	
Базовый уровень		<i>ПК-2.3.</i> <i>ПК-3.3.</i> <i>ПК-8.3.</i>	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания основных разделов дисциплины.</i>
Средний уровень		<i>ПК-2.3.</i> <i>ПК-3.3.</i> <i>ПК-8.3.</i>	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, способен самостоятельно выделять главные положения в изученном материале. Испытывает незначительные затруднения в решении задач.</i>
Высокий уровень		<i>ПК-2.3.</i> <i>ПК-3.3.</i> <i>ПК-8.3.</i>	<i>Свободно владеет навыками теоретического и экспериментального исследования, показывает глубокое знание и понимание изученного материала</i>

## **4. Методические материалы, определяющие процедуры оценивания результатов обучения**

### **Задания в форме устного опроса:**

Устный опрос используется для текущего контроля успеваемости обучающихся по дисциплине в качестве проверки результатов освоения терминологии. Каждому студенту выдается свой собственный, узко сформулированный вопрос. Ответ должен быть четким и кратким, содержащим все основные характеристики описываемого понятия, института, категории.

### **Задания в форме практических работ**

Практическая работа представляет собой контрольное мероприятие по учебному материалу каждой темы (раздела) дисциплины, состоящее в индивидуальном выполнении обучающимся практических заданий для оценки полученных знаний, умений и владений компетенциями, формируемыми по данной дисциплине.

Выполнение практических работ является средством текущего контроля успеваемости обучающихся по дисциплине и может включать в себя следующие типы заданий: задания типового вида и задания творческого характера, по результатам выполнения практических заданий обучающиеся оформляют отчеты, содержащие анализ полученных результатов и выводы.

### **Задания в форме тестирования**

Тест представляет собой контрольное мероприятие по учебному материалу каждой темы (раздела) дисциплины, состоящее в выполнении обучающимся системы стандартизированных заданий, которая позволяет автоматизировать процедуру измерения уровня знаний и умений обучающегося.

Тестирование является средством текущего контроля успеваемости обучающихся по дисциплине и может включать в себя следующие типы заданий: задание с единственным выбором ответа из предложенных вариантов, задание на определение верных и неверных суждений; задание с множественным выбором ответов.

В каждом задании необходимо выбрать все правильные ответы.

## **5. Материалы для проведения текущего контроля и промежуточной аттестации**

### **Задания в форме устного опроса**

1 Новейшие направления в области создания технологий программирования. Законы эволюции программного обеспечения.

2 Программирование в средах современных информационных систем: создание модульных программ, элементы теории модульного программирования объектно-ориентированные проектирование и программирование

3 Среда Net. Framework. Общеязыковая исполняющая среда (CLR). Стандарты C++. Технологии отладки

4 Объектно-ориентированный подход к проектированию и разработке программ. Сущность объектно-ориентированного подхода

5 Инкапсуляция. Понятие класса. Управление доступом к элементам класса. Конструкторы и деструкторы. Переменные объектного типа.



- 6 Друзья класса.
  - 7 Наследование. Базовый класс. Управление доступом при наследовании.
  - 8 Простое наследование. Виртуальные функции.
  - 9 Чистые виртуальные функции. Абстрактные классы. Непрямые базовые классы. Виртуальные деструкторы. Виртуальные базовые классы
  - 10 Полиморфизм.
  - 11 Методы общего полиморфизма: перегрузка операций. Реализация перегруженной операции. Перегрузка операции присваивания, инкремента, декремента, бинарных арифметических операций.
  - 12 Методы общего полиморфизма: перегрузка операций. Перегрузка операций ввода-вывода в потоки, индексации массивов, операций выделения и освобождения памяти.
  - 13 Методы общего полиморфизма: Преобразования типов, определяемые классом.
  - 14 Методы общего полиморфизма: перегрузка функций. Чистый полиморфизм.
  - 15 Параметрический полиморфизм. Шаблоны функций.
  - 16 Параметрический полиморфизм. Шаблоны классов
  - 17 Классы в C#. Управление доступом. Конструкторы. Деструкторы. Наследование.
  - 18 Свойства классов в C#. Скалярные и индексированные свойства.
  - 19 Перегрузка операций в C#. Преобразование типов в C#.
  - 20 Делегаты. события. Интерфейсы
  - 21 Концепции программирования для Windows. Структура Windows программ. Использование Windows Forms для создания приложений с графическим интерфейсом пользователя.
  - 22 Создание элемента управления Windows Forms
- Контролируемые компетенции: ПК-2, ПК-3, ПК-8**  
*Оценка компетенций осуществляется в соответствии с таблицей 4.*

## **Задания в форме аудиторных контрольных и самостоятельных работ**

- 1 Определить класс «Многочлен», описывающий алгебраический многочлен от одной переменной, задаваемой степенью многочлена и массивом коэффициентов. Реализовать в классе методы, позволяющие вычислять значения многочлена для заданного аргумента, выполнять операции сложения, вычитания и умножения многочленов, взятия производной произвольного порядка с получением нового объекта-многочлена. В программе продемонстрировать использование объектов класса «Многочлен»
- 2 Определить класс «Комплексное число». Реализовать в виде класса методы для выполнения следующих операций над комплексными числами:
  - сложение;
  - вычитание;
  - умножение;
  - деление;
  - модуль комплексного числа;
  - возведение комплексного числа в степень  $n$  ( $n$  — натуральное).
 В программе продемонстрировать использование объектов класса «Комплексное число»
- 3 Реализовать класс —Число в  $P$ -ичной системе счисления|. Основание системы счисления должно быть компонентным данным класса. Реализовать в виде методов следующие операции над натуральными числами в  $P$ -ичной системе счисления ( $2 \leq P \leq 9$ ):
  - сложение;
  - вычитание;
  - умножение;
  - деление;
  - перевод из десятичной системы счисления в  $P$ -ичную;
  - перевод из  $P$ -ичной системы счисления в десятичную;

проверка правильности записи числа в  $P$ -ичной системе счисления;

В программе продемонстрировать использование объектов класса «Число в  $P$ -ичной системе счисления»

4 Реализовать в виде класса «Обыкновенная дробь», описывающий обыкновенные дроби вида  $P/Q$  ( $P$  — целое,  $Q$  — натуральное). Для класса реализовать методы реализующие операции с дробями:

a. сложение;

b. вычитание;

c. умножение;

d. деление;

e. сокращение дроби;

f. возведение дроби в степень  $n$  ( $n$  — натуральное);

g. функции, реализующие операции отношения (равно, не равно, больше или равно, меньше или равно, больше, меньше).

В программе продемонстрировать использование объектов класса «Обыкновенная дробь»

В программе продемонстрировать использование объектов класса «Квадратная матрица».

## **Вариант 2**

1 Разработать класс «Римское число» для работы с числами, записанными в римской системе счисления. Реализовать возможность ввода и вывода чисел в римской систем счисления, а также методы, реализующие операции:

a. перевод натурального числа из десятичной системы счисления в римскую;

b. перевод числа из римской системы счисления в десятичную;

c. сложение;

d. вычитание;

e. умножение;

f. целочисленное деление и нахождение остатка от деления;

g. операции отношения (описать в виде логических функций)

h. проверка правильности записи римского числа.

Операции сложения, вычитания, деления и умножения римских чисел реализовать как перегрузки соответствующих операций.

В программе продемонстрировать использование объектов класса «Римское число».

2 Разработать класс «Стек», реализующий функционирования стека LIFO. В классе реализовать возможность добавление и удаление элемента, проверки, не пуст и не переполнен ли стек, получения количества элементов в стеке. Функции добавления и удаления информации в/из стека реализовать как перегрузку операций сложения и вычитания.

С помощью класса «Стек» проверить на сбалансированность по круглым и угловым скобкам текст в файле.

3 Определить класс «Множество», реализующий функционирование множества символов русского языка. В классе реализовать возможности: добавление и удаления символа в/из множества, сравнения с другим множеством, объединение множеств, пересечения множеств, вычитания множеств.

Функции объединения и вычитания множеств реализовать как перегрузки операций сложения и вычитания.

С использованием класса «Множество» найти в файле пару предложений, которые содержат один и тот же набор символов, а также пару предложений, в которых не совпадают ни один символ.

4 Определить класс «Квадратная матрица». В классе реализовать методы, позволяющие:

вычислять определитель матрицы, транспонировать матрицу, получать обратную матрицу, проверять, не является ли матрица вырожденной, перемножать на другую матрицу.

**Контролируемые компетенции: ПК-2, ПК-3, ПК-8**

*Оценка компетенций осуществляется в соответствии с таблицей 4.*

## Задания в тестовой форме

### Вариант 1

1 С помощью чего реализуется принцип полиморфизма в C ++?

- 1) - наличия множественного наследования.
- 2) - наличия виртуальных методов.
- 3) - использование виртуального наследования.
- 4) - наличия абстрактных классов.

2 В программе описано класс и объект `class A {public: int a, b, c; }; A * obj;` Как обратиться к атрибуту `c`?

- 1) - `obj.c`
- 2) - `obj-> c`
- 3) - `obj A -> -> c`
- 4) - `obj-> A.c`

3 Какая из перечисленных функций не может быть конструктором?

- 1) `void String ()`
- 2) `String ();`
- 3) `String (String & s)`
- 4) `String (const int a)`

4 Отметьте правильное утверждение для абстрактного класса для языка C ++.

- 1) - Класс, у которого все методы чисто виртуальные, называется абстрактным.
- 2) - Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
- 3) - Невозможно создать объект абстрактного класса.
- 4) - В абстрактном классе не описываются методы вообще.

5 Если в программе на языке C ++ в производном классе переопределена операция `new` то ...

- 1) - все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоны видимости, в которой она переопределена.
- 2) - производные от этого класса могут использовать глобальную операцию применив операцию базовый\_класс :: `new`.
- 3) - операцию `new` нельзя переопределить.
- 4) - в любом случае эта операция будет доступна только в пределах класса-потомка.

6 Какой из перечисленных методов может быть конструктором для класса `String` в языке C ++?

- 1) `String * String ();`
- 2) `void String ();`
- 3) - `String (String & s);`
- 4) - `const String (int a);`

7 Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам?

- 1) - Шаблонная.
- 2) - Полиморфная.
- 3) - Дружеская.
- 4) - Статическая.

8 Вызовет данный код ошибку компиляции? `class Rectangle public: int a, b; int sum (); int square (); ~Rect ();`};

- 1) - Ошибки нет, все записано верно.
- 2) - Ошибка является: имя деструктора должно совпадать с именем класса.
- 3) - Ошибка является: имя деструктора не может начинаться с маленькой буквы.

4) - Ошибка являются: никакой идентификатор в C ++ не может начинаться со знака «~».

9 Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.

- 1) virtual void SomeFunction (int x);
- 2) void SomeFunction (int x) virtual;
- 3) virtual SomeFunction (int x);
- 4) virtual void SomeFunction (int \* x);

10 Укажите правильное использование оператора friend.

- 1) class A {int\_friend CountPass (); private: short i};
- 2) class A {public: friend int H :: CountPass (); private: short i};
- 3) class A {public: int A1 :: CountPass (); friend: short i};
- 4) class A {public: friend int H :: q; short i};

1.	2
2.	2
3.	1
4.	2
5.	3
6.	3
7.	3
8.	1
9.	1
10.	2

## Вариант 2

1 Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

- 1) - Наследование.
- 2) - Сочетание.
- 3) - Инициализация.
- 4) - Инкапсуляция.

2 В программе описано абстрактный класс A и производный от этого класса класс A1. Какой из записей заведомо неверный?

- 1) A \* a = new A;
- 2) A1 a1;
- 3) A1 a1; A & a = a1;
- 4) A1 a1; A1 a2;

3 Выберите верное утверждение о деструктор класса в C ++.

- 1) - Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
- 2) - Деструктор принимает в качестве параметра указатель this.
- 3) - Деструктор не содержит параметров.
- 4) - Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

4 Укажите правильный вариант доступа к членам объектов (язык C ++), описанных следующим образом: class my {char s;public: double Z; int f (int c, int d) {return c + d;}; } T1, T2;

- 1) T1.Z = 23.1;
- 2) T2-> f (2,1);
- 3) T1.s = '#';
- 4) -. my.T2-> s = 'L';

5 Какой из вариантов записи абстрактного класса в C ++ является правильным?

- 1) abstract class A { virtual int f () = 0;};
- 2) class A { virtual int f () = 0;};
- 3) class A { virtual int f () = 0; } abstract;
- 4) class A { virtual int f ();};

6 Какое из свойств скрывает внутренние данные объекта?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

7 Какое из свойств использует виртуальные или перегружаемые элементы?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

8 Какое из свойств строит иерархию объектов?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

9 Какое из свойств предназначено для улучшения интерфейса работы с объектами?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

10.Какая из операций используется для обозначения указателя?

- 1) DIV
- 2) \*
- 3) {\$
- 4) ^

1.	4
2.	1
3.	3
4.	1
5.	2
6.	1
7.	2
8.	3
9.	4
10.	2

Контролируемые компетенции: ПК-2, ПК-3, ПК-8

Оценка компетенций осуществляется в соответствии с таблицей 4.

### Вопросы к зачету

1. Назовите основные принципы ООП.
2. Назовите составные части модели известной как трёхуровневое приложение.
3. Назначение .Net Framework?
4. Что такое managed code?
5. Что такое assembly?
6. Поддерживает ли C# множественное наследование?
7. Назовите класс .NET, от которого наследуются все классы.
8. Назовите различия между структурами и классами.

9. Какой синтаксис используется для указания класса родителя в C#?
10. Что такое абстрактный класс?
11. Что такое интерфейс класса?
12. Что означает ключевое слово «virtual» для метода или свойства?
13. Что такое делегат?
14. Как отсортировать элементы массива в убывающем порядке?
15. Чем перекрытый метод отличается от перегруженного метода?
16. Что такое JIT-компиляция?
17. Что такое «сборщик мусора»?
18. Назовите методы явного преобразования типов.
19. Что такое обобщенные классы?
20. Что такое одномерный массив, многомерный массив, зубчатый массив?
21. Назовите основные типы коллекций.
22. Как перехватить исключение в C#?
23. Опишите преимущества использования LINQ.

**Контролируемые компетенции: ПК-2, ПК-3, ПК-8**

*Оценка компетенций осуществляется в соответствии с таблицей 4.*

### **Вопросы к зачету с оценкой**

1. Эволюция методологий программирования. Парадигмы программирования.
  2. Основные принципы объектного подхода.
  3. Общеязыковая исполняющая среда (CLR). Промежуточный язык (IL).
- Сборки. Классы .NET.
4. Классы и структуры. Статические классы. Класс Object (C#).
  5. Концепция наследования. Типы наследования. Наследование реализации.
  6. Массивы в C#. Класс Array. Использование массивов в качестве параметров.
  7. Операции. Безопасность типов. Проверка объектов на предмет равенства.
  8. Интерфейсы и типы коллекций. Списки. Словари. Множества. Битовые массивы
  9. Проектирование с использованием объектов.
  10. Проектирование с использованием библиотек и шаблонов.
  11. LINQ. Стандартные операции запросов.
  12. Классы исключений. Перехват исключений. Классы исключений.
  13. Жизненный цикл объекта. Конструкторы и деструкторы
  14. Объектно-ориентированное визуальное программирование интерфейсов пользователя в .NET (Windows Forms: окна, элементы управления и компоненты).
  15. Инфраструктура Windows Presentation Foundation (WPF) как альтернатива Windows Forms.
  16. Расширяемый язык разметки для приложений (XAML).
  17. Создание документов с помощью WPF.
  18. Объектная модель ADO.NET. Подключение к БД.
  19. Объектная модель ADO.NET. Выполнение команд.
  20. Привязка данных к элементам управления Windows Forms.
  21. Платформа .NET Framework для Web-приложений. ASP.NET Web Forms: модель страницы ASPX, мастер-страницы, навигация.
  22. Привязка данных к элементам управления ASP.NET Web Forms.
  23. Платформа .NET Framework для Web-приложений. ASP.NET Web Forms и ASP.NET MVC.
  24. Асинхронное программирование.
  25. Интерфейсы пользователя.
  26. Перегрузка и переопределение методов класса.

27. Методы для установления и получения значений приватных свойств класса.
28. Обобщенные классы.
29. Абстрактные классы.
30. Объектно-ориентированная технология доступа к данным ADO.NET Entity

Framework.

Контролируемые компетенции: ПК-2, ПК-3, ПК-8

*Оценка компетенций осуществляется в соответствии с таблицей 4.*