

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ИНКЛЮЗИВНОГО ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО  
ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»**

---

Факультет Прикладной математики и информатики  
Кафедра Информационных технологий и прикладной математики

«Утверждаю»  
Зав. кафедрой   
«26» августа 2018 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ДИСЦИПЛИНЕ**

«Объектно-ориентированное программирование»

образовательная программа направления подготовки  
01.03.02 «Прикладная математика и информатика»  
Блок Б1.В.01 «Дисциплины (модули)», часть, формируемая участниками  
образовательных отношений

Профиль подготовки  
Вычислительная математика и информационные технологии

Квалификация (степень) выпускника

Бакалавр

Форма обучения очная

Курс 2,3 семестр 4,5

Москва  
2018

Составитель / составители: МГГЭУ, декан факультета ПМиИ

место работы, занимаемая должность

 Петрунина Е.В. «21» августа 2018 г.

подпись

Ф.И.О.

Дата

**Рецензент:** МГГЭУ, профессор кафедры Информационных технологий и прикладной математики

место работы, занимаемая должность

 Истомина Т.В. «22» августа 2018 г.

подпись

Ф.И.О.

Дата

Согласовано:

*Представитель работодателя или объединения работодателей*

научный сотрудник, ФГБУ ГНЦ Федеральный медицинский биофизический центр имени А.И. Бурназяна ФМБА России

(должность, место работы)

 Васильев Е.В. «26» августа 2018 г.

подпись

Ф.И.О.

Дата

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры Информационных технологий и прикладной математики (протокол № 1 от «26» августа 2018 г.)

/Зав. кафедрой ИТиПМ/  Петрунина Е.В. «26» августа 2018 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-7	<p data-bbox="483 618 1455 734">Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения</p> <p data-bbox="483 745 1455 1070">ПК-7.1. Знает теоретические основы разработки программных и алгоритмических решений в области системного и прикладного программного обеспечения; математические методы решения задач, процедурный и объектно-ориентированный подходы к разработке информационных систем; актуальные проблемы в области программирования; методы и технологии программирования; языки программирования, основы технологии модульного программирования на языках высокого уровня.</p> <p data-bbox="483 1081 1455 1283">ПК-7.2. Умеет применить математический метод для решения задачи; подобрать рациональную технологию программирования для решения профессиональной задачи; создавать программные продукты и алгоритмические решения в области системного и прикладного программного обеспечения.</p> <p data-bbox="483 1294 1455 1538">ПК-7.3. Владеет навыками применения математических методов для решения задач и применения стандартных алгоритмов; навыками разработки и создания алгоритмических и программных решений в области системного и прикладного программного обеспечения; навыками разработки программных приложений с использованием современных языков программирования.</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-7		Знает			
	Недостаточный уровень	ПК-7.1. Студент не способен к разработке и применению объектно-ориентированных программ. Не знает: основ информатики; объектно-ориентированные подходы к разработке информационных систем.	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-	Текущий контроль - опрос, отчет о практической работе

<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>приложений в .NET.</p> <p>Тема 11. Асинхронное программирование.</p> <p>Тема 12. Программирование с использованием пакета Math.Net Numerics.</p>	
Базовый уровень	ПК-7.1. Студент усвоил основное содержание материала дисциплины, но имеет несистематизированные знания об основах информатики; объектно-ориентированных подходах к разработке информационных систем.	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 - опрос, отчет о практической работе	
Средний уровень	ПК-7.1. Студент способен самостоятельно выделять главные положения в изученном материале.	Лекционные и практические занятия, самостоятельная работа обучающихся,	<p>Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#</p>	Текущий контроль - опрос, отчет о практической работе	

		Знает основ информатики; объектно-ориентированные подходы к разработке информационных систем.	подготовка и сдача промежуточной аттестации.	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-7.1. Студент знает, понимает, выделяет главные положения в изученном материале и способен дать краткую характеристику основным идеям проработанного материала дисциплины. Знает основ информатики; объектно-ориентированные подходы к разработке информационных систем. Показывает глубокое	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 - опрос, отчет о практической работе	

		знание и понимание ИТ.		<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>	
		Умеет			
Базовый уровень	ПК-7.2. Студент имеет затруднения при применении математических методов для решения задач; подобрать рациональную технологию программирования.	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</p>	Текущий контроль - опрос, отчет о практической работе	

				использованием пакета Math.Net Numerics.	
Средний уровень	ПК-7.2. Студент умеет решать стандартные профессиональные задачи с применением знаний в области объектно-ориентированного программирования. Умеет применять математические методы для решения задач; подобрать рациональную технологию программирования.	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 - опрос, отчет о практической работе	
Высокий уровень	ПК-7.2. Студент умеет анализировать элементы, устанавливать связи между ними. Умеет решать стандартные профессиональные задачи с применением знаний в	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.	Текущий контроль - опрос, отчет о практической работе	

		области объектно-ориентированного программирования; применять математические методы для решения задач; подобрать рациональную технологию программирования.		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.	
		Владеет			
Базовый уровень	ПК-7.3. Студент на базовом уровне владеет навыками применения математических методов для решения задач и применения стандартных алгоритмов; навыками разработки и создания алгоритмических и программных решений.	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.	Текущий контроль - опрос, отчет о практической работе	

				<p>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-7.3. Студент владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности. Владеет навыками применения математических методов для решения задач и применения стандартных алгоритмов; навыками разработки и создания алгоритмических и программных решений.</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 - опрос, отчет о практической работе</p>	
Высокий уровень	<p>ПК-7.3. Студент владеет концептуально-</p>	<p>Лекционные и практические занятия,</p>	<p>Тема 1. Объектно-ориентированный подход к проектированию и</p>	<p>Текущий контроль - опрос, отчет о</p>	

		<p>понятийным аппаратом, научным языком и терминологией дисциплины. Владеет навыками применения математических методов для решения задач и применения стандартных алгоритмов; навыками разработки и создания алгоритмических и программных решений.</p>	<p>самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.</p>	<p>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>практической работе</p>
--	--	---	---	---	----------------------------

## 2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Таблица 3

№	Наименование оценочного средства	Характеристика оценочного средства	Представление оценочного средства в ФОС
1	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	Практическая работа	Практическая работа представляет собой контрольное мероприятие по учебному материалу каждой темы (раздела) дисциплины, состоящее в индивидуальном выполнении обучающимся практических заданий на заданную тему для оценки полученных знаний, умений и владений компетенциями, формируемыми по данной дисциплине.	Практические задания

### **3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ**

Оценивание результатов обучения по дисциплине Алгебра и геометрия 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-7		Знает	
	Недостаточный уровень Оценка «незачтено», «неудовлетворительно»	ПК-7.1.	<i>Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины</i>
	Базовый уровень Оценка, «зачтено», «удовлетворительно»	ПК-7.1.	<i>Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении</i>
	Средний уровень Оценка «зачтено», «хорошо»	ПК-7.1.	<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-7.1.	<i>Показывает глубокое знание и понимание материала, способен применить изученный материал на практике</i>
		Умеет	
	Базовый уровень	ПК-7.2.	<i>Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач</i>
	Средний уровень	ПК-7.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач</i>
	Высокий уровень	ПК-7.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, показывает глубокое знание и понимание материала, способен решить задачу при изменении формулировки</i>
		Владеет	
	Базовый уровень	ПК-7.3.	<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	ПК-7.3.	<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	ПК-7.3.	<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 практической работы:**

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 обучающие оформляют отчеты, содержащие анализ полученных результатов и выводы.

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

### **Задания в форме устного и письменного опроса**

#### **Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#**

- 1) Основные принципы объектно-ориентированного программирования.
- 2) Критика ООП.
- 3) Отношение между C# и .NET.
- 4) Общеязыковая исполняющая среда.
- 5) Промежуточный язык (IL). Сборки.
- 6) Классы .NET Framework. Переменные.
- 7) Предопределенные типы данных.
- 8) Управление потоком выполнения. Перечисления.
- 9) Пространства имен.
- 10) Консольный ввод-вывод.
- 11) Использование комментариев.
- 12) Рекомендации по программированию на C#.

#### **Тема 2. Объекты и типы**

- 1) Классы. Анонимные типы. Структуры.
- 2) Частичные классы.
- 3) Статические классы.
- 4) Класс Object.

### **Тема 3. Наследование**

- 1) Концепция наследования.
- 2) Типы наследования.
- 3) Наследование реализации.
- 4) Модификаторы.
- 5) Интерфейсы.

### **Тема 4. Обобщения, массивы и кортежи. Коллекции.**

- 1) Обзор обобщений.
- 2) Создание обобщенных классов.
- 3) Простые массивы. Многомерные массивы.
- 4) Зубчатые массивы.
- 5) Класс Array.
- 6) Использование массивов в качестве параметров. Перечисления.
- 7) Кортежи. Интерфейсы и типы коллекций.
- 8) Списки. Словари. Множества. Битовые массивы.

### **Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события.**

#### **Язык интегрированных запросов. Ошибки и исключения.**

- 1) Операции. Безопасность типов.
- 2) Проверка объектов на предмет равенства.
- 3) Перегрузка операций.
- 4) Ссылка на методы.
- 5) Делегаты. События. Обзор LINQ.
- 6) Стандартные операции запросов.
- 7) Классы исключений.
- 8) Перехват исключений.
- 9) Классы исключений, определяемые пользователем.

### **Тема 6. Проектирование ПО.**

- 1) Проектирование с использованием объектов.
- 2) Проектирование с использованием библиотек и шаблонов.
- 3) Проектирование с целью многократного использования кода.

### **Тема 7. Методы разработки ПО.**

- 1) Модели жизненных циклов разработки ПО.
- 2) Методологии разработки ПО.

### **Тема 8. Визуальное программирование графических интерфейсов пользователя.**

- 1) Интерфейсы пользователя.
- 2) Объектно-ориентированное визуальное программирование интерфейсов пользователя в .NET (Windows Forms: окна, элементы управления и компоненты).

- 3) Инфраструктура Windows Presentation Foundation (WPF) как альтернатива Windows Forms. XAML (eXtensible Application Markup Language) – расширяемый язык разметки для приложений.
- 4) Разработка приложений с помощью WPF. Создание документов с помощью WPF.

#### **Тема 9. Доступ к данным в .NET.**

- 1) Обзор объектной модели ADO.NET.
- 2) Подключение к БД. Команды.
- 3) Управление данными и отношениями.
- 4) Программирование с применением Entity Framework.
- 5) Привязка данных к элементам управления Windows Forms.

#### **Тема 10. Основы разработки Web-приложений в .NET.**

- 1) Платформа .NET Framework для Web-приложений.
- 2) ASP.NET Web Forms: модель страницы ASPX, мастер-страницы, навигация.
- 3) Привязка данных к элементам управления ASP.NET Web Forms.
- 4) Обзор ASP.NET MVC.

#### **Тема 11. Асинхронное программирование.**

- 1) Асинхронные шаблоны.
- 2) Основы асинхронного программирования.

#### **Тема 12. Программирование с использованием пакета Math.Net Numerics.**

- 1) Обзор математической библиотеки классов Math.Net Numerics.
- 2) Решение задач линейной алгебры, математической статистики, оптимизации.

### **Контролируемые компетенции: ПК-7**

*Оценка компетенций осуществляется в соответствии с таблицей 4.*

#### **Практические задания**

##### **Вариант 1**

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.

### **Вариант 3**

Что будет выведено в результате выполнения программы?

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            var a1 = new A1();
            Console.WriteLine("1");
            var a2 = new A2();
        }
        catch (Exception)
        {
            Console.WriteLine("3");
        }
        Console.ReadLine();
    }
}
class A1
{
    public int B1;
}
class A2
{
    public int B2;
    public A2(int b2)
    {
        B2 = b2;
    }
}
```

- ```
}  
}  
1) 1  
2) 3  
3) 13  
4) Возникнет ошибка на этапе компиляции
```

#### **Вариант 4**

Что будет выведено в результате выполнения программы?

```
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine(A.B);  
        var a1 = new A();  
        Console.WriteLine(A.B);  
        a1.Write();  
        Console.ReadLine();  
    }  
}  
public class A  
{  
    public static int B;  
    public A()  
    {  
        B = 3;  
    }  
    public void Write()  
    {  
        Console.WriteLine(B);  
    }  
    static A()  
    {  
        B = 5;  
    }  
}
```

- ```
1) 033  
2) 533  
3) 553  
4) Возникнет ошибка на этапе компиляции
```

#### **Вариант 5**

```
class Program  
{  
    static void Main(string[] args)  
    {  
        var a = new A(2) {B = 3};  
        Console.WriteLine(a.B);  
        Console.ReadLine();  
    }  
}  
public class A
```

```
{
public int B { get; set; }
public A(int b)
{
Console.WriteLine("1");
B = b;
}
}
```

- 1) 3
- 2) 12
- 3) 13
- 4) Возникнет ошибка на этапе компиляции

8 Какие утверждения относительно языка C# верны?

- 1) Допустимо множественное наследование
- 2) Класс может реализовать несколько интерфейсов
- 3) Интерфейс может наследоваться от множества других интерфейсов
- 4) Нельзя наследовать от класса, помеченного ключевым словом sealed

9 Что делает оператор %?

- 1) - Переводит дробное число в проценты
- 2) - Возвращает остаток от деления
- 3) - Возвращает процентное соотношение двух операндов
- 4) - Форматирует значения разных типов в строку

10 Сколько родительских классов может иметь производный класс?

- 1) - Не больше одного
- 2) - Всегда один
- 3) - Не больше двух
- 4) - Любое количество

11 Какой класс является базовым для всех классов в C#?

Ответ:

12 Выберите элементы, которые нельзя пометить атрибутом.

- 1) -Интерфейсы
- 2) -Все перечисленное можно пометить атрибутом
- 3) -Возвращаемые значения
- 4) -Классы
- 5) -Методы
- 6) -Структуры

13 Выберите средства, которые предоставляет C# для условной компиляции.

- 1) -Директива #endif
- 2) -Директива #typedef
- 3) -Директива #else
- 4) -Директива #elseif
- 5) -Атрибут Conditional
- 6) -Директива #define
- 7) -Директива #if

Контролируемые компетенции: ПК-7

*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.

### **Вопросы к экзамену**

1. Эволюция методологий программирования. Парадигмы программирования.
2. Основные принципы объектного подхода.
3. Общезыковая 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 частных свойств класса.
28. Обобщенные классы.
29. Абстрактные классы.
30. Объектно-ориентированная технология доступа к данным ADO.NET Entity Framework.

**Контролируемые компетенции: ПК-7**

*Оценка компетенций осуществляется в соответствии с таблицей 4.*