

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

И.о. проректора по учебно-
методической работе
Хакимов Р.М.



«30»августа 2021г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
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
2021

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика (уровень бакалавриата)», утвержденного приказом Министерства образования и науки Российской Федерации № 922 от 19 сентября 2017 г. Зарегистрировано в Минюсте России 12 октября 2017 г. №48531.

Составители рабочей программы: МГГЭУ, доцент кафедры информационных технологий и прикладной математики


подпись

Истомина Т.В.
Ф.И.О.

«30» августа 2021 г.
Дата

место работы, занимаемая должность

Рецензент: МГГЭУ, доцент кафедры информационных технологий и прикладной математики


подпись

Белоглазов А.А.
Ф.И.О.

«30» августа 2021 г.
Дата

место работы, занимаемая должность

Рабочая программа утверждена на заседании кафедры Информационных технологий и прикладной математики (протокол № 2 от «30» августа 2021 г.)

Зав. кафедрой ИТиПМ —


подпись

Литрофанов Е.П.
Ф.И.О.

«30» августа 2021 г.
Дата

СОГЛАСОВАНО

Начальник
учебного отдела
«30» августа 2021 г.
Дата


подпись

И.Г.Дмитриева
Ф.И.О.

СОГЛАСОВАНО

Декан факультета ПМиИ
«30» августа 2021 г.
Дата


подпись

Е.В. Петрунина
Ф.И.О.

СОГЛАСОВАНО

Заведующая библиотекой
«30» августа 2021 г.
Дата


подпись

В.А. Ахтырская
Ф.И.О.

1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ РАЗДЕЛ

1.1. Цели и задачи изучения учебной дисциплины(модуля)

Цель освоения дисциплины: формирование у студентов теоретических знаний и практических навыков по разработке программного обеспечения (ПО) с помощью методов объектно-ориентированного программирования.

Задачи:

- изучение принципов объектно-ориентированного подхода к разработке ПО;

1.2. Требования к результатам освоения дисциплины

Изучение данной дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций
ПК-2. Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение	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.
	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.
	ПК-2.3. Владеет навыками планирования процесса разработки программного продукта; навыками задания функциональных рамок подсистем; навыками определения наиболее значимых критериев качества программного продукта.
ПК-3. Способен проектировать ИС в соответствии с профилем подготовки по видам обеспечения	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.
	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.
	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.

ПК-8. Способен программировать приложения и создавать программные прототипы решения прикладных задач	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.
	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.
	ПК-8.3. Владеет навыками разработки приложений и программных прототипов

1.3. Место дисциплины (модуля) в структуре образовательной программы направления подготовки 09.03.03 Прикладная информатика (бакалавриат).

Учебная дисциплина «Объектно-ориентированное программирование» относится к части, формируемой участниками образовательных отношений блока «Дисциплин (модулей)» Б1. Изучение учебной «Объектно-ориентированное программирование» дисциплины базируется на знаниях, умениях и навыках, полученных студентами при изучении предшествующих курсов: «Основы информатики», «Алгоритмизация и программирование», «Математическая логика и теория алгоритмов», «Системное и прикладное программное обеспечение», «Компьютерный анализ». Изучение учебной дисциплины «Объектно-ориентированное программирование» необходимо для освоения таких дисциплин, как «Высокоуровневые методы информатики и программирования», «Проектный практикум» и производственной практики «Технологическая (проектно-технологическая)».

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. Объем дисциплины и виды учебной работы в соответствии с формами обучения

Объем дисциплины «Объектно-ориентированное программирование» составляет 5 з.е./180 часов:

Вид учебной работы	Всего, часов	Курс, часов	
		4 курс	
	Очная форма	Семестр 7	Семестр 8
Аудиторная работа обучающихся с преподавателем (по видам учебных занятий), всего в том числе:	68	46	22
Лекции	28	18	10
Практические занятия	40	28	12
Лабораторные занятия			
Самостоятельная работа обучающихся	108	60	48
Промежуточная аттестация (подготовка и сдача), всего:			
Контрольная работа			
Курсовая работа			
Зачет	2	2	
Зачет с оценкой	2		2
Итого:	180/5	108/3	72/2

2.2 Содержание дисциплины по темам (разделам)

№ п/п	Наименование раздела (темы)	Содержание раздела (тематика занятий)	Формируемые компетенции (индекс)
1	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#	Основные принципы объектно-ориентированного программирования. Критика ООП. Отношение между C# и .NET. Общезыковая исполняющая среда. Промежуточный язык (IL). Сборки. Классы .NET Framework. Переменные. Предопределенные типы данных. Управление потоком выполнения. Перечисления. Пространства имен. Консольный ввод-вывод. Использование комментариев. Рекомендации по программированию на C#.	ПК-2, ПК-3, ПК-8
2	Тема 2. Объекты и типы	Классы. Анонимные типы. Структуры. Частичные классы. Статические классы. Класс Object.	ПК-2, ПК-3, ПК-8
3	Тема 3. Наследование	Концепция наследования. Типы наследования. Наследование реализации. Модификаторы. Интерфейсы.	ПК-2, ПК-3, ПК-8
4	Тема 4. Обобщения, массивы и кортежи. Коллекции.	Обзор обобщений. Создание обобщенных классов. Простые массивы. Многомерные массивы. Зубчатые массивы. Класс Array. Использование массивов в качестве параметров. Перечисления. Кортежи. Интерфейсы и типы коллекций. Списки. Словари. Множества. Битовые массивы.	ПК-2, ПК-3, ПК-8
5	Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.	Операции. Безопасность типов. Проверка объектов на предмет равенства. Перегрузка операций. Ссылка на методы. Делегаты. События. Обзор LINQ. Стандартные операции запросов. Классы исключений. Перехват исключений. Классы исключений, определяемые пользователем.	ПК-2, ПК-3, ПК-8
6	Тема 6. Проектирование ПО.	Проектирование с использованием объектов, проектирование с использованием библиотек и шаблонов, проектирование с целью многократного использования кода.	ПК-2, ПК-3, ПК-8
7	Тема 7. Методы разработки ПО.	Модели жизненных циклов разработки ПО, методологии разработки ПО.	ПК-2, ПК-3, ПК-8
8	Тема 8. Визуальное программирование графических интерфейсов пользователя.	Интерфейсы пользователя. Объектно-ориентированное визуальное программирование интерфейсов пользователя в .NET (Windows Forms: окна, элементы управления и	ПК-2, ПК-3, ПК-8

		компоненты). Инфраструктура Windows Presentation Foundation (WPF) как альтернатива Windows Forms. XAML (eXtensible Application Markup Language) – расширяемый язык разметки для приложений. Разработка приложений с помощью WPF. Создание документов с помощью WPF.	
9	Тема 9. Доступ к данным в .NET.	Обзор объектной модели ADO.NET. Подключение к БД. Команды. Управление данными и отношениями. Программирование с применением Entity Framework. Привязка данных к элементам управления Windows Forms.	ПК-2, ПК-3, ПК-8
10	Тема 10. Основы разработки Web-приложений в .NET.	Платформа .NET Framework для Web-приложений. ASP.NET Web Forms: модель страницы ASPX, мастер-страницы, навигация. Привязка данных к элементам управления ASP.NET Web Forms. Обзор ASP.NET MVC.	ПК-2, ПК-3, ПК-8
11	Тема 11. Асинхронное программирование.	Асинхронные шаблоны. Основы асинхронного программирования.	ПК-2, ПК-3, ПК-8
12	Тема 12. Программирование с использованием пакета Math.Net Numerics.	Обзор математической библиотеки классов Math.Net Numerics. Решение задач линейной алгебры, математической статистики, оптимизации.	ПК-2, ПК-3, ПК-8

2.3. Разделы дисциплин и виды занятий

№ п/п	Наименование темы дисциплины	Лекционные занятия	Практические занятия	Самостоятельная работа	Всего часов	Формы текущего контроля успеваемости
1.	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#	2	2	4	6	Опрос
2.	Тема 2. Объекты и типы	4	4	20	30	Опрос, контрольная работа
3.	Тема 3. Наследование	2	4	20	26	Опрос
4.	Тема 4. Обобщения, массивы и кортежи. Коллекции.	2	4	10	16	Опрос
5.	Тема 5. Операции и приведения. Делегаты, лямбда-выражения и	2	6	10	18	Опрос

	события. Язык интегрированных запросов. Ошибки и исключения.					
6.	Тема 6. Проектирование ПО.	2	2	10	12	Опрос
7.	Тема 7. Методы разработки ПО.	2	2	10	12	Опрос
8.	Тема 8. Визуальное программирование графических интерфейсов пользователя.	2	4	4	12	Опрос
9.	Тема 9. Доступ к данным в .NET.	2	4	10	16	Опрос, тестирование
10.	Тема 10. Основы разработки Web-приложений в .NET.	4	4	4	12	Опрос
11.	Тема 11. Асинхронное программирование.	2	2	10	14	Опрос, тестирование
12.	Тема 12. Программирование с использованием пакета Math.Net Numerics.	2	2	6	12	Опрос

2.4.Планы теоретических (лекционных) занятий

№	Наименование тем лекций	Кол-во часов в 7 семестре
	7 семестр	12
Тема 1 Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#		
1	Лекция 1: Введение в C#. Переменные. Предопределенные типы данных. Управление потоком выполнения. Перечисления. Пространства имен. Консольный ввод-вывод. Использование комментариев. Рекомендации по программированию на C#. Основные принципы объектно-ориентированного программирования. Критика ООП.	2
Тема 2 Объекты и типы		
1	Лекция 1: Классы. Анонимные типы. Структуры.	2
2	Лекция 2: Частичные классы. Статические классы. Класс Object.	2
Тема 3 Наследование		
1	Лекция 1: Концепция наследования. Типы наследования.	2
Тема 4 Обобщения, массивы и кортежи. Коллекции.		
1	Лекция 1: Обзор обобщений. Создание обобщенных классов. Простые массивы. Многомерные массивы. Зубчатые массивы. Класс Array. Использование массивов в качестве параметров. Перечисления. Кортежи.	2
Тема 5 Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.		

1	Лекция 1: Операции. Безопасность типов. Проверка объектов на предмет равенства. Перегрузка операций. Ссылка на методы. Делегаты. События.	2
8 семестр		16
Тема 6 Проектирование ПО		
1	Проектирование с использованием объектов, проектирование с использованием библиотек и шаблонов, проектирование с целью многократного использования кода.	2
Тема 7 Методы разработки ПО		
1	Интерфейсы пользователя.	2
Тема 8 Визуальное программирование графических интерфейсов пользователя.		
1	Лекция 1: Введение в Windows-программирование на C#.	2
Тема 9 Доступ к данным в .NET.		
1	Лекция 1: Обзор объектной модели ADO.NET. Подключение к БД. Команды.	2
Тема 10 Основы разработки Web-приложений в .NET		
1	Платформа .NET Framework для Web-приложений. ASP.NET Web Forms: модель страницы ASPX, мастер-страницы, навигация.	2
2	Привязка данных к элементам управления ASP.NET Web Forms. Обзор ASP.NET MVC.	2
Тема 11 Асинхронное программирование.		
1	Асинхронные шаблоны. Основы асинхронного программирования.	2
Тема 12 Программирование с использованием пакета Math.Net Numerics.		
1	Программирование с использованием пакета Math.Net Numerics.	2

2.5. Планы практических (семинарских) занятий

№	Наименование тем практических (семинарских) занятий	Кол-во часов в 7 семестре
7 семестр		28
Тема 1 Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#		
1	Архитектура .NET. Основы C#	2
Тема 2 Объекты и типы		
1.	Занятие 1: Переменные. Предопределенные типы данных. Управление потоком выполнения. Пространства имен. Консольный ввод-вывод. Использование комментариев. Классы. Анонимные типы. Структуры	2
2	Занятие 2: Создание обобщенных классов.	2
Тема 3 Наследование		
1	Занятие 1: Наследование реализации. Модификаторы. Интерфейсы.	2
2	Занятие 2: Частичные классы. Статические классы. Класс Object.	2
Тема 4 Обобщения, массивы и кортежи. Коллекции.		
1	Занятие 1: Простые массивы. Многомерные массивы. Зубчатые массивы. Класс Array.	2
2	Занятие 2: Интерфейсы и типы коллекций. Списки. Словари. Множества. Битовые массивы.	2
Тема 5 Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.		
1	Занятие 1: Операции. Безопасность типов. Проверка объектов на предмет равенства	2

2	Занятие 2: Перегрузка операций. Ссылка на методы.	2
3	Занятие 3: Делегаты. События.	2
8 семестр		20
Тема 6 Проектирование ПО		
1	Проектирование ПО	2
Тема 7 Методы разработки ПО		
1	Методы разработки ПО	2
Тема 8 Визуальное программирование графических интерфейсов пользователя.		
1	Занятие 1: Объектно-ориентированное визуальное программирование интерфейсов пользователя в .NET (Windows Forms: окна, элементы управления и компоненты).	2
2	Занятие 2: XAML (eXtensible Application Markup Language) – расширяемый язык разметки для приложений. Разработка приложений с помощью WPF.	2
Тема 9 Доступ к данным в .NET.		
1	Занятие 1: Обзор объектной модели ADO.NET. Подключение к БД. Команды.	2
2	Занятие 2: Управление данными и отношениями. Программирование с применением Entity Framework. Привязка данных к элементам управления Windows Forms.	2
Тема 10 Основы разработки Web-приложений в .NET		
1	Занятие 1: Платформа .NET Framework для Web-приложений. ASP.NET Web Forms: модель страницы ASPX, мастер-страницы, навигация.	2
2	Занятие 2: Привязка данных к элементам управления ASP.NET Web Forms.	2
Тема 11 Асинхронное программирование.		
1	Асинхронные шаблоны. Основы асинхронного программирования.	2
Тема 12 Программирование с использованием пакета Math.Net Numerics.		
1	Занятие 1, 2: Решение задач линейной алгебры, математической статистики, оптимизации.	2

2.6. Планы лабораторных работ – не предусмотрено.

2.7. Планы самостоятельной работы обучающегося по дисциплине (модулю)

№	Название разделов и тем	Виды самостоятельной работы	Трудоемкость	Формируемые компетенции	Формы контроля
1	Тема 1. Объектно-ориентированный подход к проектированию и разработке программ. Архитектура .NET. Основы C#	Работа с источниками: Архитектура .NET. Основы C#	4	ПК-2, ПК-3, ПК-8	Устный опрос
2	Тема 2. Объекты и типы	Частичные классы. Статические классы. Класс Object.	20	ПК-2, ПК-3, ПК-8	Устный опрос

3	Тема 3. Наследование	Работа с источниками по темам: Концепция наследования. Типы наследования.	20	ПК-2, ПК-3, ПК-8	Устный опрос
4	Тема 4. Обобщения, массивы и кортежи. Коллекции.	Работа с источниками Информационный поиск по темам: Интерфейсы и типы коллекций. Списки. Словари. Множества. Битовые массивы.	10	ПК-2, ПК-3, ПК-8	Устный опрос
5	Тема 5. Операции и приведения. Делегаты, лямбда-выражения и события. Язык интегрированных запросов. Ошибки и исключения.	Работа с источниками по темам: Стандартные операции запросов. Классы исключений. Перехват исключений.	10	ПК-2, ПК-3, ПК-8	Устный опрос
6	Тема 6. Проектирование ПО.	Работа с источниками Информационный поиск по темам: использованием библиотек и шаблонов, проектирование с целью многократного использования кода	10	ПК-2, ПК-3, ПК-8	Устный опрос
7	Тема 7. Методы разработки ПО.	Работа с источниками по теме: Модели жизненных циклов разработки ПО	10	ПК-2, ПК-3, ПК-8	Устный опрос
8	Тема 8. Визуальное программирование графических интерфейсов пользователя.	Работа с источниками по темам: Разработка приложений с помощью WPF. Создание документов с помощью WPF.	4	ПК-2, ПК-3, ПК-8	Устный опрос
9	Тема 9. Доступ к данным в .NET.	Работа с источниками по темам: Управление данными и отношениями. Программирование с применением Entity Framework.	10	ПК-2, ПК-3, ПК-8	Устный опрос

10	Тема 10. Основы разработки Web-приложений в .NET.	Работа с источниками Информационный поиск по темам: Web Forms. Обзор ASP.NET MVC.	4	ПК-2, ПК-3, ПК-8	Устный опрос
11	Тема 11. Асинхронное программирование.	Работа с источниками по темам: Основы асинхронного программирования.	10	ПК-2, ПК-3, ПК-8	Устный опрос
12	Тема 12. Программирование с использованием пакета Math.Net Numerics.	Работа с источниками по темам: Обзор математической библиотеки классов Math.Net Numerics.	6	ПК-2, ПК-3, ПК-8	Устный опрос

3. ОСОБЕННОСТИ ОБУЧЕНИЯ ЛИЦ С ОВЗ (ПОДА)

Обучение лиц с ограниченными возможностями здоровья осуществляется с учетом индивидуальных психофизических особенностей, а для инвалидов также в соответствии с индивидуальной программой реабилитации инвалида.

Для получения обучающимися, имеющими ограниченные физические возможности, качественного образования должны выполняться следующие важные условия: обучающийся должен иметь возможность беспрепятственно посещать образовательное учреждение и использовать в своём обучении дистанционные образовательные технологии.

Для обучения и контроля обучающийся с нарушениями координации движений предусмотрено проведение тестирования с использованием компьютера.

Во время аудиторных занятий обязательно использование средств обеспечения наглядности учебного материала с помощью мультимедийного проектора. Скорость изложения материала должна учитывать ограниченные физические возможности обучающихся.

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение дисциплины для организации самостоятельной работы студентов (содержит перечень основной литературы, дополнительной литературы, программного обеспечения и Интернет-ресурсы).

В распоряжении преподавателей и обучающихся имеется основное необходимое материально-техническое оборудование, Интернет-ресурсы, доступ к полнотекстовым электронным базам, книжный фонд библиотеки Московского государственного гуманитарно-экономического университета.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1. Основная литература

Комлев, Н. Ю. Объектно Ориентированное Программирование. Хорошая книга для Хороших Людей : практическое пособие / Н. Ю. Комлев. - Москва : СОЛОН-Пресс, 2020. - 298 с. - ISBN 978-5-91359-138-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1858782>

5.2. Перечень дополнительной литературы

1. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для прикладного бакалавриата / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2019. — 206 с. — (Университеты России). — ISBN 978-5-534-00849-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/434045>
2. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для академического бакалавриата / С. В. Зыков. — Москва : Издательство Юрайт, 2019. — 155 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00850-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/434106>

5.3. Программное обеспечение

1. Сетевой компьютерный класс, оснащенный современной техникой
2. Офисный программный пакет (например, Microsoft Office 2007 или более поздних версий).
3. Web-браузер Edge, Mozilla Firefox или Google Chrome
4. ПО для вывода на экран для проектора
5. Платформа Java.
6. Сетевой симулятор JavaNetSim.
7. Менеджер виртуальных машин VMware Player или VirtualBox.

5.4. Электронные ресурсы

1. Национальный открытый университет ИНТУИТ [Электронный ресурс]. URL: <http://www.intuit.ru>
2. Хабрахабр [Электронный ресурс]. URL: <http://habrahabr.ru/>.
3. <http://www.lessons-tva.info/> - На сайте представлены различные учебные материалы, в том числе онлайн учебники (авторские курсы) по дисциплинам: экономическая информатика, компьютерные сети и телекоммуникации, основы электронного бизнеса, информатика и компьютерная техника.
4. Электронная библиотека «Знаниум»: <https://znanium.com/>
5. Электронная библиотека «Юрайт»: <https://urait.ru/>
6. Научная электронная библиотека «Elibrary.ru»: <https://www.elibrary.ru/defaultx.asp>

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ (МОДУЛЯ)

№п/п	Наименование оборудованных учебных кабинетов, лабораторий	Перечень оборудования и технических средств обучения
1	Лекционная аудитория	Мультимедийный проектор, интерактивная доска
2	Компьютерный класс	Компьютеры MXP Pentium, мониторы LG), принтеры, мультимедиа проектор –1. Терминалы подключения к сети Internet.

7. ОЦЕНКА КОМПЕТЕНЦИЙ ПО ИЗУЧАЕМОЙ ДИСЦИПЛИНЕ

№	Критерии оценки	
	«незачтено»	«зачтено»
ЗНАТЬ		
1	<p>Студент не способен самостоятельно выделять главные положения в изученном материале дисциплины.</p> <p>Не знает программные шаблоны; метрики и риски тестирования. Не имеет представления о основных языках программирования приложений; теоретических и методических основ технологии программирования, анализа и применения алгоритмических и программных решений; методах и приёмах разработки программных прототипов решения прикладных задач.</p> <p>Студент не 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>Студент самостоятельно выделяет главные положения в изученном материале.</p> <p>Знает основные программные шаблоны; метрики и риски тестирования.</p> <p>Студент показывает хорошее знание методов разработки, анализа и проектирования ПО; функционального и технического проектирования; паттерны проектирования; номенклатуры инструментальных средств, поддерживающих создание программного обеспечения; техники распределения задач на разработку между исполнителями.</p> <p>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>
УМЕТЬ		
2	<p>Студент не умеет самостоятельно 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; реализовывать программные продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p>	<p>Студент 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; реализовывать программные продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p>

	Студент не умеет работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.	Студент показывает знание и умение работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.
ВЛАДЕТЬ		
3	Студент не владеет навыками планирования процесса разработки программного продукта. Студент не имеет навыки определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта; навыками проектирования прикладных программных продуктов. Студент не показывает базовых владений навыками разработки приложений и программных прототипов.	Студент владеет навыками планирования процесса разработки программного продукта. Студент имеет навыки определения содержания работ по созданию программного продукта; навыками проектирования прикладных программных продуктов. Студент показывает хорошее владение навыками разработки приложений и программных прототипов

№	Критерии оценки			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
ЗНАТЬ				
1	Студент не способен самостоятельно выделять главные положения в изученном материале дисциплины. Не знает программные шаблоны; метрики и риски тестирования. Не имеет представления о основных языках программирования приложений; теоретических и методических основ технологии программирования, анализа и применения алгоритмических и программных решений; методах и приёмах разработки программных прототипов решения прикладных задач.	Студент усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания об основных программных шаблоны; метрика и рисках тестирования. Студент показывает поверхностное знание методов разработки, анализа и проектирования ПО; функционального и технического проектирования; паттерны проектирования; номенклатуры инструментов средств, поддерживающих создание программного обеспечения;	Студент способен самостоятельно выделять главные положения в изученном материале. Знает основные программные шаблоны; метрики и риски тестирования. Студент знает и понимает методы разработки, анализа и проектирования ПО; функциональное и техническое проектирование; паттерны проектирования; номенклатуру инструментов средств, поддерживающих создание программного обеспечения;	Студент знает, понимает, выделяет главные положения в изученном материале и способен дать краткую характеристику основным идеям проработанного материала дисциплины. Знает основные программные шаблоны; метрики и риски тестирования. Студент показывает глубокое знание методов разработки, анализа и проектирования ПО; функционального и технического проектирования; паттерны проектирования; номенклатуры инструментов средств, поддерживающих создание программного обеспечения;

	Студент не 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.	поддерживающих создание программного обеспечения; техники распределения задач на разработку между исполнителями. 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.	техники распределения задач на разработку между исполнителями. 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.	программного обеспечения; техники распределения задач на разработку между исполнителями. 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.
--	--	---	---	---

УМЕТЬ

2	Студент не умеет самостоятельно 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; реализовывать программные	Студент испытывает затруднения самостоятельно 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	Студент умеет самостоятельно 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	Студент 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; реализовывать программные продукты для решения прикладных задач;
---	---	---	--	--

	<p>продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p> <p>Студент не умеет работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.</p>	<p>атрибутов качества; реализовывать программные продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p> <p>Студент испытывает затруднения при работе с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.</p>	<p>атрибутов качества; реализовывать программные продукты для решения прикладных задач; описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p> <p>Студент умеет хорошо работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.</p>	<p>описывать архитектуру приложений включая выделение функциональных компонентов и модулей, структур данных, внешних и внутренних интерфейсов.</p> <p>Студент показывает отличное знание и умение работать с инструментальными средствами, поддерживающими создание программного обеспечения для информационных систем.</p>
ВЛАДЕТЬ				
3	<p>Студент не владеет навыками планирования процесса разработки программного продукта. Студент не имеет навыки определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта; навыками проектирования прикладных программных продуктов. Студент не показывает базовых владений навыками разработки</p>	<p>Студент владеет базовыми навыками планирования процесса разработки программного продукта. Студент имеет навыки определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта; навыками проектирования прикладных программных продуктов. Студент показывает базовое</p>	<p>Студент владеет знаниями всего изученного материала, навыками планирования процесса разработки программного продукта. Студент имеет навыки определения содержания работ по созданию программного продукта; приемами работы с инструментальными средствами автоматизации проектирования и реализации программного продукта; навыками проектирования</p>	<p>Студент отлично владеет навыками планирования процесса разработки программного продукта. Студент имеет навыки определения содержания работ по созданию программного продукта; навыками проектирования прикладных программных продуктов. Студент показывает отличное владение навыками разработки приложений и программных прототипов.</p>

	приложений и программных прототипов.	владение навыками разработки приложений и программных прототипов.	прикладных программных продуктов. Студент показывает средний уровень владения навыками разработки приложений и программных прототипов.	
	Компетенция или ее часть не сформирована	Компетенция или ее часть сформирована на базовом уровне	Компетенция или ее часть сформирована на среднем уровне	Компетенция или ее часть сформирована на высоком уровне

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Интерактивные образовательные технологии, используемые в аудиторных занятиях и самостоятельной работе обучающихся не предусмотрены.

9. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

9.1. Организация входного, текущего и промежуточного контроля обучения

Входное тестирование – устный опрос

Текущий контроль – письменный опрос

Промежуточная аттестация – зачет, зачет с оценкой

9.2. Тематика рефератов, проектов, творческих заданий, эссе и т.п.

Не предусмотрено.

9.3. Курсовая работа

Не предусмотрено.

9.4. 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.

9.5. Вопросы к зачету с оценкой

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 приватных свойств класса.
28. Обобщенные классы.
29. Абстрактные классы.
30. Объектно-ориентированная технология доступа к данным ADO.NET Entity Framework.

9.6. Контроль освоения компетенций

Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
Устный опрос, опрос	1,2,3,4,5,6,7,8,9,10	ПК-2, ПК-3, ПК-8
Контрольная работа	2	ПК-2, ПК-3, ПК-8
Тестирование	9,11	ПК-2, ПК-3, ПК-8

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]