

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: директор института экономики и управления АПК

Дата подписания: 08.08.2025 16:54:58

Уникальный идентификационный ключ:

1e90b132d9b04dce67585160b015dddf2cb1e6a9



**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –**

**МСХА имени К.А. ТИМИРЯЗЕВА»**

**(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)**

Институт Экономики и управления АПК  
Кафедра Прикладной информатики

УТВЕРЖДАЮ:

Директор института

экономики и управления АПК

Л.И. Хоружий

“ 28 ” 08 2025 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.О.07.02 Современные технологии разработки программного обеспече-**  
**ния**

для подготовки магистров

ФГОС ВО

Направление: 09.04.03 Прикладная информатика

Направленности:

«ИТ-новации и цифровые решения для бизнеса»

«Архитектура систем искусственного интеллекта»

Курс 1, 2

Семестры 2,3,4

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчик (и): Журавлев М.В., к.ф./м.н., доцент

(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Рецензент: Ашмарина Т.И., д.э.н., доцент

(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО, профессионального стандарта и учебного плана по направлению подготовки 09.04.03 Прикладная информатика.

И.о. зав. кафедрой прикладной информатики Е.В. Худякова д.э.н., профессор

(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

**Согласовано:**

Председатель учебно-методической  
комиссии института:

Гупалова Т.Н., к.э.н., доцент

(ФИО, ученая степень, ученое звание)



«28» августа 2025 г.

И.о. Зав. выпускающей кафедрой прикладной информатики

Худякова Е.В., д.э.н., профессор

(ФИО, ученая степень, ученое звание)



(подпись)

«28» августа 2025 г.

Заведующий отделом комплектования ЦНБ



## СОДЕРЖАНИЕ

<b>1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....</b>	<b>4</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ .....</b>	<b>5</b>
<b>3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ .....</b>	<b>5</b>
<b>4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ .....	6
ПО СЕМЕСТРАМ .....	6
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	10
4.3 ЛЕКЦИИ/ЛАБОРАТОРНЫЕ/ПРАКТИЧЕСКИЕ/ ЗАНЯТИЯ.....	14
<b>5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ .....</b>	<b>20</b>
<b>6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>21</b>
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ .....	21
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ .....	23
<b>7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....</b>	<b>24</b>
7.1 Основная ЛИТЕРАТУРА .....	24
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	24
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).....</b>	<b>25</b>
<b>9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....</b>	<b>25</b>
<b>10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ).....</b>	<b>26</b>
<b>11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .</b>	<b>27</b>
Виды и формы отработки пропущенных занятий .....	28
<b>12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....</b>	<b>28</b>

## АННОТАЦИЯ

### рабочей программы учебной дисциплины

**Б1.О.07.02 «Современные технологии разработки программного обеспечения» для подготовки магистров по направлению 09.04.03 Прикладная информатика направленности - «ИТ-новации и цифровые решения для бизнеса» «Архитектура систем искусственного интеллекта»**

**Цель освоения дисциплины:** изучение теоретических основ современных технологий программирования и получение практических навыков их реализации систематическое изложение принципов математического моделирования; формирование теоретических и практических навыков по разработке надежного, качественного программного обеспечения с применением современных технологий программирования, методов и средств коллективной разработки.

**Место дисциплины в учебном плане:** дисциплина обязательную часть учебного плана по направлению подготовки 09.04.03 Прикладная информатика, осваивается в 2-4 семестрах.

**Требования к результатам освоения дисциплины:** в результате освоения дисциплины формируются следующие компетенции: ОПК-2.1; ОПК-2.2; ОПК-5.1; ОПК-5.2

**Краткое содержание дисциплины:** Дисциплина «Современные технологии разработки программного обеспечения» даёт навыки организации процесса создания ПО: от анализа требований и проектирования до тестирования, внедрения и сопровождения. Изучаются методологии (Agile, Scrum), инструменты (Git, CI/CD, Docker), стандарты, командная работа.

**Общая трудоемкость дисциплины/в т.ч. практическая подготовка:**  
**360/10** (часы/зач. ед.)

**Промежуточный контроль:** зачет с оценкой – во 2, 3 семестрах, экзамен – в 4 семестре.

### 1. Цель освоения дисциплины

Целью освоения дисциплины «Современные технологии разработки программного обеспечения» является формирование у обучающихся компетенций, обеспечивающих способность к использованию технологий искусственного интеллекта (ИИ), интеллектуальных информационных технологий (ИТ) и систем в профессиональной деятельности, а также способность практического использования навыков получения и формализации знаний, а также навыков применения методов и технологий ИИ для моделирования сложных систем и процессов. Учебный процесс по освоению дисциплины направлен на использование современных цифровых технологий, в том числе цифровых сервисов, моделей и программного обеспечения на основе технологий ИИ.

## **2. Место дисциплины в учебном процессе**

Дисциплина «Современные технологии разработки программного обеспечения» относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана. Дисциплина «Современные технологии разработки программного обеспечения» реализуется в соответствии с требованиями ФГОС ВО, профессионального стандарта, ОПОП ВО и Учебного плана по направлению 09.04.03 Прикладная информатика

Предшествующими курсами, на которых непосредственно базируется дисциплина «Современные технологии разработки программного обеспечения» являются дисциплины по программированию бакалавриата.

Дисциплина «Современные технологии разработки программного обеспечения» является основополагающей для подготовки к сдаче и сдача государственного экзамена, ВКР.

Особенностью дисциплины является получение знаний и навыков использования искусственного интеллекта в задачах оптимизации профессиональной деятельности.

Рабочая программа дисциплины «Современные технологии разработки программного обеспечения» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

## **3. Перечень планируемых результатов обучения по дисциплине**

**Б1.О.07.02 «Современные технологии разработки программного обеспечения», соотнесенных с планируемыми результатами освоения образовательной программы.**

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

## **4. Структура и содержание дисциплины**

Курс проходит во 2, 3 и 4 семестре и рассчитан на 26 часов лекций, 74 часов практических занятий, 229,9 часов самостоятельной работы и 1,1 КРА.

Текущая аттестация студентов - оценка знаний и умений проводится на практических занятиях с помощью защиты практических работ и оценки самостоятельной работы студентов.

Промежуточный контроль проводится в форме зачета во 2 и 3 семестре, а в форме экзамена в 4 семестра.

#### **4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам**

Общая трудоёмкость дисциплины составляет 10 зач. ед. (360 часов), их распределение по видам работ семестрам представлено в таблице 2А, таблице 2Б, таблице 2С.

Таблица 1

## Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1	ОПК-2	Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач	ОПК-2.1 Знать современные интеллектуальные технологии для решения профессиональных задач	современные интеллектуальные технологии для решения профессиональных задач	-	-
			ОПК-2.2 Уметь обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач	-	обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач	--
2	ОПК-5	Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	ОПК-5.1 Знать современное программное и аппаратное обеспечение информационных и автоматизированных систем	современное программное и аппаратное обеспечение информационных и автоматизированных систем	-	-

			<p>ОПК-5.2</p> <p>Уметь модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p>	-	<p>модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p>	-
--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------	---	-------------------------------------------------------------------------------------------------------------------------------------------	---

Таблица 2А

## Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость	
	час. всего/*	В т.ч. по семестрам №2
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>108</b>	<b>108</b>
<b>1. Контактная работа:</b>	36,4	36,4
<b>Аудиторная работа</b>	36,4	36,4
<i>лекции (Л)</i>	8	8
<i>практические занятия (ПЗ)</i>	28	28
<i>курсовая работа (проект) (КР/КП) (консультация, защита)</i>	-	-
<i>консультации перед экзаменом</i>	-	-
<i>контактная работа на промежуточном контроле (КРА)</i>	0,35	0,35
<b>2. Самостоятельная работа (СРС)</b>	<b>71,65</b>	<b>71,65</b>
<i>курсовая работа (подготовка)</i>	-	-
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)</i>	62,65	62,65
<i>Подготовка к зачет с оценкой (контроль)</i>	9	9
Вид промежуточного контроля:	Зачет с оценкой	

\* в том числе практическая подготовка

Таблица 2Б

Вид учебной работы	Трудоёмкость	
	час. всего/*	В т.ч. по семестрам №3
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>108</b>	<b>108</b>
<b>1. Контактная работа:</b>	28,35	28,35
<b>Аудиторная работа</b>	28,35	28,35
<i>лекции (Л)</i>	8	8
<i>практические занятия (ПЗ)</i>	20	20
<i>курсовая работа (проект) (КР/КП) (консультация, защита)</i>	-	-
<i>консультации перед экзаменом</i>	-	-
<i>контактная работа на промежуточном контроле (КРА)</i>	0,35	0,35
<b>2. Самостоятельная работа (СРС)</b>	<b>79,65</b>	<b>79,65</b>
<i>курсовая работа (подготовка)</i>	-	-
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)</i>	70,65	70,65
<i>Подготовка к зачет с оценкой (контроль)</i>	9	9
Вид промежуточного контроля:	Зачет с оценкой	

\* в том числе практическая подготовка

Таблица 2С

Вид учебной работы	Трудоёмкость
--------------------	--------------

	час. всего/*	В т.ч. по семестрам
		№4
<b>Общая трудоёмкость</b> дисциплины по учебному плану	<b>144</b>	<b>144</b>
<b>1. Контактная работа:</b>	38,4	38,4
<b>Аудиторная работа</b>	38,4	38,4
<i>лекции (Л)</i>	10	10
<i>практические занятия (ПЗ)</i>	26	26
<i>курсовая работа (проект) (КР/КП) (консультация, защита)</i>	-	-
<i>консультации перед экзаменом</i>	2	2
<i>контактная работа на промежуточном контроле (КРА)</i>	0,4	0,4
<b>2. Самостоятельная работа (СРС)</b>	<b>78,6</b>	<b>78,6</b>
<i>курсовая работа (подготовка)</i>	-	-
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т.д.)</i>	51,6	51,6
<i>Подготовка к экзамен (контроль)</i>	27	27
Вид промежуточного контроля:		Экзамен

\* в том числе практическая подготовка

## 4.2 Содержание дисциплины

Тематический план по очной форме обучения представлен в таблице 3.

Таблица 3

### Тематический план учебной дисциплины 2 семестр

Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Раздел 1. Общие сведения о языке программирования С# для проектирования и прототипирование алгоритмов ПИ	41	4	16	-	21
Раздел 2. Разработка ПО, общие сведения и планирование ПС. Технологии разработки ПО	51,65	4	16	-	31,65
<i>Контактная работа на промежуточном контроле (КРА)</i>	0,35	-	-	0,35	-
<i>Подготовка к зачету</i>	9				9
<b>Всего за 2 семестр</b>	<b>108</b>	<b>8</b>	<b>28</b>	<b>0,35</b>	<b>71,65</b>

### 3 семестр

Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	

Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Раздел 3. Основы С# для проектирования ПИ	98,65	8	20	-	70,65
Подготовка к зачету	9	-	-	-	9
Контактная работа на промежуточном контроле (КРА)	0,35			0,35	
Всего за 3 семестр	108	8	20	0,35	79,65

#### 4 семестр

Наименование разделов и тем дисциплин	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ	ПКР	
Раздел 4. Методы программирования в С#.	70	6	14	-	50
Раздел 5. ООП. Дополнительные темы.	74	4	12	-	31
Консультация перед экзаменом	2	-	-	2	-
Подготовка к экзамену	27	-	-	24,6	
Контактная работа на промежуточном контроле (КРА)	0,4			0,4	
Всего за 4 семестр	144	10	26	27	81

## Раздел 1. Общие сведения о языке программирования С# для проектирования и прототипирование алгоритмов ПИ

### Тема 1. Виды и структуры алгоритмов, функции.

Типология алгоритмов и их структурные элементы, логика, базовые конструкции (следование, ветвление, цикл) и функции. Свойства алгоритмов (дискретность, детерминированность, результативность), способы записи (псевдокод, блок-схемы, программный код), классификация по виду (линейные, разветвляющиеся, циклические, рекурсивные), роль функций как средства декомпозиции и повторного использования кода.

### Тема 2. Представление данных в С#

Система типов С# для корректного описания данных в алгоритмах. Значимые и ссылочные типы, встроенные примитивы, объявление и инициализация переменных, преобразование типов, работа с массивами и коллекциями. Соответствие между математической моделью данных и их реализацией на С#, точность прототипирования.

## Раздел 2. Разработка ПО, общие сведения и планирование ПС. Технологии разработки ПО

**Тема 3. Разработка и управление требованиями к ПС и алгоритмам ПИ**  
Наши формализации и управления требованиями как основой проектирования ПС. Виды требований (функциональные, нефункциональные, ограничения), методы их выявления и анализа, критерии качества (полнота, непротиворечивость, тестируемость), инструменты документирования (спецификации, use case, user story). Связи требований с алгоритмическими решениями и их прототипированием на С#.

**Тема 4. Моделирование предметной области внедрения ПС. Блок-схемы алгоритмов ПИ ЕСКД. Язык “электронных” блок-схем**

Визуальное моделирование: анализ предметной области, формальное описание алгоритмов в стандартах ЕСКД (Единая система конструкторской документации). Элементы и правила построения блок-схем, их соответствие базовым алгоритмическим структурам, переход от графической нотации к программному коду на С#. Использование специализированных инструментов для создания “электронных” блок-схем и их интеграции в техническую документацию проекта.

**Раздел 3. Основы С# для проектирования ПС**

**Тема 5. Стандарты языка С#.**

Ведение в типы данных. Ссылочный тип данных. Операции - расширения контекста. Операции методы. Встроенные inline-функции. Перегрузка функций. Аргументы по умолчанию. Цикл For. Цикл While. Массивы. Объявление и использование. Переменные, функции, указатели. Классы, структуры, операторов. Исключения. Понимание шаблонов в языке С#. Структура программы на языке С#. Инструкции и выражения. Переменные: типы данных; объявление, инициализация, присваивание. Функции: объявление, определение, вызов. Области видимости: локальные и глобальные переменные.

**Тема 6. Операторы языка С# и алгоритмы.**

Ввод-вывод данных. Функции, задача: алгоритм дерево и клеточный автомат. Ветвления (if-else). Циклы (while, do-while, for). Алгоритмические задачи: рекурсия. Массивы (одномерные и многомерные) и указатели. Строки. Преобразования типов. Задачи: сортировка массива (метод вставки, метод пузырьковой сортировки, быстрая сортировка), поиск подстроки, дампы памяти. Структуры. Ссылки. Передача параметров по значению и по ссылке. Задачи: векторы и матрицы в виде структур и операции над ними.

**Тема 7. Конструкторы и деструкторы С#.**

Динамическое выделение памяти. Работа с файлами. Введение в ООП. Классы и методы. Модификаторы доступа. Константные и статические поля и

методы. Друзья класса. Конструкторы. Деструкторы. Управление ресурсами. Семантика перемещения. Умные указатели. Последовательные контейнеры. Итераторы. Декларация класса. Управление доступом. Указатель this. Дружественные классы и функции. Константные функции и объекты. Статические данные и функции.

## **Раздел 4. Методы программирования в C#.**

### **Тема 8. Метод программирования по шаблонам в C#.**

Правила инициализации и использования. Конструкторы копирования. Наследование как механизм повторного использования кода. Виртуальные функции и позднее связывание. Множественное наследование. Виртуальный базовый класс. Чистые виртуальные функции и абстрактные классы. Области применения шаблонов. Создание и перегрузка шаблонных функций. Определение, специализация и использование шаблонов классов. Применение операторов. Использование виртуальной машины.

### **Тема 9. Дополнительные главы C#.**

Шаблоны с переменным числом параметров. Понятие шаблона с переменным числом параметров. Синтаксис объявления шаблонов с переменным числом параметров. Реализация шаблонов функций с переменным числом параметров.

Реализация шаблонов классов с переменным числом параметров. Шаблон function. Примеры использования шаблона. Свертка параметров шаблона. Виды свертки. Специфика использования. Операции приведения и преобразования типа данных.

### **Тема 10. Опциональные главы C#.**

Универсальные ссылки и механизм. Понятие универсальной ссылки. Цель введения универсальных ссылок. Реализация шаблонов в стандартной библиотеке и библиотек с методами искусственного интеллекта. Реализация шаблонного конструктора, принимающего в качестве параметра универсальную ссылку. Диспетчеризация дескрипторов. Использование шаблонов для решения проблем перегрузки.

## **Раздел 5. Дополнительные темы.**

### **Тема 11. Исключения C#.**

Проблемы обработки исключений. Отслеживание аварийных ситуаций на этапе выполнения. Раскрутка стека при генерации исключения. Средства для обработки межпоточных исключений. Проверки на этапе компиляции. Сравнение шаблонов. Написание кодов программ для ИИ. Универсальное хранилище для любого копируемого объекта.

## Тема 12. Добавления в ядро языка C# и в стандартную библиотеку C#.

Выравнивание данных в памяти. Неэффективность работы с невыровненными данными. Спецификаторы. Расширение функциональности контейнеров стандартной библиотеки. Реализация методов. В каких случаях можно достичь повышения эффективности за счет использования методов. Библиотеки C# с методами искусственного интеллекта.

### 4.3 Лекции/лабораторные/практические/ занятия

Таблица 4

#### Содержание лекций /практических занятий и контрольные мероприятия

1	№ раздела и темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
Раздел 1. Общие сведения о теории множеств и функциональном анализе для проектирования алгоритмов.					
1	Тема 1. Виды и структуры алгоритмов, функции.	Лекция № 1 "Понятия множества и подмножества в " структура языка	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	4
		Практическое занятие " Рекуррентные и итеративные операторы и формулы. Алгоритмы в дискретных множествах, логические алгоритмы."		Устный опрос	6
2	Тема 2. Представление данных в C#	Лекция № 2. Переопределение операторов. Исключения. Понимание шаблонов в языке C#. Структура программы на языке C#. Инструкции и выражения. Переменные: типы данных; объявление, инициализация, присваивание. Функции: объявление, определение, вызов. Области видимости: локальные и глобальные переменные.	ОПК-1.1 ОПК-7.1	-	2

1	№ раздела и темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		Практическое занятие "Операторы C#"		Устный опрос	6
<b>Раздел 2. Общие сведения о теории для проектирования алгоритмов.</b>					
3	Тема 3. Разработка требований к ПС.	Лекция № 3 "Разработка требований к программной системе"	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
		Практическое занятие "Обзор стандартов проектирования ПС"		Устный опрос	3
		Практическое занятие "Стандартизация ПС"		Контрольная работа	3
4	Тема 4. Моделирование предметной области внедрения ПС. Блок-схемы алгоритмов ПИ ЕСКД. Язык "электронных" блок-схем	Лекция № 4 "Блок схемы алгоритмов. Язык "электронных" блок-схем."	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
		Практическое занятие "Модель структуры данных. Общие подходы к организации проектирования ПС. Использование инструмента MS Visio"		Устный опрос	4
<b>Раздел 3. Основы для проектирования программ для МУ</b>					
5	Тема 5. Стандарты языка .	Лекция № 5 «Аргументы по умолчанию. Цикл For. Цикл While. Массивы. Объявление и использование. Переменные, функции, указатели. Классы»	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
		Практическое занятие «структуры, Переопределение операторов. Исключения. Понимание шаблонов в языке . Структура программы на языке »		Устный опрос	4
6	Тема 6. Операторы языка C# и алгоритмы.	Лекция № 6 "Ввод-вывод данных"	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
		Практическое занятие "Ветвления (if-else). Циклы (while, do-while, for).		Защита работы	1

1	№ раздела и темы	№ и название лекций/ практических занятий	Формир уемые компете нции (индикат оры)	Вид контрольного мероприятия	Кол- во час- сов
		Практическое занятие « Массивы (одномерные и многомерные) и указатели. Указательная арифметика. Строки. Преобразования типов.»		Защита работы	1
		Практическое занятие Задачи: сортировка массива (метод вставки, метод пу- зырьковой сортировки, быстрая сортировка), поиск подстроки, дампы памяти.		Защита работы	1
		Практическое занятие «Структуры. Ссылки. Пе- редача параметров по зна- чению и по ссылке. Задачи: векторы и матрицы в виде структур и операции над ними.»		Защита работы	1
		"			
7	Тема 7. Конструк- торы и деструкто- ры С#	Лекция № 7 " Классы и методы. Модифи- каторы доступа. Кон- стантные и статические по- ля и методы. Друзья класса. Конструкторы. "	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
		Практическое занятие " Перегрузка операций. Де- структоры. Управление ре- сурсами. Семантика пере- мещения. Умные указатели. Последовательные контей- неры."		Защита работы	2
		Практическое занятие " Итераторы. Ассоциатив- ные контейнеры. Функто- ры. Декларация класса. Управление доступом. Дружественные классы и функции. Константные функции и объекты. Стати- ческие данные и функции"		Защита работы	2

1	№ раздела и темы	№ и название лекций/ практических занятий	Формир уемые компете нции (индикат оры)	Вид контрольного мероприятия	Кол- во час- сов
Раздел 4. Методы программирования в .					
8	Тема 8. Метод программирования по шаблонам в С#.	<p>Лекция № 8 " Правила инициализации и использования. Правила перегрузки операций в С#. Примеры перегрузки основных операций. Перегрузка операции присваивания и конструктора копирования. Наследование как механизм повторного использования кода. "</p> <p>Практическое занятие " Виртуальные функции и позднее связывание. Множественное наследование. Виртуальный базовый класс. Чистые виртуальные функции и абстрактные классы. "</p>	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	4
				Защита работы	8
9	Тема 9. Дополнительные главы .	<p>Лекция № 9 " Области применения шаблонов. Создание и перегрузка шаблонных функций. Определение, специализация и использование шаблонов классов. "</p> <p>Практическое занятие " Синтаксис объявления шаблонов с переменным числом параметров. Реализация шаблонов функций с переменным числом параметров. Реализация шаблонов классов с переменным числом параметров. "</p>	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	2
				Защита работы	8
10	Тема 10. Опциональные главы С#.	Лекция № 10 Реализация шаблона function. Примеры использования. Реализация шаблона . Примеры использования. Свертка параметров шаблона. Виды свер-	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	8

1	№ раздела и темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ток. Специфика использования. Операции приведения типа."			
		Практическое занятие " Универсальные ссылки (forwarding reference). Механизм forward. Понятие универсальной ссылки. Реализация шаблонного конструктора, принимающего в качестве параметра универсальную ссылку. Диспетчеризация дескрипторов. "		Защита работы	4
		Практическое занятие "Проблемы обработки исключений. Отслеживание аварийных ситуация на этапе выполнения. Раскрутка стека при генерации исключения. Средства для обработки межпоточных исключений."		Контрольная работа	4
Раздел 5. Дополнительные темы по теории алгоритмов.					
11	Тема 11. Исключения C#	Лекция № 11-15 Исключения	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2		10
		Практическое занятие 1 Отслеживание аварийных ситуация на этапе выполнения.		Защита работы	2
		Практическое занятие 2 Классы.		Защита работы	4
		Практическое занятие 3 Функции		Защита работы	2
		Практическое занятие 4 Проверки на этапе компиляции.		Защита работы	2

1	№ раздела и темы	№ и название лекций/ практических занятий	Формир уемые компете нции (индикат оры)	Вид контрольного мероприятия	Кол- во час- сов
12	Тема 12. Добавления в ядро языка C# и в стандартную библиотеку C#	Лекция № 16-20 Специфика использования шаблонов глобальной функции, универсальное хранилище для копируемого объекта	ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2	-	4
		Практическое занятие 1 Специфика использования библиотек.		Защита работы	2
		Практическое занятие 2		Защита работы	2
		Практическое занятие 3 Шаблоны глобальной		Защита работы	2
		Практическое занятие 4 универсальное хранилище для любого копируемого объекта		Защита работы	2
		Практическое занятие 5 Специфика использования		Коллоквиум	4

Перечень вопросов для самостоятельного изучения дисциплины представлен в Таблице 5.

Таблица 5

### Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
1	Тема 1. Виды и структуры алгоритмов, функции.	Рекуррентные и итеративные операторы и формулы. Алгоритмы в дискретных множествах, логические алгоритмы. ОПК-2.1
2	Тема 2. Представление данных в C#	Переопределение операторов. Исключения. Понимание шаблонов в языке . Структура программы на языке . Инструкции и выражения. ОПК-2.1 ОПК-5.1
3	Тема 3. Разработка требований к ПС.	Обзор стандартов проектирования ПС. ОПК-7.3
4	Тема 4. Блок схемы алгоритмов. Язык “электронных” блок схем.	Модель структуры данных. Общие подходы к организации проектирования ПС. Использование инструмента MS Visio. ОПК-5.2
5	Тема 5. Стандарты языка .	Структуры, Переопределение операторов. Исключения. Понимание шаблонов в языке . Структура программы на языке . ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
6	Тема 6. Операторы языка и алгоритмы.	Структуры. Ссылки. Передача параметров по значению и по ссылке. Задачи: векторы и матрицы в виде структур и операции над ними. ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2
7	Тема 7. Конструкторы и деструкторы .	Перегрузка операций. Деструкторы. Управление ресурсами. Семантика перемещения. Умные указатели. Последовательные контейнеры. ОПК-2.1 ОПК-2.2 ОПК-5.1 ОПК-5.2
8	Тема 8. Метод программирования по шаблонам в .	Виртуальные функции и позднее связывание. Множественное наследование. Виртуальный базовый класс. Чистые виртуальные функции и абстрактные классы. ОПК-2.1 ОПК-5.1
9	Тема 9. Дополнительные главы .	Синтаксис объявления шаблонов с переменным числом параметров. Реализация шаблонов функций с переменным числом параметров. Реализация шаблонов классов с переменным числом параметров. ОПК-2.1 ОПК-5.1
10	Тема 10. Опциональные главы .	Проблемы обработки исключений. Отслеживание аварийных ситуация на этапе выполнения. Раскрутка стека при генерации исключения. Средства для обработки межпоточных исключений. ОПК-2.1 ОПК-5.1
11	Тема 11. Исключения . Деревья.	Отслеживание аварийных ситуация на этапе выполнения. Проверки на этапе компиляции. ОПК-2.1 ОПК-5.1
12	Тема 12. Специфика использования шаблонов глобальной функции, универсальное хранилище для копируемого объекта.	Специфика использования шаблонов глобальной функции, универсальное хранилище для копируемого объекта. ОПК-2.1 ОПК-5.1

## 5. Образовательные технологии

Таблица 6

### Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия	Наименование используемых активных и интерактивных образовательных технологий	
1.	Тема 1. Виды и структуры алгоритмов, функции.	Л	Информационные и коммуникационные технологии
		ПЗ	Работа студентов с учебно-методическим порталом, электронными ресурсами.
2.	Тема 2. Представление данных в C#	Л	Информационные и коммуникационные технологии
		ПЗ	Работа студентов с учебно-методическим порталом, электронными

№ п/п	Тема и форма занятия	Наименование используемых активных и интерактивных образовательных технологий	
			ресурсами.
3.	Тема 3. Разработка требований к ПС.	Л	Информационные и коммуникационные технологии
		ПЗ	Работа студентов с учебно-методическим порталом, электронными ресурсами.
4.	Тема 4. Блок схемы алгоритмов. Язык “электронных” блок схем.	Л	Информационные и коммуникационные технологии
		ПЗ	Работа студентов с учебно-методическим порталом, электронными ресурсами.

## 6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины

### 6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

#### 1) Вопросы для устного опроса:

Тема 1. Виды и структуры алгоритмов, функции

Перечислите три базовые алгоритмические структуры и приведите краткий пример каждой.

В чём принципиальное отличие циклического алгоритма от рекурсивного?

Укажите плюсы и минусы обоих подходов.

Что такое декомпозиция задачи через функции? Почему это важно при проектировании алгоритмов?

Опишите, что означает свойство *детерминированности* алгоритма и почему оно критично для программной реализации.

Как блок-схема помогает при отладке алгоритма? Назовите 2–3 стандартных элемента блок-схемы и их назначение.

Тема 2. Представление данных в C#

В чём разница между значимыми value types и ссылочными (reference types) типами в C#? Приведите по одному примеру каждого.

Объясните, что такое неявная типизация через var и когда её уместно использовать.

Какие встроенные типы данных C# вы бы выбрали для хранения: а) возраста человека; б) среднего балла; в) статуса «включено/выключено»? Обоснуйте.

Как работает преобразование типов в C#? Чем отличается явное приведение от неявного?

Что такое массив в C# и как он отличается от списка (List<T>)? В каких случаях предпочтительнее первый, а в каких — второй?

**Тема 3. Разработка требований к ПС**

Назовите 3–4 критерия качества требований к программному средству (ПС).

Почему они важны?

Чем функциональные требования отличаются от нефункциональных? Приведите пример каждого.

Для чего используют use case и user story при сборе требований? В чём их практическое различие?

Почему важно формализовать требования до начала разработки? Опишите 1–2 последствия их отсутствия или неточности.

Как требования к алгоритмам ПИ (программных интерфейсов) влияют на выбор структуры данных и языка программирования? Кратко поясните.

## **2) Примеры заданий для практических работ**

Подробный перечень заданий для практических занятий представлен в оценочных материалах дисциплины.

## **3) Перечень вопросов, выносимых на зачет:**

1. Перечислите три базовые алгоритмические структуры. Приведите краткий пример каждой и укажите, как они отображаются в программном коде (на примере С# или ).
2. В чём принципиальное различие между значимыми (value types) и ссылочными (reference types) типами данных в С#? Приведите по одному примеру каждого типа и поясните, как они хранятся в памяти.
3. Назовите 3–4 ключевых критерия качества требований к программному средству (ПС). Объясните, почему их соблюдение критично на этапе проектирования.
4. Что регламентирует стандарт ЕСКД применительно к блок-схемам алгоритмов? Укажите 2–3 обязательных элемента блок-схемы и их назначение.
5. Опишите синтаксис шаблона глобальной функции в С++. Приведите пример шаблона функции swag и объясните, почему он универсален для разных типов.
6. Что такое std::any в С++? В чём его ключевое преимущество перед типизированными контейнерами и какой существенный недостаток следует учитывать при использовании?
7. Сравните std::any, std::variant и обычный union. Укажите одно преимущество std::any с точки зрения гибкости и один недостаток — с точки зрения производительности.
8. Какие соглашения по именованию (camelCase? Pascal-Case) и др.) приняты в С#? Объясните, почему их соблюдение важно в командной разработке.

9. В чём разница между проверяемыми (checked) и непроверяемыми (unchecked) исключениями в Java? Приведите по одному примеру каждого вида и поясните, как это влияет на компиляцию и выполнение программы.
10. Опишите механизм работы блоков try, catch, Finally в Java. Приведите сценарий, где использование Finally обязательно, и объясните почему.
11. Что такое шаблон проектирования (design pattern)? Назовите три широко используемых шаблона и кратко (1–2 предложения) опишите назначение каждого.
12. В чём состоит задача конструктора в Java? Чем он отличается от обычного метода? Объясните, что такое цепочка конструкторов (this()) и как она помогает избежать дублирования кода.
13. Как приоритет и ассоциативность операторов влияют на результат выражения в Java? Приведите пример выражения, где порядок вычислений неочевиден без знания этих правил.
14. Что такое лямбда-выражения в Java? Покажите на коротком примере, как с помощью Stream и лямбды отфильтровать коллекцию чисел (например, оставить только чётные).
15. Опишите, как блок-схема алгоритма может служить мостом между аналитиком и разработчиком. Приведите пример простой блок-схемы (словесно или схематично) для алгоритма поиска максимума в массиве и укажите, какие элементы блок-схемы соответствуют: а) вводу данных; б) сравнению; в) выводу результата.

## 6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Таблица 7

### Система рейтинговой оценки успеваемости

Баллы	Балльная оценка текущей успеваемости			
	Не защищено	Защищено		
За устный опрос	0	3	4	5
За практическую работу	0	3	4	5

Таблица 8

### Итоговая сумма баллов

Виды контроля	Количество видов контроля	Количество баллов за единицу	Количество баллов
Устный опрос	10	5	50
Защита практической работы	8	5	40
Всего	-	-	90

Таблица 9

### Балльно-рейтинговая система контроля успеваемости

Шкала оценивания	Зачет
50-90	зачтено
0-49	незачтено

## 7. Учебно-методическое и информационное обеспечение дисциплины

### 7.1 Основная литература

1. Казарин, О. В. Надежность и безопасность программного обеспечения: учебник для вузов / О. В. Казарин, И. Б. Шубинский. - Электрон. дан.col. - Москва: Юрайт, 2025. - 342 с. - (Высшее образование). -

URL: <https://urait.ru/bcode/563862>, <https://urait.ru/book/cover/31FCBEAB-6BAB-4F27-9872-8094F5A5B7AB>. - ISBN 978-5-534-05142-1:

2. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения: учебник для вузов / С. А. Чернышев. - Электрон. дан.col. - Москва: Юрайт, 2025. - 176 с. - (Высшее образование). - URL: <https://urait.ru/bcode/567946>, <https://urait.ru/book/cover/00D4E38B-CF37-4973-B5CB-5E6BC04D616C>. - ISBN 978-5-534-14383-6:

### 7.2 Дополнительная литература

1. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения: учебник и практикум для вузов / О. В. Казарин, А. С. Забабури. - Электрон. дан.col. - Москва : Юрайт, 2024. - 312 с. - (Высшее образование). - URL: <https://urait.ru/bcode/538066>, <https://urait.ru/book/cover/4809243E-CF00-4D80-A72E-1E122100378B>. - ISBN 978-5-9916-9043-0:

2. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. - 2-е изд., испр. и доп. - Электрон. дан.col. - Москва: Юрайт, 2023. - 432 с. - (Высшее образование). - URL: <https://urait.ru/bcode/513067>, <https://urait.ru/book/cover/1E4DAA79-4EC0-46B1-8B19-DAD1D6EB3CC7>. - ISBN 978-5-534-07604-2:

3. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для вузов / А. Ф. Тузовский. - Электрон. дан.col. - Москва : Юрайт, 2024. - 219 с. - (Высшее образование). - URL: <https://urait.ru/bcode/537106>, <https://urait.ru/book/cover/07BD54A5-A56B-4941-9EFC-2CBC886555F6>. - ISBN 978-5-534-16300-1:

## 8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. Stepik (<https://stepik.org>)
2. [learn.microsoft.com](https://learn.microsoft.com)

## 9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины (модуля)	Наименование программы	Тип программы	Автор	Год разработки
1	Тема 1. Виды и структуры алгоритмов, функции.	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
3	Тема 2. Представление данных в C#	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 3. Разработка требований к ПС.	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 4. Блок схемы алгоритмов. Язык “электронных” блок схем.	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 5. Стандарты языка .	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 6. Операторы языка и алгоритмы.	Stepik, C#	Онлайн-платформа	Российская компания	2013

			для обучения и курсов	(Stepik Inc.)	
	Тема 7. Конструкторы и деструкторы .	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 8. Метод программирования по шаблонам в .	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 9. Дополнительные главы .	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 10. Опциональные главы .	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 11. Исключения . Деревья.	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013
	Тема 12. Специфика использования шаблонов глобальной функции, универсальное хранилище для копируемого объекта.	Stepik, C#	Онлайн-платформа для обучения и курсов	Российская компания (Stepik Inc.)	2013

**10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Таблица 10

**Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями**

Наименование специальных* помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы**
1	2
Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций (1 корпус, 407 аудитория)	проектор, экран настенный, компьютер
Компьютерный класс (1 корпус, 201 аудитория)	Персональные компьютеры 20 шт. с доступом к интернету, Парты 20 шт. Стулья 20 шт. Доска маркерная 1 шт.
ЦНБ им. Н.И. Железнова	Читальный зал (25 компьютеров)
Общежитие	Комната для самоподготовки

## 11. Методические рекомендации обучающимся по освоению дисциплины

Основными видами обучения студентов по дисциплине являются лекции, практические занятия в компьютерном классе и самостоятельная работа студентов.

Самостоятельная работа студентов по дисциплине «Современные технологии разработки программного обеспечения» направлена на углубление и закрепление знаний, полученных на лекциях и практических занятиях, на развитие практических умений и включает такие виды работ, как:

- работа с лекционным материалом;
- работа с рекомендованной литературой при подготовке к практическим занятиям;
- подготовка к зачету.

При изучении дисциплины «Современные технологии разработки программного обеспечения» используется рейтинговая система оценивания знаний студентов, которая позволяет реализовать непрерывную и комплексную систему оценивания учебных достижений студентов. Непрерывность означает, что текущие оценки не усредняются (как в традиционной технологии), а непрерывно складываются на протяжении семестра при изучении дисциплины. Комплексность означает учет всех форм учебной и самостоятельной работы студента в течение семестра.

Принципы рейтинга: непрерывный контроль (на каждом из аудиторных занятий) и получение более высокой оценки за работу, выполненную в срок. При проведении практических занятий предусмотрено широкое использование активных и интерактивных форм (разбор конкретных ситуаций, устный опрос, защита практических работ).

Бально–рейтинговая система повышает мотивацию студентов.

Промежуточным контролем по дисциплине является зачет.

В результате изучения дисциплины формируются знания и умения в области инструментальных средств, студенты получают опыт в проектирование информационных систем. Каждому студенту во время практических занятий

предоставляется полная возможность быть индивидуальным пользователем компьютера, самостоятельно отрабатывать учебные вопросы и выполнять индивидуальные учебные задания преподавателя.

Основная рекомендация сводится к обеспечению равномерной активной работы студентов над дисциплиной в течение всего семестра: студенты должны прорабатывать курс прослушанных лекций, готовиться к выполнению и защите практических работ, а также выполнять задания, вынесенные на самостоятельную работу. Рекомендуется перед каждой лекцией просматривать содержание предстоящей лекции по учебнику и конспекту с тем, чтобы лучше воспринять материал лекции. Важно помнить, что ни одна дисциплина не может быть изучена в необходимом объеме только по конспектам. Для хорошего усвоения курса нужна систематическая работа с учебной и научной литературой, а конспект может лишь облегчить понимание и усвоение материала.

В подготовке к занятиям по дисциплине студенты должны активно использовать дополнительную литературу, поскольку именно с ее помощью можно получить наиболее полное и верное представление о происходящих в стране и в мире процессах.

### **Виды и формы отработки пропущенных занятий**

Студент, пропустивший занятия обязан его отработать:

- лекцию отрабатывают путем устного ответа по пропущенной теме;
- практическое занятие путем выполнения практической работы, которая выполнялась на данном практическом занятии.

## **12. Методические рекомендации преподавателям по организации обучения по дисциплине**

В процессе обучения по дисциплине «Современные технологии разработки программного обеспечения» используются лекционно-практические занятия, разбор конкретных ситуаций, организуется работа с методическими и справочными материалами, целесообразно применение современных технических средств обучения и информационных технологий. Освоение учебной дисциплины предполагает осмысление её разделов и тем на практических занятиях, в процессе которых студент должен закрепить и углубить теоретические знания.

Дисциплина «Современные технологии разработки программного обеспечения» имеет прикладной характер, её теоретические положения и практические навыки могут быть использованы в будущей практической деятельности.

Промежуточный контроль – зачет.

Рекомендуется определять сроки проведения контрольных мероприятий, максимальная оценка за каждое из них и правила перевода общего количества баллов, полученных при изучении дисциплины, в итоговый результат (зачет).

Выполнение практических заданий является обязательным для всех обучающихся. Студенты, не выполнившие в полном объеме работы, предусмотренные учебным планом, не допускаются к сдаче зачета.

**Программу разработал:**

Журавлев М.В., к.ф.-м.н., доцент



---

(подпись)

**РЕЦЕНЗИЯ**  
**на рабочую программу дисциплины Б1. О.07.02 «Современные технологии**  
**разработки программного обеспечения»**  
**для подготовки магистров по направлению 09.04.03 Прикладная информатика**  
**направленности «ИТ-новации и цифровые решения для бизнеса»**  
**«Архитектура систем искусственного интеллекта»**  
**(квалификация выпускника – магистр)**

Ашмарина Татьяна Игоревна, к.э.н., ФГБОУ ВО РГАУ-МСХА имени К.А. Тимирязева, доцент (далее по тексту рецензент), проведена рецензия рабочей программы дисциплины «Современные технологии разработки программного обеспечения» по направлению 09.04.03 – «Прикладная информатика», направленность «ИТ-новации и цифровые решения для бизнеса» «Архитектура систем искусственного интеллекта» (магистратура) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре прикладной информатики, разработчик – Журавлев М.В., к.ф./м.н., доцент. Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Современные технологии разработки программного обеспечения» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.04.03 – «Прикладная информатика». Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе актуальность учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится обязательной части учебного цикла – Б1.

3. Представленные в Программе цели дисциплины соответствуют требованиям ФГОС ВО направления 09.04.03 – «Прикладная информатика».

4. В соответствии с Программой за дисциплиной «Современные технологии разработки программного обеспечения» закреплено 3 компетенции. Дисциплина «Современные технологии разработки программного обеспечения» и представленная Программа способна реализовать их в объявленных требованиях. Результаты обучения, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

5. Общая трудоёмкость дисциплины «Современные технологии разработки программного обеспечения» составляет 10 зачётных единиц (360 часов).

6. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Современные технологии разработки программного обеспечения» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.04.03 – «Прикладная информатика» и возможность дублирования в содержании отсутствует.

7. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

8. Программа дисциплины «Современные технологии разработки программного обеспечения» предполагает 3 занятия в интерактивной форме.

9. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 09.04.03 – «Прикладная информатика».

10. Представленные и описанные в Программе формы текущей оценки знаний (опрос, как в форме обсуждения отдельных вопросов и выступлений, а также контроль выполнения и проверка отчетности по практическим работам), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме зачета, что соответствует статусу дисциплины, как дисциплины вариативной части учебного цикла – Б1 ФГОС ВО направления **09.04.03 – «Прикладная информатика»**.

11. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

Учебно-методическое обеспечение дисциплины представлено: основной литературой – 2 источника (базовый учебник), дополнительной литературой – 3 наименования, периодическими изданиями – 2 источника со ссылкой на электронные ресурсы, Интернет-ресурсы – 2 источника и соответствует требованиям ФГОС ВО направления **09.04.03 – «Прикладная информатика»**. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины **«Современные технологии разработки программного обеспечения»** и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

12. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине **«Современные технологии разработки программного обеспечения»**.

### ОБЩИЕ ВЫВОДЫ

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины **«Современные технологии разработки программного обеспечения»** ОПОП ВО по направлению **09.04.03 – «Прикладная информатика»**, направленность **««ИТ-новации и цифровые решения для бизнеса» «Архитектура систем искусственного интеллекта»»** (квалификация выпускника – магистр), разработанная Журавлев М.В., к.ф./м.н., доцент соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Ашмарина Т.И., к.э.н., доцент

  
(подпись)

« 28 августа » 2025 г.

