

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: директор института экономики и управления АПК

Дата подписания: 27.02.2026 11:21:14

Уникальный программный 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 статистики и кибернетики

УТВЕРЖДАЮ:

Директор института
экономики и управления АПК

Л.И. Хоружий



«28» августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.12 «Программирование на языке C++»

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленности: Фуллстек разработка

Курс 3

Семестр 6

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

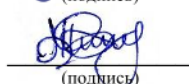
Разработчики:

Демичев В.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Титов А.Д., ассистент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Рецензент:

Кийко П.В., канд. пед. наук.
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, профессионального стандарта и учебного плана 2025 года начала подготовки

Программа обсуждена на заседании кафедры статистики и кибернетики.
Протокол № 11 от «26» августа 2025 г.

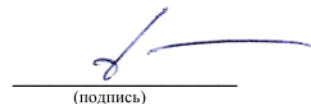
И.о. зав. кафедрой Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Согласовано:

Председатель учебно-методической
комиссии института экономики и управления АПК
Гупалова Т.Н. канд. экон. наук, доцент протокол №1
(ФИО, ученая степень, ученое звание)


(подпись)

«28» августа 2025 г.

И. о. зав. выпускающей кафедрой
статистики и кибернетики
Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2025 г.

Заведующий отделом комплектования ЦНБ


(подпись)

)

СОДЕРЖАНИЕ

АННОТАЦИЯ	4
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	5
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ.....	5
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	9
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ	9
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	9
4.3 ЛЕКЦИИ/ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	10
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	14
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	15
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ.....	15
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ.....	18
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	20
7.1 ОСНОВНАЯ ЛИТЕРАТУРА	20
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	20
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО- ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	20
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ	21
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	21
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	22
Виды и формы отработки пропущенных занятий	22
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	23

АННОТАЦИЯ

рабочей программы учебной дисциплины

Б1.В.12 «Программирование на языке С++»

для подготовки бакалавров по направлению 09.03.02 «Информационные системы и технологии по направленности «Фуллстек разработка»

Цель освоения дисциплины. Основная цель дисциплины «Программирование на языке С++» – овладение студентами основными методами разработки компьютерных программ посредством языка программирования С++ для решения практических задач.

Место дисциплины в учебном плане: дисциплина включена в часть учебного плана по направлению подготовки 09.03.02 «Информационные системы и технологии», формируемую участниками образовательных отношений.

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы): ПКос-4 (ПКос-4.1, ПКос-4.2, ПКос-4.3).

Краткое содержание дисциплины: Среды программирования. Классическая программа на С++. Компиляция. Редактирование связей. Объекты, типы и значения. Ввод данных. Переменные. Операции и операторы. Присваивание и инициализация. Составные операторы присваивания. Имена. Типы и объекты. Безопасность типов. Преобразование типов. Вычисления. Выражения. Инструкции. Инструкция выбора. Итерация. Функции. Объявление функций. Вектор. Обход вектора. Ошибки. Источники ошибок. Ошибки времени компиляции: синтаксические ошибки; ошибки, связанные с типами. Ошибки времени редактирования связей. Ошибки времени выполнения программы. Обработка ошибок в вызывающем коде. Обработка ошибок в вызываемом коде. Сообщения об ошибках. Исключения. Логические ошибки. Отладка. Пред- постусловия. Тестирование. Написание программ. Задача. Стадии, стратегия разработки программ. Лексемы. Грамматики. Превращение грамматики в программу. Выражения. Термы. Первичные выражения. Поток лексем. Структура программы. Объявления и определения. Инициализация. Заголовочные файлы. Область видимости. Функции. Вызов функции и возврат значения. Порядок вычислений. Пространство имен.

Типы, определенные пользователем. Классы и члены класса. Интерфейс и реализация. Разработка класса. Перечисления. Перегрузка операторов. Интерфейсы классов. Векторы и динамически выделяемая память. Память, адреса, указатели. Динамически распределяемая память: размещение, доступ, освобождение памяти. Деструкторы. Указатели на объекты класса. Указатели и ссылки. Указатель this. Доступ к элементам вектора. Массивы. Графические классы. Проектирование графических классов. Графическое представление функций и данных. Графические пользовательские интерфейсы.

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

Промежуточный контроль: зачет.

1. Цель освоения дисциплины

Целью освоения дисциплины «Программирование на языке C++» является овладение студентами основными методами разработки компьютерных программ посредством языка программирования C++ для решения практических задач.

2. Место дисциплины в учебном процессе

Дисциплина «Программирование на языке C++» включена в часть, формируемую участниками образовательных отношений учебного плана. Дисциплина «Компьютерная графика» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению 09.03.02 «Информационные системы и технологии».

Предшествующими курсами, на которых непосредственно базируется дисциплина «Программирование на языке C++» являются «Алгоритмизация и программирование», «Программирование на языке Python».

Дисциплина «Программирование на языке C++» является основополагающей для изучения следующих дисциплин: «Инструментальные средства информационных систем», «Тестирование программного обеспечения», а также подготовки выпускной квалификационной работы.

Особенностью дисциплины является изучение инструментов языка C++ для создания высокопроизводительных компьютерных программ, разработка и программирование программного продукта для решения практических задач.

Рабочая программа дисциплины «Программирование на языке C++» 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, соотнесенных с планируемыми результатами освоения образовательной программы

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

Таблица 1

Требования к результатам освоения учебной дисциплины

№ п/п	Код компете нции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1	ПКос-4	Способен осуществлять разработку, отладку и рефакторинг кода программного обеспечения, интеграцию программных модулей и компонент, в том числе взаимодействующих с внешней средой, средствами выбранных языков программирования	ПКос-4.1 Знать: методы и приемы формализации и алгоритмизации поставленных задач; нотации и программные продукты для графического отображения алгоритмов; алгоритмы решения типовых задач, области и способы их применения; методологии разработки программного обеспечения; синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; особенности выбранной среды программирования; методы и приемы отладки программного кода, повышения читаемости программного кода; типы и форматы сообщений об ошибках, предупреждений	базовые алгоритмические конструкции; синтаксис языка C++: типы данных, переменные, операторы, функции, структуры; принципы работы с памятью; стандартную библиотеку C++; основы модульной разработки: заголовочные файлы, компиляция, линковка	-	-
			ПКос-4.2 Уметь:	-	формализовать простую	-

№ п/п	Код компете нции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
			использовать методы и приемы формализации и алгоритмизации поставленных задач; использовать программные продукты для графического отображения алгоритмов; применять стандартные алгоритмы в соответствующих областях; применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; применять инструментарий для создания и актуализации исходных текстов программ; выявлять ошибки в программном коде, интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов; применять методы и приемы отладки программного кода		задачу в виде алгоритма; выбирать правильные типы данных и структуры для решения задачи; писать читаемый и структурированный код на C++; разделять программу на функции и модули	
			ПКос-4.3 Владеть навыками; составления формализованных описаний решений поставленных задач	-	-	навыками разработки консольных приложений на C++; навыками оформления

№ п/п	Код компете нции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
			в соответствии с требованиями технического задания; разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов; создания программного кода в соответствии с техническим заданием (готовыми спецификациями); оптимизации программного кода с использованием специализированных программных средств; анализа и проверки исходного программного кода; отладки программного кода на уровне программных модулей и межмодульных взаимодействий и взаимодействий с окружением			кода в соответствии с базовыми правилами (стиль, именование, комментирование); навыками создания и использования пользовательских функций и структур; навыками работы с динамической памятью

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 4 зачетные единицы (144 часа), их распределение по видам работ семестрам представлено в таблице 2.

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 2а

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость (семестр № 6)/*
	час.
Общая трудоёмкость дисциплины по учебному плану	144/4
1. Контактная работа	50,25/4
Аудиторная работа	50,25/4
лекции (Л)	16
практические занятия (ПЗ)	34/4
контактная работа на промежуточном контроле (КРА)	0,25
2. Самостоятельная работа (СРС)	93,75
самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)	84,75
подготовка к зачету	9
Вид промежуточного контроля:	Зачет

* в том числе практическая подготовка

4.2 Содержание дисциплины

Таблица 3

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ/*	ПКР	
Раздел 1 Основы программирования на C++	69/4	10	16/4	-	43
Раздел 2 Объектно-ориентированное программирование на C++	74,75	6	18	-	50,75
Контактная работа на промежуточном контроле	0,25	-	-	0,25	-
Итого по дисциплине	144/4	16	34	0,25	93,75

* в том числе практическая подготовка

Раздел 1 Основы программирования на C++

Среды программирования. Классическая программа на C++. Компиляция. Редактирование связей. Объекты, типы и значения. Ввод данных. Переменные. Операции и операторы. Присваивание и инициализация. Составные операторы присваивания. Имена. Типы и объекты. Безопасность типов. Преобразование типов. Вычисления. Выражения. Инструкции. Инструкция выбора. Итерация. Функции. Объявление функций. Вектор. Обход вектора. Ошибки. Источники ошибок. Ошибки времени компиляции: синтаксические ошибки; ошибки, связанные с типами. Ошибки времени редактирования связей. Ошибки времени выполнения программы. Обработка ошибок в вызывающем коде. Обработка ошибок в вызываемом коде. Сообщения об ошибках. Исключения. Логические ошибки. Отладка. Пред- постусловия. Тестирование. Написание программ. Задача. Стадии, стратегия разработки программ. Лексемы. Грамматики. Превращение грамматики в программу. Выражения. Термы. Первичные выражения. Потоки лексем. Структура программы. Объявления и определения. Инициализация. Заголовочные файлы. Область видимости. Функции. Вызов функции и возврат значения. Порядок вычислений. Пространство имен.

Раздел 2 Объектно-ориентированное программирование на C++

Типы, определенные пользователем. Классы и члены класса. Интерфейс и реализация. Разработка класса. Перечисления. Перегрузка операторов. Интерфейсы классов. Векторы и динамически выделяемая память. Память, адреса, указатели. Динамически распределяемая память: размещение, доступ, освобождение памяти. Деструкторы. Указатели на объекты класса. Указатели и ссылки. Указатель this. Доступ к элементам вектора. Массивы. Графические классы. Проектирование графических классов. Графическое представление функций и данных. Графические пользовательские интерфейсы.

4.3 Лекции/практические занятия

Таблица 4

Содержание лекций/практических занятий и контрольные мероприятия

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
1	Раздел 1. Основы программирования на C++				
	Тема 1. Введение в программирование на C++	Лекция №1. История и особенности C++	ПКос-4.1	-	2
		Практическая работа №1. Установка среды разработки. Написание первой	ПКос-4.2, ПКос-4.3	устный опрос	2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемы е компетенции	Вид контрольного о мероприятия	Кол- во часов/ из них практич еская подго товка
		программы. Работа с компилятором (g++). Основные арифметические операции.			
	Тема 2. Управляюще е конструкции и функции	Лекция №2. Условные операторы. Циклы. Функции: объявление, определение, передача параметров.	ПКос-4.1	-	2
		Практическая работа №2. Решение задач с ветвлением и циклами .	ПКос-4.2, ПКос-4.3	защита практической работы	2
		Практическая работа №3. Написание и использование функций. Отладка функций в IDE.	ПКос-4.2, ПКос-4.3	устный опрос	2
	Тема 3. Массивы, строки и работа с памятью	Лекция №3. Одномерные и двумерные массивы. Строки. Указатели и ссылки. Базовые принципы управления памятью.	ПКос-4.1	-	2
		Практическая работа №4. Обработка массивов. Работа со std::string.	ПКос-4.2, ПКос-4.3	защита практической работы	2
		Практическая работа №5. Использование указателей и ссылок. Передача массивов в функции.	ПКос-4.2, ПКос-4.3	устный опрос	2
	Тема 4. Стандартная библиотека и модульность	Лекция №4. Стандартные контейнеры. Базовые алгоритмы. Принципы модульной разработки: заголовочные файлы (.h), реализация (.cpp), компиляция нескольких файлов.	ПКос-4.1	-	2
		Практическая работа №6. Использование std::vector и std::map. Решение задач с коллекциями.	ПКос-4.2, ПКос-4.3	защита практической работы	2/2
		Практическое занятие №7. Создание многомодульной программы: разделение на .h и .cpp файлы. Компиляция из командной строки.	ПКос-4.2, ПКос-4.3	устный опрос	2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемы е компетенции	Вид контрольного о мероприятия	Кол- во часов/ из них практич еская подго товка
	Тема 5. Отладка и анализ программно го кода	Лекция №5. Типы ошибок. Работа с отладчиком. Интерпретация сообщений компилятора. Повышение читаемости кода: стиль, комментарии, именование.	ПКос-4.1	-	2
		Практическое занятие №8. Отладка программ с ошибками: поиск и устранение логических и runtime-ошибок. Работа с точками останова, просмотр переменных.	ПКос-4.2, ПКос-4.3	защита практической работы	2/2
2	Раздел 2. Объектно-ориентированное программирование на C++				
	Тема 6. Основы ООП: классы и объекты	Лекция №6. Принципы ООП. Классы и объекты. Поля, методы, конструкторы, деструкторы. Ключевые слова public/private.	ПКос-4.1	-	2
		Практическая работа №9. Создание простых классов. Использование конструкторов.	ПКос-4.2, ПКос-4.3	устный опрос	2
		Практическая работа №10. Реализация инкапсуляции: геттеры, сеттеры, проверка данных.	ПКос-4.2, ПКос-4.3	защита практической работы	2
	Тема 7. Наследовани е и полиморфиз м	Лекция №7. Наследование. Базовые и производные классы. Виртуальные функции. Полиморфизм. Абстрактные классы.	ПКос-4.1	-	2
		Практическая работа №11. Реализация иерархии.	ПКос-4.2, ПКос-4.3	устный опрос	2
		Практическая работа №12. Использование виртуальных функций и полиморфного поведения.	ПКос-4.2, ПКос-4.3	защита практической работы	2
		Практическое занятие №13. Создание и использование абстрактных классов и интерфейсов.	ПКос-4.2, ПКос-4.3	устный опрос	2
	Тема 8. Продвинуты	Лекция №8. Перегрузка операторов. Дружественные	ПКос-4.1	-	2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемы е компетенции	Вид контрольного о мероприятия	Кол- во часов/ из них практич еская подго товка
	е аспекты ООП и применение в проектах	функции. Исключения. Принципы проектирования простых приложений на C++.			
		Практическое занятие №14. Перегрузка операторов (например, +, << для класса).	ПКос-4.2, ПКос-4.3	устный опрос	2
		Практическое занятие №15. Обработка исключений (try/catch/throw).	ПКос-4.2, ПКос-4.3	защита практической работы	2
		Практическое занятие №16. Разработка небольшого ООП-проекта с использованием классов, наследования и файлового ввода-вывода.	ПКос-4.2, ПКос-4.3	защита практической работы	2
		Практическое занятие №17. Рефакторинг и улучшение читаемости кода: выделение методов, улучшение именования, документирование.	ПКос-4.2, ПКос-4.3	устный опрос	2

Таблица 5

Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
Раздел 1 Основы программирования на C++		
	Тема 1. Введение в программирование на C++	Что такое компиляция и линковка? В чём отличие компилируемых и интерпретируемых языков? Какие типы данных в C++ являются целочисленными, вещественными, логическими? Как организована структура простейшей программы на C++? Какие среды разработки (IDE) поддерживают C++? Какие команды компилятора g++ используются для сборки программы из командной строки? (ПКос-4.1)
	Тема 2. Управляющие конструкции и функции	В чём разница между циклами for, while и do-while? Когда лучше использовать каждый из них? Как передавать параметры в функцию по значению и по ссылке? В чём преимущества передачи по ссылке? Что такое область видимости переменной? Как избежать побочных эффектов в функциях? Как оформлять заголовок функции и тело? (ПКос-4.1, ПКос-4.2)
	Тема 3. Массивы, строки и работа с	В чём разница между C-строкой (char[]) и std::string? Как объявить и инициализировать одномерный и двумерный

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
	памятью	массив? Что такое указатель и как его инициализировать? Как получить адрес переменной и значение по адресу? Что такое утечка памяти и как её избежать при использовании new/delete? Почему предпочтительно использовать std::vector вместо «голых» массивов? (ПКос-4.1, ПКос-4.2)
	Тема 4. Стандартная библиотека и модульность	Какие контейнеры предоставляет стандартная библиотека C++ (vector, map, set)? В каких случаях использовать map, а не vector? Что такое заголовочный файл и зачем он нужен? Как правильно подключать пользовательские заголовочные файлы? Как избежать множественного включения (#ifndef, #pragma once)? Как компилировать программу, состоящую из нескольких .cpp-файлов? (ПКос-4.1, ПКос-4.2)
	Тема 5. Отладка и анализ программного кода	Какие виды ошибок возникают при разработке на C++? Как интерпретировать сообщения компилятора (например, «undefined reference», «segmentation fault»)? Как поставить точку останова и запустить отладку в Visual Studio Code или CLion? Что такое стек вызовов (call stack)? Какие правила оформления кода повышают его читаемость (стиль именования, комментирование, выравнивание)? (ПКос-4.1, ПКос-4.2, ПКос-4.3)
Раздел 2 Объектно-ориентированное программирование на C++		
	Тема 6. Основы ООП: классы и объекты	Что такое класс и объект? В чём смысл инкапсуляции? Какие разделы (public, private) используются в классе и для чего? Что такое конструктор и деструктор? Обязательно ли их писать? Как инициализировать поля класса? Что такое «правило трёх» (или «пяти») в C++? Почему важно разделять интерфейс (.h) и реализацию (.cpp) класса? (ПКос-4.1, ПКос-4.2)
	Тема 7. Наследование и полиморфизм	Что такое наследование и зачем оно нужно? Как объявить производный класс? Что такое виртуальная функция и как она включает полиморфизм? В чём отличие между обычной и виртуальной функцией? Что такое абстрактный класс и чисто виртуальная функция (= 0)? Можно ли создавать объекты абстрактного класса? Как использовать полиморфизм для обработки коллекции разных объектов? (ПКос-4.1, ПКос-4.2)
	Тема 8. Продвинутое ООП и применение в проектах	Зачем перегружать операторы? Какие операторы можно перегружать? Что такое дружественные функции и когда их стоит использовать? Как работают исключения в C++ (try, catch, throw)? Как обрабатывать ошибки ввода-вывода? Как спроектировать простую программу с использованием ООП (например, «Учёт книг»? Какие принципы рефакторинга помогают улучшить читаемость и поддерживаемость кода? (ПКос-4.1, ПКос-4.2, ПКос-4.3)

5. Образовательные технологии

Таблица 6

Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия	Наименование используемых активных и интерактивных образовательных технологий (форм обучения)
1.	Практическое занятие №1. Установка среды разработки. Написание первой программы "Hello, World!". Работа с компилятором (g++). Основные арифметические операции.	ПЗ Компьютерная симуляция
2.	Практическое занятие №2. Решение задач с ветвлением и циклами (нахождение НОД, сумма ряда и др.).	ПЗ Компьютерная симуляция
3.	Практическое занятие №3. Написание и использование функций. Отладка функций в IDE.	ПЗ Компьютерная симуляция
4.	Практическое занятие №4. Обработка массивов (сортировка, поиск). Работа со std::string.	ПЗ Компьютерная симуляция
5.	Практическое занятие №5. Использование указателей и ссылок. Передача массивов в функции.	ПЗ Компьютерная симуляция

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

Пример практических работ

Практическая работа №1

Тема: Написание первой программы. Работа с компилятором и базовыми операциями

Задание: Установить среду разработки (VS Code + g++ или CLion). Написать программу, выводящую «Hello, [Имя]!». Реализовать ввод имени с клавиатуры и простые арифметические вычисления (сумма, произведение двух чисел).

Вопросы для защиты:

1. Какие команды g++ используются для компиляции и запуска программы из терминала?
2. В чём разница между std::cin и std::cout?
3. Какие типы данных вы использовали? Почему выбрали именно их?
4. Что произойдёт, если ввести букву вместо числа при чтении int?

Практическая работа №2

Тема: Условные операторы и циклы

Задание: Написать программу, определяющую, является ли число простым, палиндромом или совершенным. Реализовать меню выбора задачи.

Вопросы для защиты:

1. В чём разница между if-else и switch? Когда лучше использовать каждый?
2. Почему в цикле проверки простоты достаточно идти до $\sqrt{n}$?
3. Как избежать бесконечного цикла при некорректном вводе?
4. Можно ли заменить for на while в вашей программе? Как?

Практическая работа №3

Тема: Функции и передача параметров

Задание: Вынести все логические блоки из ПР №2 в отдельные функции. Реализовать передачу параметров по значению и по ссылке (например, для обмена двух переменных).

Вопросы для защиты:

1. Зачем выделять код в функции?
2. В чём преимущество передачи по ссылке перед передачей по значению?
3. Может ли функция возвращать несколько значений? Какими способами?
4. Что такое перегрузка функций? Применимо ли это в вашей работе?

Практическая работа №4

Тема: Массивы и строки

Задание: Написать программу для обработки массива целых чисел: найти максимум, минимум, среднее, отсортировать. Работать со строками: подсчитать количество слов, перевернуть строку. Использовать `std::string`.

Вопросы для защиты:

1. В чём разница между `char[]` и `std::string`?
2. Почему в C++ массивы передаются в функции по указателю?
3. Какие алгоритмы сортировки вы знаете? Какой использовали и почему?
4. Безопасно ли использовать `std::cin >> string` для ввода с пробелами?

Практическая работа №5

Тема: Указатели и ссылки

Задание: Написать функцию, меняющую местами два элемента массива через указатели. Реализовать динамическое выделение массива (`new[]`), заполнить, обработать, освободить память.

Вопросы для защиты:

1. Что такое указатель? Как получить адрес переменной?
2. В чём разница между `*p = 5` и `p = &x`?
3. Что произойдёт, если забыть вызвать `delete[]`?
4. Почему предпочтительнее использовать `std::vector` вместо `new[]`?

Вопросы к зачету

1. Какова структура простейшей программы на C++? Что обязательно должно присутствовать?
2. Что такое компиляция? Чем отличается компилятор от интерпретатора?

3. Назовите основные типы данных в C++ и их размер (в байтах).
4. В чём разница между `int` и `double`? Когда использовать каждый?
5. Как объявить и инициализировать константу в C++? Приведите два способа.
6. Какие операторы ввода и вывода используются в C++? Приведите пример.
7. В чём разница между `if`, `if-else` и `switch`?
8. Когда предпочтительно использовать цикл `for`, а когда – `while`?
9. Что такое бесконечный цикл? Как его избежать?
10. Как вычислить факториал числа с помощью цикла?
11. Что такое функция? Зачем она нужна?
12. В чём разница между параметром и аргументом функции?
13. Как передать переменную в функцию по ссылке? Зачем это делать?
14. Что такое область видимости переменной? Приведите пример локальной и глобальной переменной.
15. Что такое массив? Как объявить одномерный массив из 10 целых чисел?
16. Можно ли изменить размер обычного массива после объявления? Почему?
17. В чём разница между `char[]` и `std::string`?
18. Как объединить две строки в C++?
19. Что такое указатель? Как получить адрес переменной?
20. Как получить значение по адресу указателя?
21. Что делает оператор `new`? Как правильно освободить память?
22. Почему после `delete` рекомендуется присваивать указателю `nullptr`?
23. Что такое утечка памяти? Как её избежать?
24. Что такое `std::vector`? Почему он безопаснее обычного массива?
25. Как добавить элемент в конец `std::vector`?
26. Что такое `std::map`? Как добавить и получить значение по ключу?
27. Зачем нужны заголовочные файлы (`.h`)?
28. Как избежать повторного включения заголовочного файла?
29. Как скомпилировать программу из двух `.cpp`-файлов с помощью `g++`?
30. Какие виды ошибок могут возникнуть при разработке на C++?
31. Что означает ошибка «segmentation fault»? Как её диагностировать?
32. Как поставить точку останова в отладчике?
33. Что такое стек вызовов (`call stack`)? Зачем он нужен при отладке?
34. Какие правила именования переменных и функций вы знаете?
35. Зачем нужны комментарии в коде? Какие виды комментариев есть в C++?
36. Что такое класс и объект? Приведите пример.
37. Что такое инкапсуляция? Как она реализуется в C++?
38. Зачем нужны разделы `public` и `private` в классе?
39. Что такое конструктор? Обязательно ли его писать?
40. Что такое деструктор? Когда он вызывается?
41. Может ли класс иметь несколько конструкторов? Как это называется?
42. Что такое наследование? Приведите пример.
43. Какие члены базового класса наследуются, а какие – нет?
44. Что такое полиморфизм? Как он достигается в C++?

45. Зачем нужен ключевое слово `virtual`?
46. Что произойдёт, если метод не объявлен как `virtual`, но вызывается через указатель на базовый класс?
47. Что такое чисто виртуальная функция? Как её объявить?
48. Можно ли создать объект абстрактного класса? Почему?
49. Зачем нужен виртуальный деструктор в базовом классе?
50. Что такое перегрузка операторов? Приведите пример.
51. Почему `operator<<` для `cout` обычно перегружается как дружественная функция?
52. Что такое исключение? Как его сгенерировать и поймать?
53. Какие стандартные классы исключений вы знаете (`std::invalid_argument`, `std::out_of_range` и др.)?
54. Почему лучше ловить исключения по константной ссылке (`const &`)?
55. Что такое «правило трёх» в C++? (Кратко)
56. Как организовать хранение нескольких объектов одного типа? Какой контейнер предпочтителен?
57. Можно ли хранить объекты разных производных классов в одном контейнере? Как?
58. Что такое рефакторинг? Приведите пример улучшения кода.
59. Как улучшить читаемость кода без изменения логики?
60. Почему важно разделять интерфейс (`.h`) и реализацию (`.cpp`) класса?

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

Текущий контроль знаний, умений и навыков проводится в форме защиты практических работ, выполняемых каждым студентом на практических занятиях. Ликвидация студентами текущих задолженностей производится также в форме выполнения индивидуальной задачи по соответствующей теме и дальнейшей ее защиты преподавателю кафедры.

Студент допускается к промежуточной аттестации (зачёту в 6 семестре), если за семестр набрал не менее 60% от максимально возможного рейтинга по итогам всех практических занятий.

Максимальная оценка за защиту одной практической работы – 10 баллов.

10 баллов – работа выполнена полностью в соответствии с требованиями; код корректен, компилируется без предупреждений, соответствует *best practices* современного C++ (RAII, безопасное управление памятью через `std::vector`/`std::string`, корректная работа с указателями и ссылками); реализованы все заявленные функции (например, ввод/вывод, обработка данных, работа с классами, наследование); соблюдена модульность (при необходимости – разделение на `.h/.cpp`); код сопровождается читаемой структурой, комментариями и/или кратким отчётом; при защите студент демонстрирует глубокое понимание использованных концепций (указатели, передача по ссылке, инкапсуляция, виртуальные функции, исключения), отвечает на все вопросы без ошибок.

9 баллов – работа выполнена полностью; возможны незначительные недочёты в оформлении кода (например, нет единого стиля именования, отсутствуют комментарии к сложным блокам) или избыточность в архитектуре (например, дублирование логики в функциях, неоптимальный выбор контейнера); при защите допущены мелкие неточности в терминологии (например, путаница между указателем и ссылкой, неточное определение конструктора или виртуальной функции), не влияющие на суть ответа.

8 баллов – работа выполнена, но содержит негрубые ошибки (например, отсутствие проверки на корректность ввода, незащищённый доступ к памяти при работе с указателями, отсутствие обработки исключений, неиспользование `const` при передаче строк); при защите студент верно объясняет общую логику решения, но допускает отдельные неточности (например, не может объяснить, зачем нужен деструктор, или путает `public` и `private`), не ведущие к искажению сути.

7 баллов – работа частично соответствует требованиям; имеются ошибки в реализации (например, программа падает при некорректном вводе, массив выходит за границы, метод класса не вызывается, наследование реализовано формально, но без полиморфизма); при защите сделаны неверные выводы (например, студент считает, что `new` не требует `delete`, или не понимает разницы между передачей по значению и по ссылке), но общее понимание темы сохранено.

6–5 баллов – работа выполнена фрагментарно; код компилируется, но не выполняет основную функцию (например, программа не обрабатывает данные, класс не хранит состояние, цикл не завершается); отсутствует обработка ошибок, пользовательского ввода или базовая логика; при защите нарушена логика объяснения, наблюдается поверхностное или искажённое понимание ключевых концепций (например, непонимание назначения функции `main`, путаница между типами данных, неумение объяснить, что такое класс или цикл).

Менее 5 баллов – работа не выполнена или содержит фундаментальные ошибки, свидетельствующие об отсутствии освоения компетенции: код не компилируется и не содержит попыток решения задачи; использован нерелевантный подход (например, попытка решить задачу через Python-синтаксис в C++); работа скопирована без понимания логики; отсутствуют базовые элементы программы (нет `#include`, `main`, переменных, операторов ввода-вывода).

Итоговая оценка учитывает результаты рейтинговой системы контроля знаний (вклад 80%), посещаемость занятий (вклад 20%) Критерии выставления оценок по системе:

0-59 % от максимального количества баллов – «незачтено»;

60 и более % – «зачтено».

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

1. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. — 5-е изд., стер. — Санкт-Петербург : Лань, 2025. — 216 с. — ISBN 978-5-507-50568-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/447392>

2. Красов, А. В. Разработка защищенного программного обеспечения : учебное пособие / А. В. Красов, А. Ю. Цветков. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2023. — 154 с. — ISBN 978-5-89160-308-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/425906>

3. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2022. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/492984>

7.2 Дополнительная литература

1. Гольдберг, Й. Нейросетевые методы в обработке естественного языка : руководство / Й. Гольдберг ; перевод с английского А. А. Слинкина. — Москва : ДМК Пресс, 2019. — 282 с. — ISBN 978-5-97060-754-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131704>

2. Ганегедара, Т. Обработка естественного языка с TensorFlow : руководство / Т. Ганегедара ; перевод с английского В. С. Яценкова. — Москва : ДМК Пресс, 2020. — 382 с. — ISBN 978-5-97060-756-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/140584>

3. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. — Красноярск : СФУ, 2019. — 288 с. — ISBN 978-5-7638-4034-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/157572>

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

1. C++ reference <https://en.cppreference.com/w/cpp>
2. Справка по C++ <https://ru.cppreference.com/w/cpp>
3. Документация по языку C++ <https://learn.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170>
4. C++ Programming Language <https://devdocs.io/cpp/>
5. Google's C++ Class. — URL: <https://developers.google.com/edu/c++>

9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Разделы 1, 2	Microsoft Visual C++ (MSVC)	расчетная, обучающая, контролирующая	Microsoft Corporation	Текущая версия
2	Разделы 1, 2	Microsoft Word	обучающая, контролирующая	Microsoft Corporation	Текущая версия
3	Разделы 1, 2	Microsoft Excel	обучающая	Microsoft Corporation	Текущая версия

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для проведения практических занятий нужен компьютерный класс с доступом в «Интернет», оснащенный программным обеспечением в соответствии с разделом 9.

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
<i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 102 ауд.)</i>	1. Экран с электроприводом 1 шт. (Инв. №558771/2) 2. Проектор 1 шт. (без инв. №) – приобретался не за счет средств вуза 3. Вандалоустойчивый шкаф 1 шт. (Инв. №558850/7) 4. Системный блок iP-4 541 3200 Mhz/1024 Mb/ 80 Gb / DVD-R с монитором 1 шт. (Инв. №558777/9) 5. Стенд «Сергеев Сергей Степанович 1910-1999» 1 шт. (Инв. №591013/25) 6. Огнетушитель порошковый 1 шт. (Инв. №559527) 7. Подвесное крепление к огнетушителю 1 шт. (Инв. № 559528) 8. Жалюзи 2шт. (Инв. №1107-221225, Инв. №1107-221225) 9. Лавка 20 шт. 10. Стол аудиторный 20 шт. 11. Стол для преподавателя 1 шт. 12. Стул 2 шт. 13. Доска маркерная 1 шт. 14. Трибуна напольная 1 шт. (без инв. №)
<i>учебная аудитория для проведения занятий семинарского типа, учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус,</i>	1. Системный блок Intel Core Intel Core i3-2100/4096Mb/500Gb/DVD-RW 10 шт. (Инв. №601997, Инв. №601998, Инв. №601999, Инв. №602000, Инв. №602001, Инв. №602002, Инв. №602003, Инв. №602004, Инв. №602005, Инв. №602006) 2. Монитор 10 шт. (без инв. №) - приобретались не за счет

302 ауд.)	средств вуза 3. Шкаф 2 шт. (Инв.№594166, Инв.№594167) 4. Тумба 1 шт. (Инв.№594168) 5. Подвесное крепление к огнетушителю 1 шт. (Инв. № 559528) 6. Огнетушитель порошковый 1 шт. (Инв. №559527) 7. Жалюзи 1 шт. (Инв.№551557) 8. Доска магнитно-маркерная 1 шт. 9. Стол 5 шт. 10. Стол компьютерный 12 шт. 11. Стул офисный 21 шт. 12. Сейф 1 шт. (без Инв.№).
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (1й учебный корпус, 212 ауд.)	Количество рабочих мест: 24 Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Структурное подразделение: Кафедра Цифровая кафедра
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (1й учебный корпус, 214 ауд.)	Количество рабочих мест: 24 Встроенные сетевые адаптеры (Intel I219-V или Realtek RTL8111H), интерфейс RJ-45, скорость 10/100/1000 Мбит/с. Точки доступа: Ubiquiti UniFi AP AC Pro, стандарты IEEE 802.11a/b/g/n/ac, частоты 2.4 ГГц (450 Мбит/с) и 5 ГГц (1300 Мбит/с), поддержка MU-MIMO, питание PoE. Структурное подразделение: Кафедра Цифровая кафедра
Студенческое общежитие	Комнаты для самоподготовки
ЦНБ имени Н.И. Железнова	Читальный зал

11. Методические рекомендации обучающимся по освоению дисциплины

Предполагается, что студент выполняет практическое задание в аудитории, дома оформляет и готовится по теоретическим вопросам к защите отчета на следующем занятии.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятие, обязан предъявить преподавателю документы установленного образца, подтверждающие необходимость пропуска. Не допускается пропуск занятий без уважительной причины.

Студент, пропустивший занятия, осваивает материал самостоятельно (выполняет практическое задание по своему варианту в компьютерном классе кафедры в часы, свободные от занятий, изучает теоретические вопросы).

Студент, пропустивший лекцию, отвечает на вопросы по пропущенной теме.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

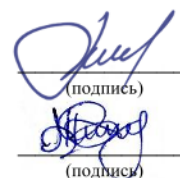
На первом занятии преподаватель закрепляет за каждым студентом номер варианта для выполнения индивидуальных работ (как правило, номер варианта соответствует порядковому номеру студента в журнале преподавателя). По каждой индивидуальной работе должна быть поставлена оценка по факту ее защиты. Защиту рекомендуется проводить на следующем после получения задания занятии. Преподаватель обязан проверить соответствие выполненного задания исходным данным варианта студента. Таким образом, исключается вероятность плагиата.

Преподаватель должен стимулировать студентов к занятию научно-исследовательской работой, изучению научной литературы по теме искусственного интеллекта, в т.ч. отечественной и зарубежной периодики.

Программу разработали:

Демичев В.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)

Титов А.Д., ассистент
(ФИО, ученая степень, ученое звание)



(подпись)
(подпись)

РЕЦЕНЗИЯ
на рабочую программу дисциплины
Б1.В.12 «Программирование на языке С++»
ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии»,
направленность «Фуллстек разработка»
(квалификация выпускника – бакалавр)

Кийко Павлом Владимировичем, доцентом кафедры высшей математики, кандидатом педагогических наук (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Компьютерная графика» ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленность «Фуллстек разработка» (бакалавриат) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре статистики и кибернетики (разработчики – Демичев Вадим Владимирович, доцент, кандидат экономических наук, Титов Артем Денисович, ассистент кафедры статистики и кибернетики).

Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины « Программирование на языке С++» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.02 «Информационные системы и технологии». 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 к части, формируемой участниками образовательных отношений учебного цикла – Б1.В

3. Представленные в Программе цели дисциплины соответствуют требованиям ФГОС ВО направления 09.03.02 «Информационные системы и технологии».

4. В соответствии с Программой за дисциплиной «Компьютерная графика» закреплена 1 профессиональная компетенция, определяемая самостоятельно (3 индикатора). Дисциплина «Компьютерная графика» 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 «Программирование на языке С++» составляет 4 зачётные единицы (144 часа/из них практическая подготовка 4 часа).

6. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Компьютерная графика» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 «Информационные системы и технологии» 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 «Программирование на языке С++» предполагает 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 09.03.02 «Информационные системы и технологии».

10. Представленные и описанные в Программе формы текущей оценки знаний (опрос, в форме обсуждения отдельных вопросов), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме зачета в 6 семестре, что соответствует статусу дисциплины,

части, формируемой участниками образовательных отношений учебного цикла – Б1.В ФГОС ВО направления 09.03.02 «Информационные системы и технологии».

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

12. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 3 источника (базовый учебник), дополнительной литературой – 3 наименования, Интернет-ресурсы – 5 источников и соответствует требованиям ФГОС ВО направления 09.03.02 «Информационные системы и технологии».

13. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Программирование на языке С++» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

14. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Программирование на языке С++».

ОБЩИЕ ВЫВОДЫ

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Программирование на языке С++» ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленность **«Фуллстек разработка»** (квалификация выпускника – бакалавр), разработанная Демичевым Вадимом Владимировичем, доцентом, кандидатом экономических наук, Титовым Артемом Денисовичем, ассистентом кафедры статистики и кибернетики, соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Кийко Павел Владимирович, доцент кафедры высшей математики, кандидат педагогических наук ФГБОУ ВО «Российский государственный аграрный университет– МСХА имени К.А. Тимирязева»



(подпись)

«26» августа 2025 г.