

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: Директор института экономики и управления АПК

Дата подписания: 2025-08-26 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.15 Компьютерная графика

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленность: Фуллстек разработка

Курс 3

Семестр 5

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.

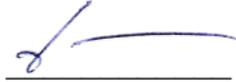
И.о. зав. кафедрой Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(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.....	6
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ	6
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	6
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	7
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ	7
ПО СЕМЕСТРАМ	7
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	11
4.3 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	12
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	15
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	19
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	20
7.1 Основная литература	20
7.2 Дополнительная литература.....	21
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	20
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	21
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....	22
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	23
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	24

Аннотация
рабочей программы учебной дисциплины
Б1.В.15 Компьютерная графика
для подготовки бакалавров по направлению 09.03.02 «Информационные
системы и технологии» по направленности «Фуллстек разработка»

Цель освоения дисциплины: Целями освоения дисциплины «Компьютерная графика» являются: освоение математических основ, основных понятий и алгоритмов компьютерной графики; получение практических навыков разработки программного обеспечения с использованием современных графических библиотек; развитие умений, позволяющих применять полученные теоретические и практические навыки для решения задач компьютерной графики, возникающих в научно-исследовательской и профессиональной деятельности.

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

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:

История развития компьютерной графики. Области применения компьютерной графики. Примеры использования. Структура программы. Графический конвейер. Аппаратные средства компьютерной графики.

Основные понятия растровой и векторной графики. Достоинства и недостатки разных способов представления изображений. Параметры растровых изображений: разрешение, глубина цвета. Свет и цвет: физические основы. Восприятие светового потока глазом человека. Характеристики цвета: яркость, светлость, тон, насыщенность. Основные цветовые модели и цветовые пространства.

Системы координат. Аффинные преобразования плоскости: масштабирование, отражение, поворот, перенос, сдвиг. Матрицы преобразований. Однородные координаты. Правосторонняя и левосторонняя системы координат. Плоскости и прямые. Аффинные преобразования пространства. Кватернионы. Однородные координаты. Поверхности. Триангуляция: диаграмма Вороного и алгоритм Делоне. Проектирование. Виды проекций. Ортогональное и перспективное проектирование. Получение проекций с помощью матричных преобразований.

Понятие растеризации. Связность пикселей. Растровое представление отрезка. Алгоритм Брезенхейма. Растровое представление окружности. Растровое представление кривых на плоскости. Кривые Безье первого, второго и третьего порядков. Многочлены Бернштейна. Интерполяция. Сплайны. Отсечение многоугольников. Этапы отображения трехмерных объектов. Отсечение. Полигональные сетки. Определение видимости. Алгоритм плавающего горизонта. Алгоритм Робертса. Метод Z-буфера. Трассировка лучей. Алгоритм художника. Алгоритм Варнока. Алгоритм Вейлера-Азертона. Методы упорядочивания. Ис-

точники света. Диффузное отражение и рассеянный свет. Зеркальное отражение. Методы закраски сплошных объектов: однотонная закраска, метод Гуро. метод Фонга. Тени и их виды. Алгоритмы затенения. Свето пропускающие поверхности. Текстуры.

Эффекты. Виды фильтров. Понятие линейного фильтра. Сглаживающие фильтры: фильтр Гаусса. Расширение динамического диапазона. Цветовая коррекция. Нелинейные фильтры.

История создания. Основы работы: создание контекста, вывод примитивов, режимы. Полигональные модели. Графический конвейер. Преобразование вершин. Работа с матрицами. Шейдеры и язык GLSL. Освещение. Работа с текстурами. Буфер глубины. Смешивание цветов и прозрачность. Тени. Постобработка и фильтры. Анимация.

Задача визуализации данных. Способы представления данных. Примеры. Алгоритм визуализации научных данных. Библиотеки визуализации. Проблемы существующих систем визуализации.

Общая трудоемкость дисциплины составляет: 3 зачетные единицы (108 часов, в том числе практической подготовки 4 часа).

Промежуточный контроль: зачет.

1. Цель освоения дисциплины

Целями освоения дисциплины «Компьютерная графика» являются: освоение математических основ, основных понятий и алгоритмов компьютерной графики; получение практических навыков разработки программного обеспечения с использованием современных графических библиотек; развитие умений, позволяющих применять полученные теоретические и практические навыки для решения задач компьютерной графики, возникающих в научно-исследовательской и профессиональной деятельности.

2. Место дисциплины в учебном процессе

Дисциплина «Компьютерная графика» 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 «Компьютерная графика» являются «Линейная алгебра», «Математический анализ», «Математическая статистика», «Теория вероятностей», «Информационные технологии», «Алгоритмизация и программирование».

Дисциплина «Компьютерная графика» является основополагающей для изучения следующих дисциплин: «Инструментальные средства информационных систем», «Администрирование информационных систем», «Управление IT-проектами» и др.

Особенностью дисциплины является ориентация на формирование инженерных компетенций в области разработки, отладки и интеграции графических модулей в составе полноценных информационных систем, с акцентом на практическое применение алгоритмов компьютерной графики, использование современных Python-библиотек и веб-технологий, а также отработку навыков рефакторинга, модульного проектирования и взаимодействия графического компонента с внешней средой (пользовательский интерфейс, API, файловые и сетевые ресурсы).

Рабочая программа дисциплины «Компьютерная графика» 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

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций, представленных в таблице 1.

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 3 зач.ед. (108 часов, в том числе практической подготовки 4 часа), их распределение по видам работ семестрам представлено в таблице 2.

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	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	основные понятия компьютерной графики; классические алгоритмы 2D/3D-графики: Брезенхема, заливки, отсечения, Z-буфера, текстурирования; принципы работы графических API; архитектуру типичного графического приложения;	-	-
			ПКос-4.2 Уметь: использовать методы и приемы формализации и алгоритмизации поставленных задач; исполь-	-	формализовать задачу визуализации в виде алгоритма; выбирать и применять	-

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
			<p>зовать 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</p>		<p>подходящие алгоритмы компьютерной графики для решения поставленной задачи; использовать графические библиотеки Python для реализации 2D/3D-сцен; интегрировать графический модуль в полноценное приложение;</p>	
			<p>ПКос-4.3 Владеть навыками; составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания; разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных</p>	-	-	<p>навыками разработки читаемого, модульного и документированного кода графических приложений на Python; навыками отладки на уровне как отдельных функций, так и взаимодействия с внешней средой; навыками рефакторин-</p>

№ п/п	Код компе- тенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
			документов; создания про- граммного кода в соответ- ствии с техническим задани- ем (готовыми спецификация- ми); оптимизации программ- ного кода с использованием специализированных про- граммных средств; анализа и проверки исходного про- граммного кода; отладки про- граммного кода на уровне программных модулей и межмодульных взаимодей- ствий и взаимодействий с окружением			га: выделение графиче- ского ядра в отдельный модуль, параметриза- ция сцены, управление ресурсами; навыками создания технической докумен- тации по графическому компоненту (описание интерфейса, форматов данных, зависимостей).

Таблица 2

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость	
	час. всего/*	В т.ч. по семестрам № 6
Общая трудоёмкость дисциплины по учебному плану	108/4	108/4
1. Контактная работа:	68,25/4	68,25/4
Аудиторная работа	68,25/4	68,25/4
<i>Лекции (Л)</i>	34	34
<i>практические занятия (ПЗ)</i>	34/4	34/4
<i>контактная работа на промежуточном контроле (КРА)</i>	0,25	0,25
2. Самостоятельная работа (СРС)	39,75	39,75
<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>	30,75	30,75
<i>Подготовка к зачету</i>	9	9
Вид промежуточного контроля:	Зачет	

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 компьютерной графики и инструментарий	18,3	6	6		6,3
Раздел 2. Алгоритмы 2D-графики и их реализация	18,45/2	6	6/2		6,45
Раздел 3. Введение в 3D-графику и интерактивную визуализацию	24/2	8	8/2		8
Раздел 4. Отладка, оптимизация и интеграция графических модулей	47	14	14		19
Контактная работа на промежуточном контроле (КРА)	0,25			0,25	
Итого по дисциплине	108/4	34	34/4	0,25	39,75

Раздел 1. Основы компьютерной графики и инструментарий

Вводный раздел, знакомящий с ключевыми понятиями графики (растровая/векторная, цветовые модели, координаты) и современным Python-стеком для визуализации (Pillow, Matplotlib, PyGame). Акцент на интеграцию графики в веб- и десктоп-приложения через REST API и фреймворки.

Раздел 2. Алгоритмы 2D-графики и их реализация

Рассматриваются классические алгоритмы (Брезенхема, заливка, аффинные преобразования) и их практическая реализация на Python. Студенты учатся

не только применять библиотеки, но и понимать «под капотом», а также генерировать отчёты и диаграммы в векторных форматах (SVG, PDF) для интеграции в ИС.

Раздел 3. Введение в 3D-графику и интерактивную визуализацию

Студенты осваивают основы 3D-рендеринга: проекции, камеры, освещение, текстурирование. Практика строится вокруг WebGL/Three.js (в вебе) и moderngl (в Python), что позволяет создавать интерактивные 3D-сцены и интегрировать их в полноценные приложения без глубокого погружения в C++/OpenGL.

Раздел 4. Отладка, оптимизация и интеграция графических модулей

Фокус смещается с «как нарисовать» на «как сделать надёжно и эффективно». Охватываются отладка, профилирование, рефакторинг, тестирование и взаимодействие с внешней средой (БД, API, файлы).

4.3 Лекционные/Практические занятия

Таблица 4

Содержание лекционных/практических занятий и контрольные мероприятия

№ п/п	№ раздела	№ и название практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
1	Раздел 1. Основы компьютерной графики и инструментарий для фуллстек-разработчиков	Лекция № 1. Введение в графику: задачи, модели цвета, координаты	ПКос-4.1	-	2
		Практическая работа № 1. Работа с изображениями (Pillow, NumPy)	ПКос-4.2, ПКос-4.3	Защита практической работы	2
		Лекция № 2. Графические библиотеки Python и архитектура приложений	ПКос-4.1	-	2
		Практическая работа № 2. Интерактивное 2D-окно на PyGame	ПКос-4.2, ПКос-4.3	Защита практической работы	2
		Лекция № 3. Модульное проектирование и интеграция в ИС	ПКос-4.1	-	2
2	Раздел 2. Алгоритмы 2D-графики и их реализация	Лекция № 4. Классические алгоритмы растровой графики	ПКос-4.1	Защита практической работы	
		Практическая работа № 4. Реализация алгоритма Брезенхема на Python	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 5. Аффинные преобразования и матрицы	ПКос-4.1	Защита практической работы	2
		Практическая работа № 5. Применение трансформаций к изображениям	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 6. Векторная графика, шрифты, генера-	ПКос-4.1	Защита практической работы	2

№ п/п	№ раздела	№ и название практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ция отчётов			
		Практическая работа № 6. Генерация PDF/SVG-отчёта и интеграция в API	ПКос-4.2, ПКос-4.3	-	2
3	Раздел 3. Введение в 3D-графику и интерактивную визуализацию	Лекция № 7. Основы 3D: проекции, камера, конвейер	ПКос-4.1	Защита практической работы	2
		Практическая работа № 7. Визуализация 3D-данных (Matplotlib, plotly)	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 8. WebGL и Three.js в фуллстек-приложениях	ПКос-4.1	Защита практической работы	2
		Практическая работа № 8. Веб-приложение с 3D-сценой (Flask + Three.js)	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 9. Текстурирование, материалы, форматы моделей	ПКос-4.1	Защита практической работы	2
		Практическая работа № 9. Загрузка и отображение glTF-модели	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 10. GPU-ускорение: moderngl, шейдеры, GLSL	ПКос-4.1	Защита практической работы	2
		Практическая работа № 10. Написание и запуск шейдера через moderngl	ПКос-4.2, ПКос-4.3	-	2
4	Раздел 4. Отладка, оптимизация и интеграция графических модулей	Лекция № 11. Методы отладки графических приложений	ПКос-4.1	Защита практической работы	2
		Практическая работа № 11. Отладка сломанного графического кода	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 12. Оптимизация и профилирование графики	ПКос-4.1	Защита практической работы	2
		Практическая работа № 12. Профилирование и оптимизация 3D-приложения	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 13. Рефакторинг: модули, документация, типизация	ПКос-4.1	Защита практической работы	2
		Практическая работа № 13. Рефакторинг монолитного скрипта	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 14. Взаимодействие с внешней средой (БД, API)	ПКос-4.1	Защита практической работы	2
		Практическая работа № 14. Визуализация данных из	ПКос-4.2, ПКос-4.3	-	2

№ п/п	№ раздела	№ и название практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		SQLite			
		Лекция № 15. Динамическая графика: Streamlit, Dash	ПКос-4.1	Защита практической работы	2
		Практическая работа № 15. Интерактивный дашборд на Streamlit	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 16. Тестирование графических приложений	ПКос-4.1	Защита практической работы	2
		Практическая работа № 16. Написание unit- и snapshot-тестов	ПКос-4.2, ПКос-4.3	-	2
		Лекция № 17. Итоговая архитектура графического компонента	ПКос-4.1	Защита практической работы	2
		Практическая работа № 17. Полноценное фуллстек-приложение с графикой, БД, API, документацией	ПКос-4.1, ПКос-4.2, ПКос-4.3	-	2

Таблица 5

Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
1.	Раздел 1. Основы компьютерной графики и инструментарий для фуллстек-разработчиков	Основные понятия компьютерной графики: растровая и векторная графика, цветовые модели (RGB, HSV), координатные системы. Архитектура графического приложения: цикл рендеринга, обработка событий. Обзор Python-библиотек для графики (Pillow, Matplotlib, PyGame). Интеграция графики в веб-приложения через REST API. ПКос-4 (ПКос-4.1; ПКос-4.2; ПКос-4.3)
2.	Раздел 2. Алгоритмы 2D-графики и их реализация	Алгоритм Брезенхема для рисования линий и окружностей. Алгоритмы заливки и отсечения. Аффинные преобразования: матрицы масштабирования, поворота, сдвига. Композиция трансформаций. Генерация векторной графики (SVG, PDF) и работа с шрифтами. ПКос-4 (ПКос-4.1; ПКос-4.2; ПКос-4.3)
3.	Раздел 3. Введение в 3D-графику и интерактивную визуализацию	Основы 3D-графики: проекции (ортогональная, перспективная), камера, сцена, объекты. Графический конвейер. Форматы 3D-моделей (OBJ, glTF). Освещение, материалы, текстурирование. Введение в WebGL и Three.js. Основы шейдеров и язык GLSL. ПКос-4 (ПКос-4.1; ПКос-4.2; ПКос-4.3)
4.	Раздел 4. Отладка, оптимизация и интеграция графических модулей	Методы отладки графических приложений: визуальная диагностика, логирование, анализ ошибок шейдеров. Профилирование производительности (cProfile, browser devtools). Оптимизация рендеринга: минимизация перерисовки, кэширование, GPU-ускорение. Рефакторинг графического кода: модульность, типизация, документирование. Тестирование

№ п/п	№ раздела и темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		графики: unit-тесты, snapshot-тесты. Взаимодействие с внешней средой: работа с файлами, базами данных, сетевыми API. ПКос-4 (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
1	Практическая работа № 3. Разработка REST API на Flask, возвращающего сгенерированное изображение	ПЗ
2	Лекция № 2. Графические библиотеки в экосистеме Python: Pillow, Matplotlib, PyGame, moderngl, Three.js	Л
3	Лекция № 10. GPU-ускорение: moderngl, OpenGL, шейдеры, язык GLSL	Л

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. Работа с изображениями в Python: загрузка, обработка и визуализация

Цель: Освоить базовые операции с растровой графикой с использованием Python-библиотек.

Задача:

Написать скрипт на Python, который:

1. Загружает изображение (JPEG/PNG) с диска,
2. Применяет к нему три операции: изменение размера, поворот на 45°, преобразование в оттенки серого,
3. Визуализирует исходное и обработанное изображения в одном окне с помощью Matplotlib,
4. Сохраняет результат в новый файл.

Требования: Использовать только Pillow и Matplotlib. Код должен быть документирован.

Результат: Рабочий Python-скрипт + изображения до/после обработки.

Вопросы для защиты:

1. Какие цветовые модели поддерживает Pillow? В каком формате хранятся пиксели?
2. Почему при повороте изображения возникают чёрные углы? Как это исправить?
3. Можно ли выполнить те же операции с помощью OpenCV? В чём разница в подходе?

Практическая работа №2. Создание интерактивного 2D-окна с PyGame

Цель: Освоить базовую архитектуру графического приложения с циклом рендеринга и обработкой событий.

Задача:

Разработать приложение на PyGame, в котором:

1. Рисуются круг,
2. При нажатии клавиш WASD круг перемещается,
3. При нажатии пробела — меняет цвет,
4. При закрытии окна — программа корректно завершается.
5. Требования: Использовать `pygame.display`, `pygame.event`, `pygame.draw`.

Результат: Рабочая программа с исходным кодом.

Вопросы для защиты:

1. Что такое игровой цикл? Почему он должен быть бесконечным?
2. Как избежать «зависания» приложения при высокой нагрузке в цикле?
3. Почему важно вызывать `pygame.quit()` при завершении?

Практическая работа №3. Разработка REST API для генерации изображений

Цель: Научиться интегрировать графику в веб-приложения через API.

Задача:

Создать Flask-приложение с одним endpoint `/plot`, который:

1. Принимает GET-запрос с параметрами (например, `?x=1,2,3&y=4,5,6`),
2. Генерирует график (линейный) с помощью Matplotlib,
3. Возвращает его как PNG-изображение (без сохранения на диск).

Требования: Использовать `io.BytesIO` для in-memory обработки.

Результат: Рабочий Flask-сервер + скриншот результата в браузере.

Вопросы для защиты:

1. Зачем нужен BytesIO? Почему нельзя просто вернуть файл с диска?
2. Как обрабатывать ошибки (например, несовпадение длины `x` и `y`)?
3. Как масштабировать такое API под высокую нагрузку?

Практическая работа №4. Реализация алгоритма Брезенхема на чистом Python

Цель: Понять принципы растеризации и научиться реализовывать графические алгоритмы без библиотек.

Задача:

Написать функцию `draw_line(x0, y0, x1, y1, image)`, которая рисует линию на массиве NumPy (имитация изображения) с использованием алгоритма Брезенхема. Визуализировать результат через Matplotlib.

Требования: Без использования Pillow/OpenCV для рисования.

Результат: Скрипт + сравнение с `plt.plot()`.

Вопросы для защиты:

1. Почему алгоритм Брезенхема использует только целочисленную арифметику?
2. Как модифицировать алгоритм для рисования окружности?
3. В каких современных системах он всё ещё актуален?

Практическая работа №5. Применение аффинных преобразований к изображениям

Цель: Освоить матричные преобразования в компьютерной графике.

Задача:

С помощью NumPy реализовать функции:

`scale(image, sx, sy)`,

`rotate(image, angle)`,

`shear(image, shx, shy)`.

Применить их к изображению и визуализировать цепочку: оригинал → масштаб → поворот → сдвиг.

Требования: Не использовать OpenCV.

Результат: Скрипт + изображение-результат.

Вопросы для защиты:

1. Почему порядок трансформаций важен?
2. Как избежать потери пикселей при повороте?
3. Как представляется точка в однородных координатах?

Вопросы к зачету

1. В чём разница между растровой и векторной графикой? Приведите примеры форматов.
2. Какие цветовые модели вы знаете? Чем RGB отличается от HSV?
3. Что такое координатная система в компьютерной графике? Чем экранная отличается от мировой?
4. Как устроен цикл рендеринга в интерактивном графическом приложении?
5. Какие Python-библиотеки используются для работы с изображениями? Назовите их назначение.
6. Как загрузить, изменить и сохранить изображение с помощью Pillow?
7. Как визуализировать массив данных как изображение в Matplotlib?
8. Что такое REST API для генерации изображений? Приведите пример использования.
9. Как передать изображение из Flask-приложения в браузер без сохранения на диск?
10. Зачем нужен BytesIO при работе с графикой в вебе?
11. Какие типы событий обрабатываются в PyGame?
12. Почему важно вызывать `pygame.quit()` при завершении?

13. Как интегрировать графический модуль в полноценную информационную систему?
14. Какие требования предъявляются к графическому компоненту в составе ИС?
15. Какие ошибки типичны при работе с путями к изображениям в Python?
16. В чём суть алгоритма Брезенхема? Почему он использует только целочисленную арифметику?
17. Как модифицировать алгоритм Брезенхема для рисования окружности?
18. Что такое аффинное преобразование? Приведите примеры.
19. Как с помощью матриц выполнить поворот изображения?
20. Почему порядок применения трансформаций важен?
21. Что такое однородные координаты и зачем они нужны?
22. Как реализовать масштабирование изображения без искажений?
23. Что такое алгоритм заливки? Как он работает?
24. Какие методы отсечения вы знаете? Объясните Cohen–Sutherland.
25. Как генерировать график в формате SVG из Python?
26. В чём преимущество векторной графики при масштабировании?
27. Как добавить текст на изображение с помощью Pillow?
28. Как работать с шрифтами в ReportLab?
29. Как встроить изображение в PDF-отчёт?
30. Почему при повороте изображения появляются чёрные области? Как это исправить?
31. В чём разница между ортогональной и перспективной проекцией?
32. Что такое графический конвейер? Из каких этапов он состоит?
33. Какие компоненты входят в 3D-сцену (камера, источник света, объект)?
34. Как визуализировать 3D-точки в Matplotlib?
35. Чем plotly отличается от Matplotlib в контексте 3D?
36. Что такое WebGL и зачем он нужен в вебе?
37. Как подключить Three.js к Flask-приложению?
38. Какие форматы 3D-моделей поддерживаются в Three.js?
39. Чем glTF лучше OBJ?
40. Как загрузить glTF-модель в Three.js?
41. Что такое материал и текстура в 3D?
42. Как настроить освещение в Three.js?
43. Что такое GPU-ускорение и зачем оно в графике?
44. Какие библиотеки Python позволяют работать с OpenGL?
45. Что такое шейдер? Какие типы шейдеров вы знаете?
46. Какие ошибки типичны при разработке графических приложений?
47. Как отладить «сломанный» PyGame-цикл?
48. Как профилировать производительность веб-приложения с 3D-графикой?
49. Что такое «draw call» и почему его нужно минимизировать?
50. Как ограничить частоту кадров (FPS) в Three.js?
51. Что такое рефакторинг графического кода? Зачем он нужен?
52. Как выделить графическое ядро в отдельный модуль?
53. Зачем нужна типизация в графических приложениях?
54. Как работает snapshot-тестирование для графики?

55. Как написать unit-тест для функции поворота точки?
56. Как интегрировать графический модуль с SQLite?
57. Как защитить графический API от некорректных входных данных?
58. Какие инструменты используются для отладки шейдеров?
59. Как масштабировать графическое приложение под высокую нагрузку?
60. Как документировать графический компонент в составе ИС?

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 5 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; графический компонент функционален, код корректен, модулен и соответствует best practices (RAII, управление ресурсами, обработка ошибок); реализованы все заявленные функции (например, API, анимация, интеграция с БД); код сопровождается пояснительным комментарием или отчётом; при защите студент демонстрирует глубокое понимание использованных технологий (Pillow, Flask, Three.js, OpenGL, moderngl и др.), отвечает на все вопросы без ошибок.

9 баллов – работа выполнена полностью; возможны незначительные недочёты в оформлении отчёта, стиле кода или избыточность в архитектуре (например, дублирование логики); при защите допущены мелкие неточности в терминологии (например, путаница между vertex и fragment шейдером, неточное определение аффинного преобразования), не влияющие на суть ответа.

8 баллов – работа выполнена, но содержит негрубые ошибки (например, отсутствие обработки исключений при загрузке изображения, неоптимальная перерисовка сцены, избыточное копирование данных в GPU); при защите студент верно объясняет логику решения, но допускает отдельные неточности (например, не может объяснить, зачем нужен Z-буфер или как работает REST API), не ведущие к искажению сути.

7 баллов – работа частично соответствует требованиям; имеются ошибки в реализации (например, график не обновляется при изменении параметров, 3D-сцена не отображается из-за неправильной проекции, API возвращает ошибку 500); при защите сделаны неверные выводы (например, студент считает, что Pillow работает на GPU, или не понимает разницы между растровой и векторной графикой), но общее понимание темы сохранено.

6–5 баллов – работа выполнена фрагментарно; графический компонент не запускается или не выполняет основную функцию; код не взаимодействует с

внешней средой (нет связи с БД, API не отдаёт данные); отсутствует обработка пользовательского ввода или ошибок; при защите нарушена логика объяснения, наблюдается поверхностное или искажённое понимание ключевых концепций (например, непонимание цикла рендеринга, путаница между Flask и Streamlit, неумение объяснить, что такое шейдер или REST).

Менее 5 баллов – работа не выполнена или содержит фундаментальные ошибки, свидетельствующие об отсутствии освоения компетенции (например, отсутствует графический вывод, использован нерелевантный инструментарий вроде TensorFlow вместо Pillow, работа скопирована без понимания логики, код не компилируется и не содержит попыток решения задачи).

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. Боресков, А. В. Основы компьютерной графики : учебник и практикум для вузов / А. В. Боресков, Е. В. Шикин. — Москва : Издательство Юрайт, 2025. — 219 с. — (Высшее образование). — ISBN 978-5-534-13196-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560176>
2. Колошкина, И. Е. Компьютерная графика : учебник и практикум для среднего профессионального образования / И. Е. Колошкина, В. А. Селезнев, С. А. Дмитроченко. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 237 с. — (Профессиональное образование). — ISBN 978-5-534-17739-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562117>
3. Колошкина, И. Е. Компьютерная графика : учебник и практикум для вузов / И. Е. Колошкина, В. А. Селезнев, С. А. Дмитроченко. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 237 с. — (Высшее образование). — ISBN 978-5-534-17757-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561854>
4. Боресков, А. В. Компьютерная графика : учебник и практикум для среднего профессионального образования / А. В. Боресков, Е. В. Шикин. — Москва : Издательство Юрайт, 2025. — 219 с. — (Профессиональное образование). — ISBN 978-5-534-11630-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566514>

7.2 Дополнительная литература

1. Васильева, В. А. Инженерная и компьютерная графика в садоводстве : учебник и практикум для вузов / В. А. Васильева. — Москва : Издательство Юрайт, 2025. — 182 с. — (Высшее образование). — ISBN 978-5-534-18242-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568663>
2. Хейфец, А. Л. Компьютерная графика для строителей : учебник для вузов / А. Л. Хейфец, В. Н. Васильева, И. В. Буторина ; под редакцией А. Л. Хейфеца. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 255 с. — (Высшее образование). — ISBN 978-5-534-19652-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/556850>.
3. Инженерная 3D-компьютерная графика : учебник и практикум для среднего профессионального образования / А. Л. Хейфец, А. Н. Логиновский, И. В. Буторина, В. Н. Васильева. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 596 с. — (Профессиональное образование). — ISBN 978-5-534-20468-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/558194>.

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

1. Язык программирования Python. URL: <https://www.python.org/> (открытый доступ)
2. Анаконда. URL: <https://www.anaconda.com/distribution/> (открытый доступ)

9. Перечень программного обеспечения

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Разделы 1–4	Python	Интерпретатор, язык программирования	Python Software Foundation	Текущая версия (3.10+)
2	Разделы 1–4	Pillow (PIL Fork)	Библиотека обработки изображений	Alex Clark et al.	Текущая версия
3	Разделы 1–2, 4	Matplotlib	Библиотека визуализации данных	John D. Hunter et al.	Текущая версия
4	Раздел 1	PyGame	Библиотека для 2D-графики и игр	Pete Shinnery et al.	Текущая версия
5	Разделы 1, 3–4	Flask	Веб-фреймворк (microframework)	Armin Ronacher	Текущая версия
6	Раздел 4	Streamlit	Фреймворк для интерактивных дашбордов	Streamlit Inc.	Текущая версия
7	Раздел 3	Three.js	JavaScript-библиотека 3D-графики	Mr.doob (Ricardo Cabello)	Текущая версия
8	Раздел 3	moderngl	Python-обёртка для	Szabolcs	Текущая

			OpenGL	Dombi	версия
9	Раздел 4	SQLite	Встраиваемая СУБД	D. Richard Hipp	Текущая версия
10	Разделы 1–4	NumPy	Библиотека численных вычислений	NumPy Developers	Текущая версия
11	Раздел 2	ReportLab	Библиотека генерации PDF	Andy Robinson et al.	Текущая версия
12	Раздел 3	Plotly	Библиотека интерактивной визуализации	Plotly Technologies Inc.	Текущая версия
13	Разделы 1–4	Visual Studio Code	Интегрированная среда разработки (IDE)	Microsoft	Текущая версия
14	Разделы 1–4	Jupyter Notebook / Lab	Среда для интерактивного программирования и визуализации	Project Jupyter	Текущая версия
15	Раздел 4	pytest	Фреймворк для тестирования Python-кода	Holger Krekel et al.	Текущая версия

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

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>	<ol style="list-style-type: none"> 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, 302 ауд.)</i>	<ol style="list-style-type: none"> 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 средств вуза 3. Шкаф 2 шт. (Инв.№594166, Инв.№594167) 4. Тумба 1 шт. (Инв.№594168) 5. Подвесное крепление к огнетушителю 1 шт. (Инв. №

	<p>559528)</p> <p>6. Огнетушитель порошковый 1 шт. (Инв. №559527)</p> <p>7. Жалюзи 1 шт. (Инв.№551557)</p> <p>8. Доска магнитно-маркерная 1 шт.</p> <p>9. Стол 5 шт.</p> <p>10. Стол компьютерный 12 шт.</p> <p>11. Стул офисный 21 шт.</p> <p>12. Сейф 1 шт. (без Инв.№).</p>
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.)	<p>Количество рабочих мест: 24</p> <p>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.</p> <p>Структурное подразделение: Кафедра Цифровая кафедра</p>
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.)	<p>Количество рабочих мест: 24</p> <p>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.</p> <p>Структурное подразделение: Кафедра Цифровая кафедра</p>
Студенческое общежитие	Комнаты для самоподготовки
ЦНБ имени Н.И. Железнова	Читальный зал

11. Методические рекомендации студентам по освоению дисциплины

Приступая к изучению дисциплины «Компьютерная графика», студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке РГАУ-МСХА им. К.А. Тимирязева, получить в библиотеке рекомендованные учебники и учебно-методические пособия, завести новую тетрадь для работы с первоисточниками.

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

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятия обязан самостоятельно выполнить сообщение (презентацию), рассмотренную на практическом занятии и подготовиться по контрольным вопросам к защите работы в рамках часов консультаций.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

Комплексное освоение студентами учебной дисциплины «Компьютерная графика» предполагает изучение рекомендуемой учебно-методической литературы, подготовку к практическим занятиям, самостоятельную работу при выполнении практических заданий, домашних заданий.

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.

В рамках курса предусмотрены формы работы студентами в малых группах, ориентированные на развитие навыков взаимодействия у студентов при решении профессиональных задач.

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

Студент может провести собственное статистическое наблюдение за социально-экономическими явлениями, представляющими его научный интерес, построить статистическую модель, сделать прогноз. В случае надлежащего качества, его работа может быть заслушана на научном кружке кафедры или на студенческой научной конференции. По решению кафедры, студенты, занявшие призовые места на научных студенческих конференциях, могут освобождаться от сдачи зачета по дисциплине.

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.15 «Компьютерная графика»
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 «Компьютерная графика» (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 «Компьютерная графика» составляет 3 зачётные единицы (108 часов/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 «Компьютерная графика» 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 5 семестре, что соответствует статусу дисциплины, ча-

сти, 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 – 4 источника (базовый учебник), дополнительной литературой – 3 наименования, Интернет-ресурсы – 2 источник и соответствует требованиям ФГОС ВО направления *09.03.02 «Информационные системы и технологии»*.

13. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Компьютерная графика» 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 «Компьютерная графика».

ОБЩИЕ ВЫВОДЫ

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Компьютерная графика» 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.