

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: Директор института экономики и управления АПК

Дата подписания: 27.08.2025 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



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.В.13 Разработка пользовательских интерфейсов**

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленность:

«Фуллстек разработка»,

«Системная аналитика и разработка программного обеспечения»

Курс 4

Семестр 7

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.

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)

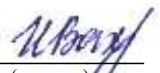

(подпись)

«26» августа 2025 г.

Рецензент:

Вахрушева И.А., 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

(ФИО, ученая степень, ученое звание)


(подпись)

протокол №1 «28» августа 2025 г.

И. о. зав. выпускающей кафедрой
статистики и кибернетики

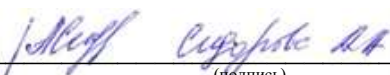
Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)


(подпись)

«28» 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.....	6
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	12
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ	12
ПО СЕМЕСТРАМ	12
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	12
4.3 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	14
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	18
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	18
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ	18
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	26
7.1 ОСНОВНАЯ ЛИТЕРАТУРА	26
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
7.3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ, РЕКОМЕНДАЦИИ И ДРУГИЕ МАТЕРИАЛЫ К ЗАНЯТИЯМ..	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).....	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	26
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....	29
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.....	30
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	31

АННОТАЦИЯ

рабочей программы учебной дисциплины Б1.В.13 «Разработка пользовательских интерфейсов» для подготовки бакалавров по направлению 09.03.02 Информационные системы и технологии, направленности «Фуллстек разработка», «Системная аналитика и разработка программного обеспечения»

Цель освоения дисциплины: формирование у студентов профессиональных компетенций в области проектирования, разработки и сопровождения современных графических пользовательских интерфейсов для приложений. В ходе изучения дисциплины студенты осваивают методы анализа пользовательских сценариев, создания wireframe-макетов и визуальных гайдлайнов, применяют принципы UX/UI-дизайна и юзабилити, а также приобретают навыки реализации профессиональных интерфейсов с использованием инструментов Qt Designer и библиотеки PySide6. Особое внимание уделяется вопросам адаптивности, доступности, стилизации (Qt Style Sheets), созданию иконки приложения и упаковке готового программного продукта для распространения.

Место дисциплины в учебном плане: дисциплина включена в часть, формируемую участниками образовательных отношений дисциплин 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-3(ПКос-3.1, ПКос-3.2, ПКос-3.3), ПКос-4(ПКос-4.1, ПКос-4.2, ПКос-4.3).

Краткое содержание дисциплины:

Основы UX/UI-дизайна: принципы юзабилити, ментальные модели, законы восприятия, пользовательские сценарии и истории. Проектирование интерфейсов: создание wireframe-макетов, информационная архитектура, визуальные гайдлайны. Архитектура десктопных приложений на PySide6: цикл событий, сигналы и слоты, многооконность, динамическое управление интерфейсом. Работа с Qt Designer: создание и загрузка .ui-файлов, компоновка виджетов, использование QStackedWidget и QTabWidget. Визуальное оформление: стилизация с помощью Qt Style Sheets (QSS), создание тем, работа с цветами, шрифтами и отступами. Графические ресурсы: использование иконок (ICO, PNG, SVG), создание иконки приложения с применением цифровых инструментов и генеративных ИИ-моделей. Отображение данных: таблицы (QTableView), древовидные структуры (QTreeView), интеграция графиков. Адаптивность и доступность: проектирование под разные разрешения экрана, поддержка клавиатурной навигации, обеспечение контрастности и масштабируемости. Работа с внешними данными: загрузка и сохранение файлов (CSV, JSON), интеграция с REST API. Тестирование интерфейса: юзабилити-тестирование, ручная проверка сценариев, итеративное улучшение. Подготовка к распространению: упаковка приложения в исполняемый файл с помощью PyInstaller, встраивание ресурсов, работа с иконками и версиями.

Общая трудоемкость дисциплины: 144 / 4 (часы/зач. ед.)

Промежуточный контроль: экзамен

1. Цель освоения дисциплины

Целью освоения дисциплины «Разработка пользовательских интерфейсов» является формирование у студентов системного понимания принципов проектирования и реализации современных графических пользовательских интерфейсов. Студент должен знать основы UX/UI-дизайна, включая принципы юзабилити, законы зрительного восприятия, методы анализа пользовательских сценариев и создания wireframe-макетов; архитектурные особенности десктопных приложений на базе фреймворка Qt, включая цикл событий, сигналы и слоты, организацию многооконного взаимодействия и работу с динамическим интерфейсом; возможности инструмента Qt Designer и библиотеки PySide6 для визуального и программного конструирования интерфейсов; методы стилизации с использованием Qt Style Sheets (QSS), принципы создания визуальных гайдлайнов и тем; подходы к обеспечению адаптивности под различные разрешения экрана и доступности для пользователей с ограничениями; а также практики подготовки приложения к распространению, включая создание иконки, упаковку в исполняемый файл и встраивание ресурсов. По окончании изучения дисциплины студент должен уметь проектировать интерфейс на основе анализа задачи и пользовательских потребностей, создавать wireframe-макеты и визуальные спецификации, реализовывать многофреймовые и адаптивные интерфейсы в Qt Designer и PySide6, интегрировать отображение табличных и графических данных, применять валидацию ввода и обеспечивать обратную связь с пользователем, стилизовать интерфейс с помощью QSS, создавать иконку приложения с использованием цифровых инструментов и ИИ-генераторов, а также упаковывать готовое приложение в автономный исполняемый файл. Также по окончании изучения дисциплины студент должен владеть навыками проектирования и реализации профессиональных десктопных приложений с использованием паттернов проектирования (Model-View), навыками создания пользовательских сценариев и проведения юзабилити-тестирования, методами обеспечения адаптивности и доступности интерфейсов, техниками визуального оформления (цвет, типографика, иконографика), а также практиками документирования архитектуры интерфейса и оформления пользовательской документации в соответствии с профессиональными стандартами.

2. Место дисциплины в учебном процессе

Дисциплина «Разработка пользовательских интерфейсов» включена в часть, формируемую участниками образовательных отношений дисциплин учебного плана. Дисциплина «Разработка пользовательских интерфейсов» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии.

Дисциплина «Разработка пользовательских интерфейсов» изучается на четвертом курсе образовательного цикла.

Предшествующими курсами, включенными в учебный план, на которых непосредственно базируются дисциплина «Разработка пользовательских интерфейсов», являются «Алгоритмизация и программирование», «Введение в компьютерные науки», «Программирование на языке Python» и «Разработка

профессиональных приложений», в ходе которых студенты получили навыки программирования на Python, работы с графическими библиотеками, проектирования архитектуры приложений и взаимодействия с пользователем.

Дисциплина «Разработка пользовательских интерфейсов» является базовой для последующего освоения курса «Тестирование программного обеспечения», поскольку формирует понимание пользовательских сценариев, требований к юзабилити и структуры интерфейсных компонентов, что необходимо для разработки эффективных тест-кейсов и проведения функционального и UX-тестирования.

Особенностью дисциплины является синтез инженерного и дизайнерского подходов: студенты не только программируют интерфейсы с использованием современных инструментов (PySide6, Qt Designer), но и осваивают методы проектирования, визуального оформления, адаптации под пользователя и подготовки приложения к распространению – от wireframe-макета до исполняемого файла с собственной иконкой.

Рабочая программа дисциплины «Разработка пользовательских интерфейсов» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учётом 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 «Разработка пользовательских интерфейсов»

№ п/п	Код компе- тенции	Содержание компетенции (или её ча- сти)	Индикато- ры компе- тенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	ПКос-3	Способен проектировать и разрабатывать информационные ресурсы с использованием полного стека технологий	ПКос-3.1	Студент должен знать: методы проектирования пользовательских интерфейсов от анализа пользовательских сценариев до создания wireframe-макетов и визуальных гайдлайнов; архитектурные принципы десктопных приложений на Qt; структуру проекта на PySide6, назначение компонентов Qt Designer, типовые шаблоны компоновки, а также подходы к организации данных для отображения; методы обеспечения адаптивности (размерные политики, спейсеры), доступности (клавиатурная навигация, контрастность) и принципы подготовки приложения к распространению (иконки, ресурсы, упаковка).		
			ПКос-3.2		Студент должен уметь: проектировать структуру	

					графического интерфейса в соответствии с поставленной задачей; использовать Qt Designer для визуального создания .ui-файлов и интегрировать их в код на PySide6; применять типовые решения: многофреймовые интерфейсы через QStackedWidget, вкладки через QTabWidget, динамическое управление видимостью элементов; проектировать и реализовывать модели данных, связывать интерфейс с внешними источниками; адаптировать интерфейс под разные разрешения экрана и обеспечивать базовую доступность.	
			ПКос-3.3			Студент должен владеть: навыками проектирования и разработки полноценных приложений от анализа требований и создания wireframe до упаковки исполняемого файла; практиками архитектурного проектирования: разделение логики и представления (Model-View), модульность, повторное использование компонентов; навыками создания и настройки

						визуальной идентичности, применение Qt Style Sheets (QSS) для стилизации, создание тем, подбор цветовых схем и типографики; техниками подготовки приложения к распространению: генерация иконки (в т.ч. с использованием ИИ-инструментов), встраивание ресурсов, работа с PyInstaller.
2.	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	Студент должен знать: особенности программирования графических приложений на Python с использованием PySide6, обработку событий через сигналы и слоты, управление жизненным циклом окон, работу с потоками; синтаксис и идиомы Python, применяемые в GUI-разработке; стандартные модули Python для интеграции с GUI: sys, os, json, pathlib, а также библиотеки для работы с графикой и файлами; типичные источники ошибок в GUI-приложениях: утечки памяти при неправильном подключении сигналов,		

				обращение к виджетам после закрытия окна, блокировка основного потока и форматы диагностических сообщений.		
			ПКос-4.2		Студент должен уметь: читать, писать и отлаживать код GUI-приложений на PySide6, выявляя логические и архитектурные ошибки; использовать среду разработки для навигации по проекту, запуска и пошаговой отладки; интерпретировать сообщения об ошибках; интегрировать внешние компоненты: загружать данные из JSON/CSV, отображать графики; проводить рефакторинг, улучшать читаемость кода, соблюдать PEP 8.	
			ПКос-4.3			Студент должен владеть: навыками разработки, отладки и рефакторинга профессиональных GUI-приложений на Python; техниками повышения надёжности: обработка исключений в слотах, использование try/except с логированием, изоляция потоков; практиками написания чистого и сопровождаемого кода: именован-

						ние переменных, документирование, разделение ответственностей; навыками тестирования интерфейса: ручная проверка сценариев, валидация ввода, проверка реакции на ошибочные данные; оптимизировать взаимодействие с внешней средой.
--	--	--	--	--	--	--

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 4 зач.ед. (144 часов), их распределение по видам работ и семестрам представлено в таблице 2.

Таблица 2

Распределение трудоёмкости дисциплины по видам работ

Вид учебной работы	Трудоёмкость
	час. всего
Общая трудоёмкость дисциплины по учебному плану	144
1. Контактная работа:	53,4
Аудиторная работа	53,4
лекции (Л)	16
практические занятия (ПЗ)	32
курсовой проект (КП) (консультация, защита)	3
консультации перед экзаменом	2
контактная работа на промежуточном контроле (КРА)	0,4
2. Самостоятельная работа (СРС)	63,6
самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)	63,6
Подготовка к экзамену (контроль)	27
Вид промежуточного контроля:	экзамен

4.2 Содержание дисциплины

Таблица 3

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ всего	ПКР	
Тема 1. «Введение в UI/UX: принципы проектирования современных интерфейсов»	8	2	-	-	6
Тема 2 «Проектирование интерфейса: от идеи к макету»	12	2	2	-	8
Тема 3 «Основы PySide6 и Qt Designer»	16	3	6	-	7
Тема 4 «Динамический интерфейс и навигация»	16	2	6	-	8
Тема 5 «Визуальный дизайн и графика»	16	2	5	-	9
Тема 6 «Работа с данными и визуализация»	17	2	6	-	9
Тема 7 «Адаптивность и доступность»	10	1	3	-	6
Тема 8 «Интеграция, тестирование и финальная сборка»	16,6	2	4	-	10,6
Курсовой проект (КП) (консультация, защита)	3	-	-	3	-

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ всего	ПКР	
Подготовка к экзамену (контроль)	27	-	-	-	27
контактная работа на промежуточном контроле (КРА)	0,4	-	-	0,4	-
Консультации перед экзаменом	2	-	-	2	-
Итого по дисциплине	144	16	32	5,4	90,6

Введение в UI/UX: принципы проектирования современных интерфейсов.

Основы пользовательского опыта: различие между UI и UX, принципы юзабилити (ясность, обратная связь, минимализм), законы зрительного восприятия, анализ пользовательских сценариев, типичные ошибки проектирования, требования к интерфейсам в профессиональных и промышленных приложениях (включая АПК).

Проектирование интерфейса: от идеи к макету.

Методы сбора требований: пользовательские истории, сценарии использования, создание wireframe-макетов (бумажных и цифровых), проектирование информационной архитектуры, разработка визуальных гайдлайнов (цветовая палитра, типографика, иконографика), использование инструментов проектирования.

Основы PySide6 и Qt Designer.

Архитектура фреймворка Qt: цикл событий, роль QApplication, жизненный цикл окон, базовые виджеты (QLabel, QPushButton, QLineEdit, QComboBox), компоновка интерфейса, механизм сигналов и слотов, создание и загрузка интерфейсов в Qt Designer (.ui-файлы), интеграция с кодом на Python.

Динамический интерфейс и навигация.

Реализация многофреймовых приложений с использованием QStackedWidget и QTabWidget, динамическое добавление и удаление виджетов, обработка пользовательского ввода и валидация в реальном времени, реализация модальных и немодальных диалогов, управление состоянием интерфейса, обработка ошибок и пользовательская обратная связь (QMessageBox, статусные строки).

Визуальный дизайн и графика.

Стилизация интерфейсов с помощью Qt Style Sheets (QSS): селекторы, свойства, псевдосостояния, создание тем (светлая/тёмная), работа с цветами, шрифтами и отступами, использование иконок (PNG, ICO, SVG), создание иконки приложения с применением цифровых инструментов и генеративных ИИ-моделей, интеграция векторной графики.

Работа с данными и визуализация.

Отображение табличных данных с использованием QTableView и QStandardItemModel, создание кастомных моделей на основе QAbstractTableModel, работа с древовидными структурами (QTreeView), загрузка и сохранение данных в форматах CSV и JSON, интеграция графиков

(matplotlib, pyplot), обработка больших объемов данных без «зависания» интерфейса.

Адаптивность и доступность.

Проектирование интерфейсов, адаптированных под разные разрешения экрана: использование sizePolicy, spacers, minimum/maximum размеров, тестирование на виртуальных устройствах, обеспечение доступности: поддержка клавиатурной навигации (Tab, Enter), достаточная контрастность, масштабируемость шрифтов, использование accessibleName и accessibleDescription в Qt.

Интеграция, тестирование и финальная сборка.

Интеграция с внешними системами: загрузка данных из REST API, обработка файлов, фоновые операции через QThread, проведение юзабилити-тестирования с реальными пользователями, итеративное улучшение интерфейса, упаковка приложения в автономный исполняемый файл с помощью PyInstaller, встраивание иконки и ресурсов, подготовка дистрибутива и пользовательской документации.

4.3 Практические занятия

Таблица 4

Содержание практических занятий и контрольные мероприятия

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
Тема 1. «Введение в UI/UX: принципы проектирования современных интерфейсов»	Лекция 1. «Основы UX/UI: от пользователя к интерфейсу»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	2
Тема 2. «Проектирование интерфейса: от идеи к макету»	Лекция 2. «Методы проектирования: сценарии, wireframe, гайдлайны»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	2
	Практическая работа 1. «Создание пользовательских сценариев и wireframe-макетов»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	2
Тема 3. «Основы PySide6 и Qt Designer»	Лекция 3. «Архитектура Qt и базовые виджеты»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Лекция 4. «Компоновка, сигналы и слоты»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Лекция 5. «Интеграция Qt Designer и Python-кода»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-	-	1

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		4.2, ПКос-4.3.		
	Практическая работа 2. «Создание первой формы в Qt Designer и подключение логики на PySide6»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос -	2
	Практическая работа 3. «Реализация интерактивной формы с валидацией и обратной связью»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос -	4
Тема 4. «Динамический интерфейс и навигация»	Лекция 6. «Многофреймовые интерфейсы и управление состоянием»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	2
	Практическая работа 4. «Реализация многооконного приложения: авторизация – главное окно»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	2
	Практическая работа 5. «Создание динамического интерфейса с QStackedWidget и валидацией ввода»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	4
Тема 5. «Визуальный дизайн и графика»	Лекция 7. «Стилизация интерфейсов с помощью Qt Style Sheets (QSS)»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Лекция 8. «Графические ресурсы: иконки, SVG, создание иконки приложения»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Практическая работа 6. «Разработка темы оформления и применение QSS к интерфейсу»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос -	2
	Практическая работа 7. «Создание иконки приложения с использованием ИИ и редакторов, интеграция в приложение»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	3
Тема 6. «Работа с данными и визуализация»	Лекция 9. «Табличные данные и модели в PySide6»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Лекция 10. «Интеграция графиков и внешних данных»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Практическая работа 8. «Отображение данных из CSV	ПКос-3.1, ПКос-3.2, ПКос-3.3,	Устный опрос	3

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
	в QTableView с кастомной моделью»	ПКос-4.1, ПКос-4.2, ПКос-4.3.		
	Практическая работа 9. «Добавление графиков и динамического обновления данных»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	3
Тема 7. «Адаптивность и доступность»	Лекция 11. «Адаптация интерфейса под разные экраны и обеспечение доступности»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Практическая работа 10. «Тестирование и настройка адаптивности, проверка клавиатурной навигации»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	3
Тема 8. «Интеграция, тестирование и финальная сборка»	Лекция 12. «Интеграция с внешними системами и фоновые операции»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Лекция 13. «Юзабилити-тестирование и подготовка к развёртыванию»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	-	1
	Практическая работа 11. «Проведение юзабилити-теста и финальная доработка интерфейса»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3.	Устный опрос	2
	Практическая работа 12. «Упаковка приложения в .exe с иконкой и ресурсами через PyInstaller»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, 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
1.	Тема 1. «Введение в UI/UX: принципы проектирования современных интерфейсов»	Принципы юзабилити Нильсена: актуальность для десктопных приложений. Отличие UI-дизайна для профессионального ПО от consumer-приложений. Роль ментальных моделей в проектировании интерфейсов. Анализ ошибок в интерфейсах. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
2.	Тема 2. «Проектирование интерфейса: от идеи к макету»	Сравнение инструментов проектирования для учебных проектов. Преимущества и ограничения wireframe-макетов по сравнению с прототипами. Как создать визуальный гайдлайн без опыта в дизайне. Использование систем компонентов (design system) в небольших проектах.

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
		(ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
3.	Тема 3. «Основы PySide6 и Qt Designer»	Разница между наследованием от QMainWindow и QWidget. Преимущества и недостатки загрузки .ui-файлов через QUiLoader и uic. Почему нельзя обновлять интерфейс напрямую из фонового потока. Особенности жизненного цикла окна Qt: closeEvent, showEvent. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
4.	Тема 4. «Динамический интерфейс и навигация»	Сравнение QStackedWidget, QTabWidget и динамической замены центрального виджета. Как избежать утечки памяти при многократном создании окон. Обработка валидации в реальном времени с помощью textChanged и timer. Реализация «мастера» (wizard) с помощью QStackedWidget. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
5.	Тема 5. «Визуальный дизайн и графика»	Особенности работы QSS по сравнению с CSS: поддерживаемые свойства, псевдосостояния. Как создать тёмную тему без дублирования кода. Преимущества SVG-иконок перед PNG в десктопных приложениях. Использование ИИ-генераторов для создания иконок: ограничения и доработка в редакторах. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
6.	Тема 6. «Работа с данными и визуализация»	Разница между QStandardItemModel и QAbstractTableModel: когда что использовать. Как отображать иерархические данные (например, дерево категорий задач). Интеграция matplotlib vs pyqtgraph: плюсы и минусы для десктопа. Обработка больших таблиц без замедления интерфейса (виртуализация строк). (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
7.	Тема 7. «Адаптивность и доступность»	Как Qt обрабатывает изменение размера окна: role spacers и stretch. Настройка минимального и предпочтительного размера виджетов. Поддержка screen readers в Qt: доступные свойства и их ограничения. Требования к контрастности по стандарту WCAG и их применимость к десктопу. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-4.1, ПКос-4.2, ПКос-4.3).
8.	Тема 8. «Интеграция, тестирование и финальная сборка»	Как PyInstaller упаковывает зависимости и ресурсы. Почему иконка не отображается в .exe и как это исправить. Ограничения работы с QThread: запрет прямого обращения к виджетам из потока. Методы юзабилити-тестирования в условиях учебной группы (cognitive walkthrough, А/В-тестирование макетов). (ПКос-3.1, ПКос-3.2, ПКос-3.3, 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.	Тема 2. «Проектирование интерфейса: от идеи к макету»	ПЗ	Мозговой штурм
2.	Тема 4. «Динамический интерфейс и навигация»	ПЗ	Разбор конкретных ситуаций
3.	Тема 5. «Визуальный дизайн и графика»	ПЗ	Разбор конкретных ситуаций
4.	Тема 7. «Адаптивность и доступность»	ПЗ	Разбор конкретных ситуаций

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

Вопросы для подготовки к устным опросам

Тема 1. Введение в UI/UX: принципы проектирования современных интерфейсов

1. В чём разница между UI и UX? Почему оба аспекта важны даже в утилитарных приложениях (например, учёт техники в АПК)?
2. Назовите три принципа юзабилити по Нильсену и приведите пример их нарушения в реальном ПО.
3. Что такое «ментальная модель» пользователя и как она влияет на проектирование интерфейса?
4. Почему интерфейс профессионального ПО (например, 1С) часто сложнее, чем consumer-приложения (например, Telegram)?
5. Какие ошибки чаще всего допускают студенты при проектировании первых интерфейсов?

Тема 2. Проектирование интерфейса: от идеи к макету

1. Зачем нужен wireframe, если можно сразу начать писать код в Qt Designer?
2. Как пользовательские сценарии помогают избежать «лишних» кнопок и полей в интерфейсе?

3. Что входит в визуальный гайдлайн и почему он полезен даже для одного разработчика?
4. Какие инструменты вы использовали для создания макета и почему выбрали именно их?
5. Можно ли использовать ИИ для генерации wireframe? Почему это не всегда уместно?

Тема 3. Основы PySide6 и Qt Designer

1. В чём разница между QMainWindow и QWidget как корневым контейнере?
2. Почему нельзя обновлять виджеты напрямую из функции, вызванной по таймеру или в другом потоке?
3. Как загрузить .ui-файл в Python-код и подключить к нему логику обработки событий?
4. Что такое сигнал и слот? Приведите пример без использования lambda.
5. Почему важно вызывать super().__init__() при наследовании от виджетов Qt?

Тема 4. Динамический интерфейс и навигация

1. В чём разница между QMainWindow и QWidget как корневым контейнере?
2. Почему нельзя обновлять виджеты напрямую из функции, вызванной по таймеру или в другом потоке?
3. Как загрузить .ui-файл в Python-код и подключить к нему логику обработки событий?
4. Что такое сигнал и слот? Приведите пример без использования lambda.
5. Почему важно вызывать super().__init__() при наследовании от виджетов Qt?

Тема 5. Визуальный дизайн и графика

1. В чём разница между QMainWindow и QWidget как корневым контейнере?
2. Почему нельзя обновлять виджеты напрямую из функции, вызванной по таймеру или в другом потоке?
3. Как загрузить .ui-файл в Python-код и подключить к нему логику обработки событий?
4. Что такое сигнал и слот? Приведите пример без использования lambda.
5. Почему важно вызывать super().__init__() при наследовании от виджетов Qt?

Тема 6. Работа с данными и визуализация

1. В чём разница между QStandardItemModel и QAbstractTableModel? Когда стоит использовать второй?
2. Как отобразить данные из CSV-файла в QTableView без загрузки всего файла в память?
3. Почему при обновлении модели напрямую интерфейс может «зависнуть»? Как этого избежать?
4. Как добавить график на виджет и обновлять его по таймеру?
5. Как реализовать сортировку и фильтрацию в таблице без перезагрузки данных?

Тема 7. Адаптивность и доступность

1. Как сделать так, чтобы кнопка всегда оставалась прижатой к правому краю окна при его растягивании?
2. Что такое `sizePolicy` и как он влияет на поведение виджета при изменении размера окна?
3. Как проверить, поддерживает ли ваш интерфейс навигацию с клавиатуры (Tab, Enter)?
4. Какие минимальные требования к контрастности текста и фона для обеспечения базовой доступности?
5. Почему одни и те же координаты виджетов могут выглядеть по-разному на разных мониторах?

Тема 8. Интеграция, тестирование и финальная сборка

1. Почему иконка не отображается в `.exe` после упаковки через PyInstaller и как это исправить?
2. Как передать данные между потоком (`QThread`) и основным интерфейсом без нарушения правил Qt?
3. Какие шаги вы предприняли для юзабилити-тестирования своего приложения?
4. Что такое «файл спецификации» (`.spec`) в PyInstaller и зачем он нужен?
5. Как убедиться, что ваше приложение корректно работает на компьютере без Python?

Критерии оценки: опрос осуществляет из всего списка студентов. За каждый вопрос, студент может получить максимум 10 баллов. 1 студент, в течение устного опроса отвечает только на 1 вопрос. Таким образом, за каждый опрос студент может получить максимум 10 баллов.

Перечень вопросов, выносимых на промежуточную аттестацию (экзамен)

1. Как принцип «согласованность» (consistency) влияет на скорость освоения ПО агрономом?
2. Почему «минимизация когнитивной нагрузки» особенно важна в интерфейсах для работы в поле?
3. В чём опасность избыточной «игровой» стилизации в профессиональных приложениях АПК?
4. Как использовать паттерны проектирования интерфейсов (UI patterns) без потери оригинальности?
5. Почему интерфейс не должен «навязывать» пользователю единственный способ выполнения задачи?
6. Как `wireframe` помогает выявить избыточность функционала до начала программирования?
7. Зачем указывать состояния элементов (ошибка, загрузка, успех) уже на этапе макета?
8. Как проектировать интерфейс, если пользователь — не один, а несколько ролей (фермер, аналитик, админ)?
9. Почему важно фиксировать пользовательские сценарии в письменном виде, а не держать в голове?

10. Какие ограничения накладывает бесплатная версия Figma на учебные проекты?
11. Почему в Qt Designer нельзя задать обработчик события напрямую, как в Visual Studio?
12. Какие виджеты Qt не являются QWidget под капотом? (Подсказка: QLayout)
13. Как правильно отключить сигнал, чтобы избежать множественного вызова слота?
14. Почему QApplication.setStyle() может не сработать на Windows?
15. Как отладить, какой именно виджет получает событие нажатия мыши?
16. Как реализовать «забыл пароль» в форме авторизации без открытия нового окна?
17. Почему QStackedWidget предпочтительнее setVisible() при частой смене экранов?
18. Как сохранить введенные данные при переключении между вкладками QTabWidget?
19. Как обрабатывать закрытие диалога по крестику и отличать его от нажатия «Отмена»?
20. Как динамически изменить надпись на кнопке в зависимости от состояния приложения?
21. Почему QSS не поддерживает margin и padding как в вебе? Как достичь аналогичного эффекта?
22. Как загрузить SVG-иконку в QPushButton и изменить её цвет динамически?
23. Какие риски несёт использование ИИ-генераторов для создания иконок в профессиональном ПО?
24. Как применить тему оформления ко всему приложению, включая диалоговые окна?
25. Почему QSS-стили могут не применяться к кастомным виджетам и как это исправить?
26. Как реализовать «ленивую» загрузку строк в QTableView для очень больших данных?
27. Почему QAbstractTableModel требует реализации rowCount(), columnCount(), data()?
28. Как обновить только одну ячейку таблицы, не перерисовывая всю модель?
29. Как добавить выпадающий список (QComboBox) внутрь ячейки QTableView?
30. Почему matplotlib может «зависать» интерфейс и как его интегрировать через QThread?
31. Как протестировать интерфейс на мониторе с масштабированием 150% без физического устройства?
32. Почему setMinimumSize() может нарушить адаптивность и когда его стоит избегать?
33. Как обеспечить, чтобы все виджеты были доступны через Tab даже при сложной компоновке?

34. Какие системные настройки Windows/macOS влияют на отображение вашего приложения?
35. Почему «адаптивность» в десктопе — не то же самое, что «responsive design» в вебе?
36. Как включить логирование в упакованное PyInstaller-приложение для диагностики ошибок?
37. Почему PyInstaller может не найти модуль при сборке и как указать скрытые импорты?
38. Как протестировать, что ваше приложение корректно освобождает ресурсы при закрытии?
39. Какие файлы обязательно должны быть включены в дистрибутив (кроме .exe)?
40. Как автоматизировать сборку .exe для разных платформ (Windows, Linux) из одного кода?

Практическая работа № 1.

«Создание пользовательских сценариев и wireframe-макетов»

Цель: освоить методы проектирования пользовательского интерфейса на основе анализа задач и потребностей пользователя; научиться создавать wireframe-макеты с использованием цифровых инструментов.

Условие: разрабатывается десктопное приложение для учёта сельхозтехники в агрохолдинге. Пользователи – механик (ввод данных о технике и ТО) и инженер (просмотр отчётов, планирование ремонта).

Требуется:

1. Составить пользовательские сценарии:
 - а. Описать 2-3 типичные задачи для механика (например: «Добавить новую технику», «Зарегистрировать ТО»).
 - б. Описать 2-3 задачи для инженера (например: «Просмотреть список техники, требующей ремонта», «Сформировать отчёт за месяц»).
2. Создать wireframe-макеты:
 - а. Используя Figma, Excalidraw или аналог, разработать макеты всех экранов приложения (не менее 3: главное меню, форма ввода, отчёт).
 - б. На макетах обозначить: кнопки, поля ввода, таблицы, навигацию.
 - с. Указать состояния элементов (например: «поле с ошибкой», «кнопка неактивна»).
3. Разработать визуальный гайдлайн:
 - а. Определить цветовую палитру (основной, акцентный, фон, текст).
 - б. Выбрать шрифты и размеры для заголовков и основного текста.
 - с. Описать правила использования иконок и отступов.
4. Оформить результат:
 - а. Сохранить макеты в PDF или PNG.
 - б. Написать краткое пояснение: как сценарии повлияли на структуру интерфейса.

Практическое задание № 2

«Создание первой формы в Qt Designer и подключение логики на PySide6»

Цель: освоить базовые навыки проектирования интерфейса в Qt Designer и интеграции его с программной логикой на PySide6.

Условие: требуется реализовать форму «Личные данные оператора» для дальнейшего использования в системе учёта.

Требуется, используя Qt Designer и PySide6:

1. Создать интерфейс в Qt Designer:
 - a. Создать форму с полями: ФИО (QLineEdit), дата рождения (QDateEdit), пол (QRadioButton или QComboBox), должность (QLineEdit), email (QLineEdit).
 - b. Добавить кнопки «Сохранить» и «Отмена».
 - c. Настроить компоновку с использованием QFormLayout и QVBoxLayout.
 - d. Сохранить интерфейс в файл form.ui.
2. Подключить логику на PySide6:
 - a. Написать Python-скрипт, загружающий form.ui.
 - b. Подключить обработчики к кнопкам:

При нажатии «Сохранить» – вывести в консоль все введённые данные.

При нажатии «Отмена» – закрыть окно.
3. Реализовать базовую валидацию:
 - a. Проверить, что поле «ФИО» не пустое.
 - b. Проверить корректность email (наличие «@»).
 - c. При ошибке – вывести QMessageBox с пояснением, не закрывая форму.
4. Протестировать работу:
 - a. Запустить приложение, ввести корректные и некорректные данные.
 - b. Убедиться, что обработчики работают, а интерфейс не «зависает».

Комплект разноуровневых заданий

Тема 1 «Введение в UI/UX: принципы проектирования современных интерфейсов»

Задание репродуктивного уровня

Перечислите пять принципов юзабилити и дайте краткое определение каждому.

Задание реконструктивного уровня

Приведите пример интерфейса (реального или гипотетического), в котором нарушены три принципа юзабилити. Предложите конкретные улучшения для каждого нарушения.

Тема 2 «Проектирование интерфейса: от идеи к макету»

Задание репродуктивного уровня

Опишите, что такое пользовательская история и из каких трёх частей она обычно состоит («Как..., я хочу..., чтобы...»).

Задание реконструктивного уровня

Дано: пользовательская история – «Как агроном, я хочу видеть список полей, чтобы планировать посев». Создайте wireframe-макет экрана, удовлетворяющего этой истории, и обоснуйте выбор расположения элементов с точки зрения UX.

Тема 3 «Основы PySide6 и Qt Designer»

Задание репродуктивного уровня

Опишите разницу между QMainWindow, QWidget и QDialog. В каких случаях какой из них следует использовать?

Задание реконструктивного уровня

Дан фрагмент кода на PySide6, в котором интерфейс «зависает» при нажатии кнопки. Найдите ошибку (предположительно – долгая операция в основном потоке) и предложите исправленную архитектуру с использованием сигналов и потоков.

Тема 4 «Динамический интерфейс и навигация»

Задание репродуктивного уровня

Опишите, как реализовать:

1. скрывание/показ элементов интерфейса,
2. динамическое добавление строк в форму,
3. обработку нажатия на кнопку, созданную программно.

Задание реконструктивного уровня

Дано: приложение с QStackedWidget, в котором при переключении между экранами теряются введённые пользователем данные. Предложите архитектурное решение, позволяющее сохранять состояние каждого экрана без пересоздания виджетов.

Тема 5 «Визуальный дизайн и графика»

Задание репродуктивного уровня

Перечислите основные свойства, поддерживаемые в Qt Style Sheets (QSS), и приведите пример стиля для QPushButton в состояниях :hover и :pressed.

Задание реконструктивного уровня

Студент создал иконку приложения в формате PNG 256×256, но на экране с высоким DPI она выглядит размытой. Объясните причину и предложите решение с использованием векторной графики и инструментов Qt.

Тема 6 «Работа с данными и визуализация»

Задание репродуктивного уровня

Опишите разницу между прямой работой с QTableView и использованием паттерна Model-View. Какой подход предпочтительнее для масштабируемого приложения?

Задание реконструктивного уровня

Опишите поэтапно, как реализовать приложение для отображения данных из CSV-файла:

1. с использованием QStandardItemModel,

2. с использованием кастомной модели на основе QAbstractTableModel.

Укажите преимущества второго подхода.

Тема 7 «Адаптивность и доступность»

Задание репродуктивного уровня

Какие инструменты Qt позволяют обеспечить адаптацию интерфейса под разные разрешения экрана? Опишите назначение QSpacerItem, sizePolicy и stretch.

Задание реконструктивного уровня

Приложение корректно отображается на мониторе 1920×1080, но на ноутбуке с разрешением 1366×768 часть элементов уходит за границу окна. Предложите три конкретных шага по переделке компоновки, которые решат проблему без ущерба для UX.

Тема 8 «Интеграция, тестирование и финальная сборка»

Задание репродуктивного уровня

Опишите основные шаги упаковки приложения в исполняемый файл с помощью PyInstaller. Как включить в дистрибутив иконку и QSS-файл?

Задание реконструктивного уровня

После упаковки приложения через PyInstaller оно запускается, но не отображает иконку и игнорирует QSS-стили. Проанализируйте возможные причины и предложите решение с примером файла .spec и корректной загрузки ресурсов.

Критерии оценки: принимаются преподавателем и оцениваются вместе с устными опросами.

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

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущей работы в семестре.

Работы должны быть выполнены по своему варианту, оформлены в соответствии с требованиями стандартов по оформлению текстовых документов в текстовом редакторе MS Word. Работы сдаются в электронном виде.

По результатам защиты могут быть получены следующие баллы:

9-10 баллов – расчеты (если имеются) проведены корректно, результаты правильно интерпретированы. Полностью выполнены все пункты выданного задания. Работа оформлена в соответствии с требованиями стандартов по оформлению текстовых документов. Студент развернуто и свободно ответил на все вопросы при защите работы.

7-8 баллов – работа выполнена, выполнены все пункты выданного задания, но не полностью, либо с несущественными ошибками, имеются незначительные ошибки в интерпретации результатов и/или оформлении. Студент в целом ответил на все поставленные вопросы, ориентируется в работе.

4-6 баллов – работа в целом выполнена, выполнены основные, но не все пункты выданного задания, либо с существенными ошибками, имеются значительные ошибки в интерпретации полученных результатов и представления данных, оформления работы. Некоторые вопросы по работе вызывают затруднения.

1-3 балла – имеются грубые ошибки в методике выполнения, интерпретации полученных результатов и представления данных, оформления работы, большая часть пунктов выданного задания не выполнена. Студент не отвечает на вопросы при защите.

В течение периода обучения по дисциплине студент должен выполнить и защитить 11 практических заданий (индивидуальных или групповых проектов), каждое из которых оценивается максимум на 10 баллов. За посещение занятий добавляется 0,15 балла за каждый час ($68 \cdot 0,15$), участие в конференции с докладом по теме, связанной с возможностями практического применения языка Python – 10 баллов. Таким образом, максимально возможная сумма баллов равна: $11 \cdot 10 + 68 \cdot 0,15 + 10 = 110 + 10 + 10 = 130$.

Зачет по дисциплине получают студенты, набравшие не менее 60% от максимального количества баллов, т.е. 78 баллов и более.

Итоговая оценка по дисциплине выставляется преподавателем в соответствии со шкалой:

Текущий рейтинг	Оценка			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
в процентах	0-59	60-69	70-84	85-100
в баллах	0-77	78-90	91-110	111-130

Студенты, набравшие в течение семестра менее 78 баллов, пишут итоговую зачетную работу. К написанию экзамены допускаются студенты, **в случае выполнения всех практических работ.**

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

1. Баланов, А. Н. Прототипирование и разработка пользовательского интерфейса: оптимизация UX : учебное пособие для вузов / А. Н. Баланов. – Санкт-Петербург : Лань, 2024. – 220 с. – ISBN 978-5-507-49211-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/414929> – Режим доступа: для авториз. пользователей.

2. Климченко, К. П. Разработка пользовательских интерфейсов : методические рекомендации / К. П. Климченко, Р. А. Исаев, Р. С. Толмасов. –

Москва : РТУ МИРЭА, 2022. – 25 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/310874> – Режим доступа: для авториз. пользователей.

3. Золкин, А. Л. Инструментальные средства разработки интеллектуальных информационных систем : учебник для вузов / А. Л. Золкин. – Санкт-Петербург : Лань, 2025. – 140 с. – ISBN 978-5-507-51532-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/450848> – Режим доступа: для авториз. пользователей.

7.2 Дополнительная литература

1. Богданова, С. В. Технологии программирования : учебник / С. В. Богданова, А. Н. Ермакова. – Ставрополь : СтГАУ, 2024 – Часть 1 – 2024. – 236 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/510225> – Режим доступа: для авториз. пользователей.

2. Владыко, И. Ю. Разработка интерфейса кроссплатформенных бизнес-приложений : учебно-методическое пособие / И. Ю. Владыко, Е. А. Зеленая. – Москва : РТУ МИРЭА, 2025. – 113 с. – ISBN 978-5-7339-2601-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/504860> – Режим доступа: для авториз. пользователей.

3. Никулова, Г. А. Проектирование и реализация Web-интерфейса : учебно-методическое пособие / Г. А. Никулова. – Липецк : Липецкий ГПУ, 2020. – 66 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/156075> – Режим доступа: для авториз. пользователей.

4. Елисеев, А. И. Разработка программных интерфейсов веб-приложений с использованием фреймворка FastAPI : учебное пособие / А. И. Елисеев, Ю. В. Минин. – Тамбов : ТГТУ, 2024. – 81 с. – ISBN 978-5-8265-2821-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/472310> – Режим доступа: для авториз. пользователей.

5. Остроух, А. В. Проектирование информационных систем : монография / А. В. Остроух, Н. Е. Суркова. – 2-е изд., стер. – Санкт-Петербург : Лань, 2021. – 164 с. – ISBN 978-5-8114-8377-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/175513> – Режим доступа: для авториз. пользователей.

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

1. Python 3.11.1 documentation. – URL: <https://docs.python.org/3/>
2. Google's Python Class. – URL: <https://developers.google.com/edu/python>
3. Machine Learning Crash Course. – URL: <https://developers.google.com/machine-learning/crash-course>

4. Введение в Python. – URL: <https://steps.2035.university/collections/c4706f68-0aa9-419b-8d8a-c9a968a108fc>
5. Цифровые профессии: Искусственный интеллект. – URL: <https://steps.2035.university/collections/f6361b9a-ea2e-41b1-a18f-9a2f84a9fcd4>
6. Kaggle. – URL: <https://www.kaggle.com/>
7. Qt for Python. – https://wiki.qt.io/Qt_for_Python
8. PySide – <https://pypi.org/project/PySide6/>
9. QtDesigner – <https://doc.qt.io/qt-6/qtdesigner-manual.html>
10. Anaconda – <https://www.anaconda.com/download>

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	Тема 1. «Введение в веб-разработку и архитектура современных приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
2	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
3	Тема 3 «Продвинутая разработка на Django»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
4	Тема 4 «Современные API с FastAPI и асинхронностью»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
5	Тема 5 «Тестирование, безопасность и развёртывание веб-приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012

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
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 (2й учебный корпус, 102 ауд.)	1. Компьютер – 29 шт.; 2. Стенд «Сергеев Сергей Степанович 1910-1999» (Инв.№591013/25) – 1 шт.; 3. Огнетушитель порошковый (Инв. №559527) – 1 шт.; 4. Подвесное крепление к огнетушителю (Инв. № 559528) – 1 шт.; 5. Жалюзи (Инв. №1107-221225, Инв. №1107-221225) – 2шт.; 6. Стул – 29 шт.; 7. Стол компьютерный – 28 шт.; 8. Стол для преподавателя – 1 шт.; 9. Доска маркерная (Инв. № 558762/5) – 1 шт.; 10. Трибуна напольная (без инв. №) – 1 шт.
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, помещение для самостоятельной работы (2й учебный корпус, 106 ауд.)	1. Рабочая станция FORSITE TH1516G512G, Российская Федерация A4Tech Fstyler F1512 – 16 шт.; 2. Стол наборный (Инв. №410136000010828) – 1 шт. 3. Стол компьютерный (Инв. № 410136000010813-410136000010827) – 15 шт.; 4. Стул (Инв. № 410136000010829-410136000010853) – 25 шт.; 5. Интерактивная панель (Инв. № 410124000603715) – 1 шт.
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, помещение для самостоятельной работы (2й учебный корпус, 302 ауд.)	1. Компьютер – 16 шт. 2. Телевизор – 1 шт. 3. Стол для преподавателя –1 шт. 4. Стол компьютерный –16 шт. 5. Стул офисный – 17 шт. 6. Компьютер: PRO-3159209 Intel Core i5-10400 2900МГц, Intel B460, 16Гб DDR4, Intel UHD Graphics 630 (встроенная), SSD 240Гб, 500Вт, Mini-Tower – 1 шт. 7. Кондиционер HAIER HSU -24HPL03/R3 (Инв. № 210134000062198) – 1 шт. 8. Вешалка напольная (Инв.№1107-333144, Инв.№1107-333144) – 2 шт.
учебная аудитория для проведения за-	1. Трибуна напольная (Инв.№ 599206) – 1 шт.;

нятий 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, помещение для самостоятельной работы (2й учебный корпус, 303 ауд.)	2. Жалюзи (Инв.№591110) – 1 шт.; 3. Доска маркетинговая (Инв.№ 35643/4) – 1 шт.; 4. Стол – 15 шт.; 5. Скамейка – 14 шт.; 6. Стол эрго – 1 шт.; 7. Стул – 16 шт.
Центральная научная библиотека имени Н.И. Железнова	Читальные залы библиотеки
Студенческое общежитие	Комната для самоподготовки

11. Методические рекомендации студентам по освоению дисциплины

Приступая к изучению дисциплины «Разработка пользовательских интерфейсов», 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.

В ходе занятий вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к практическим занятиям изучить основную литературу, ознакомиться с дополнительной литературой в соответствии с поставленной задачей. При этом учесть рекомендации преподавателя и требования учебной программы. Необходимо дорабатывать свой конспект, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

При подготовке к экзамену (в конце семестра) повторять пройденный материал в строгом соответствии с учебной программой. Использовать конспекты и литературу, рекомендованную преподавателем. Обратить особое внимание на темы учебных занятий, пропущенных студентом по разным причинам. При необходимости обратиться за консультацией и методической помощью к преподавателю.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятия обязан самостоятельно подготовиться к теме устного опроса, которые состоялись на практическом занятии. В рамках часов консультаций студент может ответить на вопросы пропущенного устного опроса, которые были пропущены.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

Курс «Разработка пользовательских интерфейсов» должен давать не абстрактно-формальные, а прикладные знания. Данная цель может быть реализована только при условии соблюдения в учебных планах преемственности учебных дисциплин. Базовые знания для изучения Разработка пользовательских интерфейсов дают такие дисциплины, как программирование на Python, Алгоритмизация и программирование, Разработка профессиональных приложений, Введение в компьютерные науки. Освоение основных тем данной дисциплины позволит студентам сформировать представление о таком сложном предмете как тестирование программного обеспечения, понять всю ширину науки и получить необходимые знания для последующего профессионального развития в этой области.

Студент может подготовить доклад по теме, представляющей его научный интерес, представить результаты в виде презентации. 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 экзамена по этой дисциплине.

Преподаватель должен указывать, в какой последовательности следует изучать материал дисциплины, обращать внимание на особенности изучения отдельных тем и разделов, помогать отбирать наиболее важные и необходимые сведения из учебных пособий, а также давать объяснения вопросам программы курса, которые обычно вызывают затруднения. При этом преподавателю необходимо учитывать следующие моменты:

1. Не следует перегружать студентов творческими заданиями.
2. Чередовать творческую работу на занятиях с заданиями во внеаудиторное время.
3. Давать студентам четкий инструктаж по выполнению самостоятельных заданий: цель задания; условия выполнения; объем; сроки; требования к оформлению.
4. Осуществлять текущий учет и контроль за самостоятельной работой.
5. Давать оценку и обобщать уровень усвоения навыков самостоятельной, творческой работы.

Программу разработал:

Демичев В.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)



(подпись)

(подпись)

РЕЦЕНЗИЯ

на рабочую программу дисциплины Б1.В.13 «Разработка пользовательских интерфейсов» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Фуллстек разработка», «Системная аналитика и разработка программного обеспечения» (квалификация выпускника – бакалавр)

Вахрушевой Инны Алексеевны, доцентом кафедры высшей математики, кандидатом педагогических наук (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Разработка пользовательских интерфейсов» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Фуллстек разработка», «Системная аналитика и разработка программного обеспечения» (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, Храмов Дмитрий Эдуардович, ассистент кафедры статистики и кибернетики).

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

1. Предъявленная рабочая программа дисциплины «Разработка пользовательских интерфейсов» (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 дисциплинам части, формируемой участниками образовательных отношений – 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 «Разработка пользовательских интерфейсов» закреплены **2 профессиональные компетенции (ПКос)**. Дисциплина «Разработка пользовательских интерфейсов» и представленная Программа способна реализовать их в объявленных требованиях.

5. **Результаты обучения**, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

6. Общая трудоёмкость дисциплины «Разработка пользовательских интерфейсов» составляет 4 зачётные единицы (144 часа).

7. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Разработка пользовательских интерфейсов» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии и возможность дублирования в содержании отсутствует.

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

9. Программа дисциплины «Разработка пользовательских интерфейсов» предполагает проведение занятий в интерактивной форме.

10. 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.

11. Представленные и описанные в Программе формы *текущей* оценки знаний (устный опрос), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме экзамена в восьмом семестре, что соответствует статусу дисциплины, как дисциплины части, формируемой участниками образовательных отношений дис-

циплин учебного плана – Б1.В ФГОС ВО направления 09.03.02.Информационные системы и технологии.

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

13. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 3 источника (базовые учебники), дополнительной литературой – 5 наименования, Интернет-ресурсы – 10 источников и соответствует требованиям ФГОС ВО направления 09.03.02 Информационные системы и технологии.

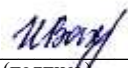
14. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Разработка пользовательских интерфейсов» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

15. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Разработка пользовательских интерфейсов».

ОБЩИЕ ВЫВОДЫ

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Разработка пользовательских интерфейсов» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Фуллстек разработка», «Системная аналитика и разработка программного обеспечения» (квалификация выпускника – бакалавр), разработанная Демичевым Вадимом Владимировичем, доцентом, кандидатом экономических наук, Храмовым Дмитрием Эдуардовичем, 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: Вахрушева Инна Алексеевна, доцент кафедры высшей математики ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат педагогических наук


(подпись)

«26» августа 2025 г.