

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:

Проректор по учебной работе,
качеству образования – первый
проректор

_____ Хагуров Т.А.

подпись

« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1. О.27 Кроссплатформенные десктоп приложения

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Кроссплатформные десктоп приложения» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная математика и информатика.

Программу составили:

В.И. Шиян, ст. преп. КВТ

И.О. Фамилия, должность, ученая степень, ученое звание



подпись

Т.А. Приходько, доц. КВТ, к.т.н., доц.

И.О. Фамилия, должность, ученая степень, ученое звание



подпись

Рабочая программа дисциплины «Кроссплатформные десктоп приложения» утверждена на заседании кафедры вычислительных технологий протокол №1 от 26 августа 2025 г.

И.о. заведующего кафедрой (разработчика)

Т. А. Приходько

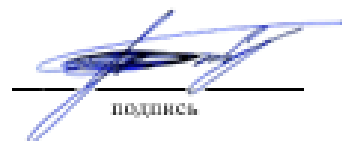


подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №1 от 28 августа 2025 г.

Председатель УМК факультета

А. В. Коваленко



подпись

Рецензенты:

Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг»,
e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, д.э.н., к.т.н., профессор кафедры компьютерных технологий и систем ФГБОУ ВО «КубГАУ им. И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цель дисциплины – освоение принципов и практик разработки кроссплатформенных десктопных приложений на Python и C++ с использованием Qt, GTK и Tkinter, включая архитектурные паттерны (MVC/MVVM), визуализацию данных, тестирование, упаковку и доставку приложений под Windows, Linux и macOS.

1.2 Задачи дисциплины

1. Изучить событийно-ориентированную модель и жизненный цикл GUI-приложений.
2. Освоить Qt (C++/PySide/PyQt), GTK (gtkmm/PyGObject) и Tkinter (Python) для создания UI.
3. Освоить библиотеки визуализации данных: Qt Charts/QCustomPlot, Matplotlib, PyQtGraph, VTK.
4. Освоить межязыковое взаимодействие Python↔C++ (расширения/обёртки), IPC, многопоточность/асинхронность.
5. Освоить сборку и упаковку (CMake/qmake/Meson; windeployqt/macdeployqt; PyInstaller/Briefcase; AppImage/Flatpak/Snap; MSI/DMG).
6. Освоить локализацию, доступность (a11y), тестирование GUI, профилирование и оптимизацию.
7. Освоить практики CI/CD для десктопных приложений (GitLab CI, автосборки, код-сайнинг, автообновления).

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Кроссплатформенные десктоп приложения» относится к базовой части Б1.О.27.

Дисциплина в значительной степени **взаимодействует для формирования компетенций** с дисциплинами:

1. Фронтенд-разработка
2. Базы данных
3. Алгоритмы и структуры данных
4. Операционные системы
5. Алгебра и введение в тензорный анализ

Требованием к «входным» знаниям является понимание основ программирования на C++, алгоритмов и структур данных, операционных систем.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: **Data Engineer (Инженер по данным)**

Задачи:

- Проектирование и построение ETL-процессов
- Создание и оптимизация хранилищ данных
- Обеспечение качества и доступности данных
- Настройка инфраструктуры для обработки больших данных
- Интеграция разрозненных источников данных
- Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: ML Engineer (Инженер МО)

Задачи:

- Реализация ML-моделей в продуктивных системах
- Оптимизация производительности и масштабирование моделей
- Разработка ML-пайплайнов и автоматизация процессов
- Мониторинг качества моделей в продуктиве
- Интеграция ML-решений с бизнес-приложениями

Роль 3: MLOps (Специалист по эксплуатации ИИ)

Задачи:

- Автоматизация процессов обучения и развертывания моделей
- Мониторинг производительности ML-систем
- Управление версиями моделей и данных
- Обеспечение CI/CD для ML-проектов
- Оптимизация вычислительных ресурсов

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

- УК-2** *Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений*
- УК-2.3 Использует принципы проектной методологии для решения профессиональных задач
Знает основы проектного управления (цели/результаты, WBS, спринты, метрики), риск-менеджмент; принципы формулирования требований и критериев приёмки; специфику кроссплатформенной разработки (Windows/Linux/macOS), факторы выбора стека GUI (Qt/GTK/Tkinter) и библиотек визуализации.
Умеет декомпозировать постановку на задачи/сроки; вести бэклог/дорожную карту; составлять критерии готовности; аргументировать выбор инструментов под ограничения (производительность, доступность, лицензии, сроки). Владеет практиками планирования в трекерах (GitLab Issues/Jira), ведением ADR/рисков, релизным календарём и версионированием (SemVer) для desktop-проектов.
- УК-2.4 Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария
Знает методы сравнительной оценки альтернатив (multi-criteria, TCO/риски), основные профайлеры и средства диагностики GUI; требования к локализации и доступности (i18n/a11y).
Умеет готовить прототипы на Qt/GTK/Tkinter; проводить бенчмаркинг/профилирование (время отклика UI, FPS графики), готовить обоснование выбора (trade-off).
Владеет инструментами измерений (Qt Creator Analyzer, Valgrind, cProfile, PyQtGraph/VTK бенчмарки), приёмами снижения рисков (фича-флаги, PoC, инкрементальные релизы).
- ОПК-4** *Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности*
- ОПК-4.1 Аргументировано применяет современные информационные технологии, в

том числе отечественные, при создании программных продуктов и программных комплексов различного назначения
 Знает кроссплатформенные GUI-фреймворки Qt/GTK/Tkinter, их событийные циклы и паттерны (MVC/MVVM); стек визуализации (Qt Charts/QCustomPlot/Matplotlib/PyQtGraph/VTK); принципы локализации/all; инструменты сборки/упаковки и основы CI для desktop.
 Умеет строить функциональные экраны/диалоги; подключать визуализацию (2D/3D) и экспорт в PNG/SVG/PLY; реализовывать неблокирующую обработку данных (поток/задачи/async); собирать и пакетировать приложение под Windows/Linux/macOS.
 Владеет CMake/qmake/Meson, PyInstaller/Briefcase, windeployqt/macdeployqt; средствами профилирования/логирования; практиками автоматической сборки релизов (GitLab CI).

ОПК-4.2 Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов
 Знает практики сопровождения desktop-приложений (логирование/трассировка, автообновления, телеметрия, обратная совместимость), подходы к тестам GUI (smoke/regression).
 Умеет настраивать сбор телеметрии/журналов, локализацию и переключение языка в рантайме; внедрять автообновление и стратегию релизов; писать дымовые/регрессионные тесты GUI.
 Владеет инструментами pytest-qt/dogtail (или аналогами), системами доставки обновлений и публикации артефактов, приемами поддержания качества (код-ревью, линт, статанализ).

PL-1 П Способен применять язык программирования Python для решения задач в области ИИ

PL-1.1 Разрабатывает и отлаживает прикладные решения разной сложности и для разного круга конечных пользователей с использованием языка программирования Python, тестирует, испытывает и оценивает качество таких решений
 Владеет основными библиотеками для выполнения большинства рутинных задач в крупных проектах: ввод-вывод, серверное программирование (FastAPI, Flask, Django REST Framework), применение многопоточности (модуль threading). Самостоятельно участвует в разработке серверных приложений и их поддержке

PL-1.2 Осуществляет выбор инструментов разработки на Python, приемлимых для создания прикладной системы обработки научных данных, машинного обучения и визуализации с заданными требованиями
 Оптимизирует код с использованием библиотек для научных вычислений
 Знает и применяет библиотеки машинного обучения, в том числе глубокого обучения, такие как scikit-learn, PyTorch и TensorFlow

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)				
		4				

Контактная работа, в том числе:	50,2	50,2					
Аудиторные занятия (всего):	48	48					
Занятия лекционного типа	16	16					
Лабораторные занятия	32	32					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	2,2	2,2					
Контроль самостоятельной работы (КСР)	2	2					
Промежуточная аттестация (ИКР)	0,2	0,2					
Самостоятельная работа, в том числе:	21,8	21,8					
Курсовая работа							
Выполнение индивидуальных заданий	20	20					
Реферат							
Подготовка к текущему контролю	1,8	1,8					
Контроль:							
Подготовка к зачету							
Общая трудоемкость	час.	72	72				
	в том числе контактная работа	50,2	50,2				
	зач. ед.	2	2				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые в 4 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Введение и основы событийной модели GUI	10	2		5	3
2.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	10	2		5	3
3.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	11	3		5	3
4.	Tkinter (Python): прототипирование, интеграция с matplotlib	11	3		5	3
5.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	13	3		6	4
6.	Тестирование, упаковка, деплой, CI/CD	13	3		6	4
ИТОГО по разделам дисциплины		68	16		32	20
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		0,2				
Подготовка к текущему контролю		1,8				
Общая трудоемкость по дисциплине		72				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Введение и основы событийной модели GUI	Архитектура GUI-приложений. Событийная модель, рендеринг, очереди сообщений. Паттерны MVC/MVP/MVVM.	ЛР
2.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	Qt Widgets и Qt Quick (QML). Сигналы/слоты, ресурсы, модель-представление, многопоточность/асинхронность.	ЛР
3.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	GTK/gtkmm/PyGObject. GObject, GLib, контейнеры, сигналы, интеграция с Cairo.	ЛР
4.	Tkinter (Python): прототипирование, интеграция с matplotlib	Tkinter: виджеты, Canvas, файловые диалоги, связка matplotlib.	ЛР
5.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	Визуализация: Qt Charts, QCustomPlot, PyQtGraph, VTK (2D/3D, взаимодействие, производительность).	ЛР
6.	Тестирование, упаковка, деплой, CI/CD	Локализация (Qt Linguist, gettext), доступность (a11y), тестирование GUI (pytest-qt, squish, dogtail/sikuli), профилирование.	ЛР
7.	Тестирование, упаковка, деплой, CI/CD	Сборка и упаковка: CMake, qmake, Meson; macdeployqt, windeployqt, MSI (WiX), DMG, AppImage/Flatpak/Snap, PyInstaller/Briefcase.	ЛР
8.	Тестирование, упаковка, деплой, CI/CD	CI/CD для десктопа: GitLab CI, кросс-компиляция, подписывание, автообновления, публикация (Store/Snapcraft/Homebrew).	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП – выполнение курсового проекта, КР – курсовой работы, РГЗ – расчетно-графического задания, Р – написание реферата, Э – эссе, К – коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	Мини-приложение на Qt Widgets (C++): сигналы/слоты, меню, действия.	ЛР
2.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	Аналог на PyQt/PySide (Python): сравнение подходов.	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
3.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	Модель–представление (QAbstractItemModel) + Qt Charts: интерактивный график.	ЛР
4.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	Потоки/асинхронность: QThread/QFuture, индикаторы прогресса.	ЛР
5.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	QML/Qt Quick: панель мониторинга, связка с C++/Python backend.	ЛР
6.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	GTK (gtkmm, C++): Meson, сигналы, контейнеры, диалоги.	ЛР
7.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	GTK (PyGObject, Python): интеграция с Matplotlib.	ЛР
8.	Tkinter (Python): прототипирование, интеграция с matplotlib	Tkinter: визуализатор с matplotlib и диалогами.	ЛР
9.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	PyQtGraph: высокочастотные графики, профилирование отрисовки.	ЛР
10.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	VTK (Python): 3D-визуализация, взаимодействие, экспорт.	ЛР
11.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	VTK (C++): CMake-сборка, пайплайн, виджеты взаимодействия.	ЛР
12.	Тестирование, упаковка, деплой, CI/CD	Пакетирование Python-приложения: PyInstaller (Win/Linux), notarization (макет) на macOS.	ЛР
13.	Тестирование, упаковка, деплой, CI/CD	Пакетирование C++ Qt-приложения: windeployqt/macdeployqt, MSI/DMG/AppImage.	ЛР
14.	Тестирование, упаковка, деплой, CI/CD	Локализация: Qt Linguist и gettext (GTK), переключение языка в рантайме.	ЛР
15.	Тестирование, упаковка, деплой, CI/CD	Тестирование GUI: pytest-qt/squish/dogtail, дымовые/регрессионные тесты.	ЛР
16.	Тестирование, упаковка, деплой, CI/CD	CI/CD: GitLab CI для кросс-сборок, релизные артефакты, автосборки тегов.	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП – выполнение курсового проекта, КР – курсовой работы, РГЗ – расчетно-графического задания, Р – написание реферата, Э – эссе, К – коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Курсовая работа не предусмотрена. В качестве проекта на зачёт студенты разрабатывают кроссплатформенное десктоп-приложение по кейсу индустриального партнёра.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой вычислительных технологий, протокол №7 от 07.05.2025

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

1 Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

2 Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

3 Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

4 Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

5 Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

6 Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

7 Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

8 Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

9 Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

10 Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

11 Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

12 работа в малых группах (команде) – совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

13 проектная технология – индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

14 анализ конкретных ситуаций – анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

15 развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
4	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	32
4	Л	Лекционные занятия	16
Итого			48

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме оценки лабораторных работ к проекта к **зачету**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	ОПК-4.1; ОПК-4.2; УК-2.4; PL-1	<i>ЛР 1. Мини-приложение на Qt Widgets (C++): сигналы/слоты, меню, действия. (отчёт + защита)</i>	–
2.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	ОПК-4.1; ОПК-4.2; УК-2.4; PL-1	<i>ЛР 2. Аналог на PyQt/PySide (Python): сравнение подходов. (отчёт + защита)</i>	–
3.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	ОПК-4.1; УК-2.4; PL-1	<i>ЛР 3. Модель–представление (QAbstractItemModel) + Qt Charts: интерактивный график. (отчёт + защита)</i>	–
4.	Qt (C++/Python): виджеты, QML, сигналы/слоты, потоки	ОПК-4.1; ОПК-4.2; УК-2.4; PL-1	<i>ЛР 4. Потоки/асинхронность: QThread/QFuture, индикаторы прогресса. (отчёт + защита)</i>	–
5.	Qt (C++/Python): виджеты, QML,	ОПК-4.1; ОПК-4.2; УК-2.4; PL-1	<i>ЛР 5. QML/Qt Quick: панель</i>	<i>Проект по кейсу №1</i>

	сигналы/слоты, потоки		мониторинга, связка с C++/Python backend. (отчёт + защита)	
6.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	ОПК-4.1; ОПК-4.2; УК-2.4	ЛР 6. GTK (gtkmm, C++): Meson, сигналы, контейнеры, диалоги. (отчёт + защита)	–
7.	GTK (gtkmm/PyGObject): виджеты, GObject, асинхронность	ОПК-4.1; ОПК-4.2; УК-2.4	ЛР 7. GTK (PyGObject, Python): интеграция с Matplotlib. (отчёт + защита)	–
8.	Tkinter (Python): прототипирование, интеграция с matplotlib	ОПК-4.1; УК-2.3; PL-1	ЛР 8. Tkinter: визуализатор с matplotlib и диалогами. (отчёт + защита)	–
9.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	ОПК-4.1; УК-2.4; PL-1	ЛР 9. PyQtGraph: высокочастотны е графики, профилирование отрисовки. (отчёт + защита)	–
10.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	ОПК-4.1; УК-2.4; PL-1	ЛР 10. VTK (Python): 3D-визуализация, взаимодействие, экспорт. (отчёт + защита)	–
11.	Визуализация данных: Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK	ОПК-4.1; УК-2.4; PL-1	ЛР 11. VTK (C++): CMake-сборка, пайплайн, виджеты взаимодействия. (отчёт + защита)	–
12.	Тестирование, упаковка, деплой, CI/CD	ОПК-4.2; PL-1	ЛР 12. Пакетирование Python-приложен ия: PyInstaller (Win/Linux), notarization (макет) на macOS. (отчёт + защита)	–
13.	Тестирование, упаковка, деплой, CI/CD	ОПК-4.2; PL-1	ЛР 13. Пакетирование C++ Qt-приложения:	–

			<i>windeployqt/macdeployqt, MSI/DMG/AppImage. (отчёт + защита)</i>	
14.	Тестирование, упаковка, деплой, CI/CD	ОПК-4.2; PL-1	<i>ЛР 14. Локализация: Qt Linguist и gettext (GTK), переключение языка в рантайме. (отчёт + защита)</i>	–
15.	Тестирование, упаковка, деплой, CI/CD	ОПК-4.2; PL-1	<i>ЛР 15. Тестирование GUI: pytest-qt/squish/dogtail, дымовые/регрессионные тесты. (отчёт + защита)</i>	–
16.	Тестирование, упаковка, деплой, CI/CD	ОПК-4.2; PL-1	<i>ЛР 16. CI/CD: GitLab CI для кросс-сборок, релизные артефакты, автосборки тегов. (отчёт + защита)</i>	<i>Проект по кейсу №2</i>

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **зачтено**):

УК-2

Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

Использует принципы проектной методологии для решения профессиональных задач

Знает основы проектного управления (цели/результаты, WBS, спринты, метрики), риск-менеджмент; принципы формулирования требований и критериев приёмки; специфику кроссплатформенной разработки (Windows/Linux/macOS), факторы выбора стека GUI (Qt/GTK/Tkinter) и библиотек визуализации.

Умеет декомпозировать постановку на задачи/сроки; вести бэклог/дорожную карту; составлять критерии готовности; аргументировать выбор инструментов под ограничения (производительность, доступность, лицензии, сроки).

Владеет практиками планирования в трекерах (GitLab Issues/Jira), ведением ADR/рисков, релизным календарём и версионированием (SemVer) для desktop-проектов.

Выбирает оптимальный способ решения задач, имеющихся ресурсов и ограничений, оценки рисков на основе проектного инструментария. Знает методы сравнительной оценки альтернатив (multi-criteria, TCO/риски), основные профайлеры и средства диагностики GUI; требования к локализации и доступности (i18n/all y).

Умеет готовить прототипы на Qt/GTK/Tkinter; проводить бенчмаркинг/профилирование (время отклика UI, FPS графики), готовить обоснование выбора (trade-off).

Владеет инструментами измерений (Qt Creator Analyzer, Valgrind, cProfile, PyQtGraph/VTK бенчмарки), приёмами снижения рисков (фича-флаги, PoC, инкрементальные релизы).

ОПК-4

Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности

Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

Знает кроссплатформенные GUI-фреймворки Qt/GTK/Tkinter, их событийные циклы и паттерны (MVC/MVVM); стек визуализации (Qt Charts/QCustomPlot/Matplotlib/PyQtGraph/VTK); принципы локализации/all y; инструменты сборки/упаковки и основы CI для desktop.

Умеет строить функциональные экраны/диалоги; подключать визуализацию (2D/3D) и экспорт в PNG/SVG/PLY; реализовывать неблокирующую обработку данных (потoki/задачи/async); собирать и пакетировать приложение под Windows/Linux/macOS.

Владеет CMake/qmake/Meson, PyInstaller/Briefcase, winderployqt/macdeployqt; средствами профилирования/логирования; практиками автоматической сборки релизов (GitLab CI).

Ориентируется в современных положениях и концепциях прикладного и системного программного обеспечения, архитектуры компьютеров и сетей (в том числе и глобальных), технологии создания и сопровождения программных продуктов и программных комплексов

Знает: практики сопровождения desktop-приложений (логирование/трассировка, автообновления, телеметрия, обратная совместимость), подходы к тестам GUI (smoke/regression).

Умеет: настраивать сбор телеметрии/журналов, локализацию и переключение языка в рантайме; внедрять автообновление и стратегию релизов; писать дымовые/регрессионные тесты GUI.

Владеет инструментами pytest-qt/dogtail (или аналогами), системами доставки обновлений и публикации артефактов, приёмами поддержания качества (код-ревью, линт, статанализ).

ПЛ-1

Способен применять язык программирования Python для решения задач в области ИИ

Владеет основными библиотеками для выполнения большинства рутинных задач в крупных проектах: ввод-вывод, серверное программирование (FastAPI, Flask, Django REST Framework), применение многопоточности (модуль threading). Самостоятельно участвует в разработке серверных приложений и их поддержке

Оптимизирует код с использованием библиотек для научных вычислений

Знает и применяет библиотеки машинного обучения, в том числе глубокого обучения, такие как scikit-learn, PyTorch и TensorFlow

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Практические задания для лабораторных работ

ЛР-1. Qt Widgets (C++)

Задание: окно с меню/toolbar/statusbar; File/Open; чтение CSV → QTableView (через модель), обработка ошибок.

Сдать: исходники, исполняемый файл, README с инструкцией запуска под 2 ОС, отчёт (шаблон в РПД).

Компетенции: ОПК-4.1; ОПК-2.1; УК-2.3.

ЛР-2. PyQt/PySide (Python)

Задание: аналог ЛР-1 на Python; сохранить настройки (QSettings); сравнить подходы C++ vs Python.

Сдать: исходники, отчёт с сравнением (таблица «время/размер/удобство»).

Компетенции: PL-1; ОПК-4.1; УК-2.4.

ЛР-3. Model–View + Qt Charts

Задание: собственная модель (QAbstractItemModel) + связанный интерактивный график.

Сдать: проект + 2 скриншота сценариев взаимодействия.

Компетенции: ОПК-4.1; УК-2.4.

ЛР-4. Потоки/асинхронность

Задание: вынести длительную операцию (парсинг/расчёт) в QThread/QFuture; индикатор прогресса; отмена.

Сдать: проект + замер времени (до/после).

Компетенции: ОПК-4.1; УК-2.4.

ЛР-5. QML/Qt Quick + backend

Задание: панель мониторинга (QML) со связью C++/Python; минимум 3 виджета, 2-сторонняя привязка.

Сдать: проект; короткое видео (до 1 мин).

Компетенции: ОПК-4.1; УК-2.4; PL-1.

Примечание: завершение блока Qt – результаты пойдут в проект по кейсу №1 (зачёт).

ЛР-6. GTK (gtkmm, C++)

Задание: окно со списком/фильтром, диалог редактирования; сборка Meson.

Сдать: исходники + скрипт сборки для Linux/Windows (MSYS2/MinGW).

Компетенции: ОПК-4.1; ОПК-4.2.

ЛР-7. GTK (PyGObject, Python)

Задание: интеграция с Matplotlib; интерактивный график и сохранение в PNG/SVG.

Сдать: код + скриншоты.

Компетенции: ОПК-4.1; PL-1.

ЛР-8. Tkinter (Python)

Задание: визуализатор с Canvas + matplotlib; drag&drop файла.

Сдать: код + видео-демо.

Компетенции: ОПК-4.1; УК-2.3; PL-1.

ЛР-9. PyQtGraph (Python)

Задание: высокочастотные графики, профилирование отрисовки; цели по FPS.

Сдать: код + таблица сравнения FPS по настройкам.

Компетенции: ОПК-4.1; УК-2.4.

ЛР-10. VTK (Python)

Задание: 3D-сцена, манипуляторы, экспорт скриншотов/PLY.

Сдать: проект + 2 изображения.

Компетенции: ОПК-4.1.

ЛР-11. VTK (C++)

Задание: CMake-проект; интерактивные виджеты; запись в STL/VTK.

Сдать: репозиторий с инструкцией.

Компетенции: ОПК-4.1; ОПК-4.2.

ЛР-12. Пакетирование Python-приложения

Задание: PyInstaller (Win/Linux), макет нотаризации macOS; минимизация размера.

Сдать: релизные артефакты + README.

Компетенции: ОПК-4.2; PL-1.

ЛР-13. Пакетирование Qt C++-приложения

Задание: windeployqt/macdeployqt; подготовить MSI/DMG/AppImage.

Сдать: инсталлятор/образ + чек-лист теста установки.

Компетенции: ОПК-4.2.

ЛР-14. Локализация

Задание: Qt Linguist и gettext (GTK), переключение языка в рантайме.

Сдать: PO/TS-файлы + видео с переключением языка.

Компетенции: ОПК-4.2; УК-2.3.

ЛР-15. Тестирование GUI

Задание: дымовые/регрессионные сценарии с pytest-qt/dogtail (≥ 3 сценариев).

Сдать: набор тестов + отчёт о прогонах.

Компетенции: ОПК-4.2; PL-1.

ЛР-16. CI/CD

Задание: GitLab CI для кросс-сборок (Win/Linux), релизные артефакты, автосборки по тегам.

Сдать: .gitlab-ci.yml, скрин артефактов, лог пайплайна.

Компетенции: ОПК-4.2; PL-1.

Примечание: завершение блока CI/CD – результаты идут в проект по кейсу №2 (зачёт).

Пример лабораторной работы

Лабораторная работа №1: Qt Widgets (C++): сигналы/слоты, меню, действия

Тема: основы событийной модели и виджетов в Qt

Цель

Освоить построение базового окна приложения и связывание событий UI с обработчиками.

Оборудование и ПО

ПК (Windows/Linux), Qt 6.x (MSVC/MinGW/Clang), CMake, Git.

Исходные данные

CSV (≥ 100 строк) с 4–6 колонками (пример даёт преподаватель).

Задание

1. Создать проект (CMake) с Qt Widgets; реализовать MainWindow с меню (File/Open, File/Exit), toolbar, статус-баром.
2. Реализовать диалог открытия CSV; загрузить первые 100 строк в QTableView через собственную модель.
3. Обработка ошибок (несуществующий файл/неправильная кодировка); уведомления в статус-баре.
4. Не блокировать UI при чтении – вынести парсинг в QThread/QtConcurrent и показать индикатор прогресса.
5. Подготовить README с инструкцией запуска для двух ОС.

Порядок выполнения

1. Инициализация проекта.
2. Верстка меню/toolbar.
3. Модель и QTableView.
4. Поток/прогресс.
5. Финализация и тесты.

Контрольные вопросы

1. Событийная модель Qt.
2. Сигналы/слоты.
3. Отличие QTableWidget и QTableView.
4. Принципы потоков в Qt.
5. Обработка ошибок.

Отчёт

1. Титульный лист.
2. Цель/задачи.
3. Архитектура решения (схема).
4. Ключевые фрагменты кода.
5. Скриншоты UI.
6. Результаты замеров.
7. Выводы.

Критерии приёма

Зачтено: если проект запускается на двух ОС, реализованы все пункты задания (меню, toolbar, статус-бар, открытие и отображение CSV через собственную модель, обработка ошибок, асинхронность чтения с прогрессом), присутствует README с инструкцией и оформленный отчёт с архитектурой, кодом, скриншотами и выводами.

Не зачтено: если отсутствует или не работает хотя бы один из обязательных элементов задания, не реализована асинхронность или обработка ошибок, нет README или отчёта, либо проект не запускается на одной из указанных ОС.

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Форма: письменная часть + защита проекта по 2 кейсам индустриальных партнёров.

1) Письменная часть (45-60 мин.) – типовой вариант билета

Короткие ответы (любой 3 из 5):

1. Отличия событийных циклов Qt/GTK/Tkinter.
2. Модель-представление в Qt и её преимущества.
3. Асинхронность/поток в Qt и влияние на UX.
4. Подходы к локализации (Qt Linguist vs gettext).
5. Сравнение способов пакетирования (PyInstaller/Briefcase vs windeployqt/macdeployqt).

Задача на проектирование (1 из 2):

- А) Спроектировать окно визуализации с панелью параметров и сохранением настроек.
- В) Спроектировать пайплайн CI: матрица сборок Win/Linux, релиз артефактов, проверка тестов GUI.

Мини-практика (1 из 2):

- А) Привести сигнатуры сигналов/слотов и связать их для меню «Open».
- В) Дописать фрагмент .gitlab-ci.yml для публикации артефактов релиза.

2) Защита проекта по кейсам (10-15 мин. на команду)

Кейс №1 (после блока Qt): «АВА ЛАБ – Fastboard desktop-клиент» (NLP-запросы → отчёты; визуализация; экспорт PDF/PNG; локализация; релизы Win/Linux).

Кейс №2 (после CI/CD): «СвязьРесурс-Кубань – desktop-генератор документов» (параметрическая форма; предпросмотр DOCX/PDF; локализация; релизы 2 ОС; CI).

Перечень компетенций (части компетенции), проверяемых оценочным средством:
УК-2.3; УК-2.4; ОПК-2.1; ОПК-4.2; ОПК-6.1; ОПК-6.2; PL-1.

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет. Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из заданий и результатов текущего контроля.

Форма проведения зачета: письменно.

Преподавателю предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи зачета заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и проекта.

Критерии оценки:

Зачтено – выполнено 60% лабораторных работ (на оценку зачтено) и выполнен проект на 60% или более.

Не зачтено – выполнено менее 60% лабораторных работ или выполнено менее 60% проекта.

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

Условия применения:

- Курс рассчитан на студентов 2 курса.
- Наличие персональных компьютеров под управлением Windows и/или Linux (допускается macOS).
- Установленные среды разработки и инструменты: компилятор C++ (MSVC / MinGW / GCC / Clang); Python версии 3.9 и выше; Qt 6.x (Qt Widgets, Qt Quick/QML), PyQt или PySide; GTK (gtkmm или PyGObject); Tkinter (в составе стандартной поставки Python).
- Доступ к средам разработки Qt Creator и/или Visual Studio Code.
- Использование систем контроля версий (Git) и репозитория (GitLab/GitHub).

Цели, задачи и ожидаемые результаты:

Цели:

- Обеспечить студентов унифицированной и воспроизводимой средой разработки кроссплатформенных десктопных приложений.
- Сформировать практические навыки настройки и сопровождения рабочего окружения для разработки GUI-приложений.

Задачи преподавателя:

- Подготовка инструкций по установке и настройке инструментов разработки под разные операционные системы.
- Формирование типовой структуры проектов (CMake/qmake/Meson, виртуальные окружения Python).
- Подготовка тестовых заготовок проектов для лабораторных работ (Qt, GTK, Tkinter).

Ожидаемые результаты студентов:

- Умение самостоятельно настраивать рабочее окружение для разработки десктопных приложений.
- Навыки работы с инструментами сборки и запуска приложений под различными ОС.
- Понимание принципов организации вычислительного процесса и жизненного цикла GUI-приложения.

Порядок реализации:

Настройка окружения:

- Для C++-разработки: установка Qt 6.x, настройка CMake/qmake, проверка сборки тестового Qt-проекта.
- Для Python-разработки: настройка виртуального окружения (venv/conda), установка PyQt/PySide, PyGObject, Matplotlib, PyQtGraph.
- Настройка Git-репозитория и базовой структуры проекта.

Шаблоны работ:

Типовая структура проекта лабораторной работы:

1. Описание задачи и требований
2. Структура проекта и используемые технологии
3. Реализация пользовательского интерфейса
4. Логика обработки данных и событий
5. Тестирование и отладка

6. Выводы

Порядок проверки корректности:

Чек-лист работоспособности окружения:

- Успешная сборка и запуск демонстрационного GUI-приложения
- Корректная работа среды разработки
- Отсутствие ошибок при подключении библиотек
- Возможность запуска проекта минимум под одной ОС

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Наличие настроенного рабочего окружения у всех обучающихся.
- Доступ к методическим материалам и шаблонам лабораторных работ.
- Использование системы контроля версий для сдачи и проверки работ.

Цели, задачи и ожидаемые результаты:

Цели:

- Закрепить теоретические знания по разработке кроссплатформенных GUI-приложений.
- Сформировать устойчивые практические навыки проектирования и реализации пользовательских интерфейсов.

Задачи преподавателя:

- Разработка пошаговых методических указаний к лабораторным работам.
- Подготовка типовых заданий по Qt, GTK и Tkinter.
- Организация проверки, защиты и обратной связи по выполненным работам.

Ожидаемые результаты студентов:

- Умение разрабатывать десктопные приложения с использованием современных GUI-фреймворков.
- Навыки применения событийной модели, сигналов и слотов, асинхронности.
- Способность оформлять отчёты и аргументированно защищать выполненные решения.

Порядок реализации:

План лабораторных работ:

ЛР1. Мини-приложение на Qt Widgets (C++): сигналы и слоты, меню, действия.

ЛР2. Реализация аналогичного приложения на PyQt/PySide (Python): сравнительный анализ подходов C++ и Python.

ЛР3. Архитектура модель–представление (QAbstractItemModel) и визуализация данных с использованием Qt Charts.

ЛР4. Асинхронность и многопоточность в Qt-приложениях: QThread и QFuture, индикаторы прогресса.

ЛР5. Разработка пользовательского интерфейса на QML/Qt Quick и интеграция с C++/Python-backend.

ЛР6. Разработка GUI-приложения на GTK (gtkmm, C++): виджеты, сигналы, контейнеры, сборка Meson.

ЛР7. Разработка GUI-приложения на GTK (PyGObject, Python) с интеграцией Matplotlib.

ЛР8. Прототипирование десктопного приложения с использованием Tkinter и matplotlib.

- ЛР9. Высокопроизводительная визуализация данных с использованием PyQtGraph и анализ производительности.
- ЛР10. Трёхмерная визуализация данных с использованием VTK (Python).
- ЛР11. Разработка 3D-приложения с использованием VTK (C++): CMake-сборка и пайплайн визуализации.
- ЛР12. Пакетирование Python-приложения: PyInstaller для Windows и Linux, основы notarization для macOS.
- ЛР13. Пакетирование Qt-приложения на C++: windeployqt, macdeployqt, формирование MSI/DMG/AppImage.
- ЛР14. Локализация десктопных приложений: Qt Linguist и gettext, переключение языка в рантайме.
- ЛР15. Тестирование графических интерфейсов: pytest-qt, dogtail, дымовые и регрессионные тесты.
- ЛР16. Настройка CI/CD для кроссплатформенных десктопных приложений: автоматические сборки и релизные артефакты.

Пример индивидуального задания (ЛР4):

Задача: Реализовать длительную вычислительную операцию в Qt-приложении с использованием многопоточности без блокировки пользовательского интерфейса.

Критерии оценки: корректность реализации асинхронной обработки, стабильность работы UI, обоснованность архитектурного решения

Контрольные вопросы:

- Особенности событийной модели Qt
- Назначение сигналов и слотов
- Принципы организации потоков в GUI-приложениях
- Причины блокировки интерфейса и способы их устранения

Критерии оценки:

Зачтено: лабораторная работа выполнена в полном объёме, приложение работоспособно, студент демонстрирует понимание применённых технологий и может обосновать принятые решения.

Не зачтено: работа выполнена не полностью, приложение неработоспособно либо студент не ориентируется в используемых механизмах и архитектуре решения.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

- Курс рассчитан на студентов 2 курса.
- Время на выполнение проекта — 10–15 часов (индивидуально или в команде до 3 человек).
- Наличие базовых навыков работы с C++ и/или Python.

Цели, задачи и ожидаемые результаты:

Цели:

- Применить полученные знания для разработки законченного кроссплатформенного десктопного приложения.
- Сформировать навыки командной работы и проектного подхода.

Задачи преподавателя:

- Формирование тем проектов, соответствующих содержанию дисциплины и уровню студентов.
- Подготовка технических заданий и критериев оценки.
- Организация промежуточного контроля и итоговой защиты проектов.

Ожидаемые результаты студентов:

- Опыт проектирования и реализации GUI-приложений под несколько ОС.
- Умение оформлять проектную и пользовательскую документацию.
- Навыки презентации и защиты результатов работы.

Порядок реализации:**Примеры проектов:**

- Кроссплатформенный визуализатор данных
- Десктоп-клиент для генерации отчётов
- GUI-приложение для мониторинга параметров системы
- Интерактивное учебное приложение с визуализацией

Типовое техническое задание проекта:

- Реализовать пользовательский интерфейс на Qt/GTK/Tkinter
- Обеспечить кроссплатформенный запуск
- Реализовать визуализацию данных
- Подготовить сборку приложения под две ОС
- Оформить документацию и презентацию

Критерии оценки:

Зачтено: проект соответствует ТЗ, демонстрирует корректную архитектуру и работоспособность.

Не зачтено: проект не соответствует базовым требованиям или не запускается.

Порядок проверки корректности:

- Соответствие техническому заданию
- Качество архитектурных решений
- Работоспособность приложения
- Качество презентации и отчётных материалов

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**5.1 Основная литература:**

1. Титов А. Н., Тагиева Р. Ф. Введение в Tkinter: разработка графических интерфейсов в Python: учебно-методическое пособие. — Казань: КНИТУ, 2023. — 100 с. [Электронный ресурс] URL: <https://biblioclub.ru/index.php?id=714011&page=book> (дата обращения: 25.08.2025).
2. Титов А. Н., Тагиева Р. Ф. Визуализация данных в Python: работа с библиотекой Seaborn: учебно-методическое пособие. — Казань: КНИТУ, 2023. — 144 с. [Электронный ресурс.] URL: https://biblioclub.ru/index.php?contrast=0&id=714010&lang=ru&page=book_red&razdel=276 (дата обращения: 25.08.2025).
3. Титов А. Н., Тагиева Р. Ф. Интерактивная визуализация данных: работа с библиотекой Plotly: учебно-методическое пособие. — Казань: КНИТУ, 2023.

- [Электронный ресурс.] URL: <https://biblioclub.ru/index.php?id=714008&page=book> (дата обращения: 25.08.2025).
4. Титов А. Н., Тагиева Р. Ф. Визуализация данных в Python. Работа с библиотекой Matplotlib: учебно-методическое пособие. — Казань: КНИТУ, 2022. — 92 с. [Электронный ресурс.] URL: <https://biblioclub.ru/index.php?id=702231&page=book> (дата обращения: 25.08.2025).
 5. Титов А. Н., Тагиева Р. Ф. Основы работы с библиотекой NumPy: учебно-методическое пособие. — Казань: КНИТУ, 2024. [Электронный ресурс.] URL: <https://biblioclub.ru/index.php?id=721178&page=book> (дата обращения: 25.08.2025).
 6. Рамальо Л. Python — к вершинам мастерства: практическое пособие. 2-е изд. — Москва: ДМК Пресс, 2022. — 899 с. [Электронный ресурс.] URL: https://biblioclub.ru/index.php?contrast=1&id=703928&lang=ru&page=book_red&razdel=165 (дата обращения: 25.08.2025).
 7. Лутц М. Изучаем Python. — СПб.: Диалектика, 2019. — 832 с. — ISBN 978-5-907144-85-0
 8. Любанович Б. Простой Python. Современный стиль программирования. — СПб.: Питер, 2021. — 592 с. — ISBN 978-5-4461-1456-6
 9. Шоу Э. Внутри CPython: гид по интерпретатору Python. — СПб.: Питер, 2023. — 352 с. — ISBN 978-5-4461-1925-7
 10. Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow / пер. А. А. Слинкина. — М.: ДМК, 2023. — 688 с. — ISBN 978-5-93700-123-7
 11. Рашка С. Машинное обучение с PyTorch и Scikit-Learn. — М.: ДМК, 2023. — 790 с. — ISBN 978-5-93700-145-9
 12. ПНСТ 838-2023 Искусственный интеллект. Структура описания систем искусственного интеллекта, использующих машинное обучение
 13. ГОСТ Р 71752-2024 Искусственный интеллект. Техническое задание. Требования к содержанию

5.2 Дополнительная литература:

1. Blanchette J., Summerfield M. C++ GUI Programming with Qt 5/6. Addison-Wesley.
2. Shlee M. Mastering Qt 5/6. Packt.
3. Matplotlib 3.x Documentation (official).
4. VTK User's Guide and API Reference (Kitware).
5. GTK 4.0 API Reference, PyGObject Tutorial, gtkmm documentation.
6. Tkinter Official Reference (Python docs).

5.3 Периодические издания:

1. IEEE Transactions on Big Data – научные статьи по обработке больших данных.
2. Journal of Big Data (SpringerOpen) – открытый журнал с исследованиями в области Big Data.
3. Big Data Research (Elsevier) – публикации по анализу, управлению и визуализации данных.
4. Data Science Journal (CODATA) – междисциплинарные исследования данных.
5. ACM Transactions on Knowledge Discovery from Data (TKDD) – методы извлечения знаний из больших данных.
6. <https://openreview.net/forum?id=FMMF1a9ifL>
7. <https://openreview.net/forum?id=ElUrNM9U8c#discussion>
8. <https://openreview.net/forum?id=JoO6mtCLHD>
9. <https://aclanthology.org/2024.findings-emnlp.760/>
10. <https://aclanthology.org/2020.coling-main.588/>
11. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30

12. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10

13. <https://aclanthology.org/2024.findings-naacl.288/>

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prlib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

Бесплатные образовательные ресурсы

1. Jupyter Notebook – интерактивные вычисления
2. Visual Studio Code – редактор кода с поддержкой Python
3. Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

1. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;

10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.uceba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--plai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ
<http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ
<http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций
<http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru;>
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

5.5 Публикации конференций А*

1. Tim Menzies, Oussama Elrawas, Jairus Hihn, Martin Feather, Ray Madachy, and Barry Boehm. 2007. The business case for automated software engineering. In Proceedings of the 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE '07). Association for Computing Machinery, New York, NY, USA, 303–312. <https://doi.org/10.1145/1321631.1321676>
2. Farzana Ahamed Bhuiyan and Akond Rahman. 2021. Characterizing co-located insecure coding patterns in infrastructure as code scripts. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20). Association for Computing Machinery, New York, NY, USA, 27–32. <https://doi.org/10.1145/3417113.3422154>
3. Michael Hilton, Timothy Tunnell, Kai Huang, Darko Marinov, and Danny Dig. 2016. Usage, costs, and benefits of continuous integration in open-source projects. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE '16). Association for Computing Machinery, New York, NY, USA, 426–437. <https://doi.org/10.1145/2970276.2970358>

6. Методические указания для обучающихся по освоению дисциплины (модуля)

В рамках курса предусмотрено проведение лекционных занятий, на которых излагается систематизированный материал по разработке кроссплатформенных десктопных приложений. На лекциях рассматриваются основные концепции, современные подходы и инструменты, используемые для создания приложений, работающих на различных операционных системах. После каждой лекции рекомендуется выполнение практических заданий для закрепления изученного материала и освоения ключевых технологий.

Лабораторные занятия направлены на практическое освоение методов и инструментов разработки кроссплатформенных десктопных приложений. В ходе лабораторных работ студенты реализуют основные компоненты пользовательских

интерфейсов, обеспечивают взаимодействие с файловой системой, используют средства для работы с сетью и базами данных. Для выполнения заданий используются популярные фреймворки и среды разработки, такие как Qt, Electron или аналогичные.

В самостоятельной работе студентам рекомендуется изучать официальную документацию к выбранным фреймворкам и инструментам, а также дополнительную литературу по проектированию и оптимизации десктопных приложений.

Важной частью курса является самостоятельная проектная работа, в рамках которой студент разрабатывает завершённое кроссплатформенное десктопное приложение для решения конкретной задачи (кейса), предложенной промышленными партнёрами или преподавателем. Допускается выполнение проектов индивидуально или в командах до 3-х человек.

Кейс №1 (АВА ЛАБ – Fastboard desktop-клиент):

Задание: кроссплатформенный desktop-клиент (Qt/PyQt/GTK) для естественно-языковых запросов к Fastboard; визуализация отчётов (таблицы/графики); экспорт PDF/PNG; локализация; релизы под 2 ОС; пайплайн CI.

Минимум к зачёту: UI + визуализация + локализация + пакеты Win/Linux + артефакты CI.

Кейс №2 (СвязьРесурс-Кубань – desktop-генератор документов):

Задание: приложение (Tkinter/PyQt) с параметрической формой, предпросмотр DOCX/PDF, логирование; локализация; релизы 2 ОС; CI-сборка.

Минимум к зачёту: работоспособный предпросмотр и экспорт; две сборки; CI + инструкция пользователя.

Для студентов с ограниченными возможностями здоровья предусмотрены дополнительные индивидуальные консультации, на которых преподаватель подробно разъясняет сложные аспекты дисциплины, помогает адаптировать практические задания и обеспечивает специальные условия для освоения методов работы с системами искусственного интеллекта. Индивидуальный подход позволяет таким студентам полноценно участвовать в учебном процессе и достигать требуемых результатов обучения.

7. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется промышленным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб

2	Виртуальная машина с GPU	Аренда публичного IP	1		Шт
		Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 ГБ RAM ОС Ubuntu_24.04	1	1	Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
3	K8S	Аренда публичного IP	1		Шт
		Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб
4	ML Inference Instance Type GPU	Аренда публичного IP	1		Шт
		Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML- моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
		5	LLM	Токены GigaChat 2 Max	50
Токены Embeddings	400				Млн. Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения,

	аттестация	компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.