

Аннотация рабочей программы дисциплины

Б1.О.27 «Кроссплатформные десктоп приложения»

Курс 2 Семестр 4 Количество з.е. 2

Объем трудоемкости: 2 зачетных единиц (72 ч., из них – 48 час. аудиторной нагрузки: лекционных 16 ч., лабораторных работ – 32 ч., 21,8 часов самостоятельной работы, 2 часов КСР, 0,2 часа ИКР.), форма контроля – зачет.

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.

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).

Место дисциплины в структуре ООП ВО:

Дисциплина «Кроссплатформные 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.

Результаты обучения (знания, умения, опыт, компетенции):

Содержание и структура дисциплины:

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

УК-2	<i>Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений</i>
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/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-2	<i>Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности</i>
ОПК-2.1	Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ Знает основы ООП и событийно-ориентированной модели; архитектуру ЯП C++/Python, модели памяти и конкурентности; принципы межъязыкового взаимодействия (Python↔C++), устройства тулкетов Qt/GTK/Tkinter и стека визуализации (Qt Charts, QCustomPlot, Matplotlib, PyQtGraph, VTK); средства сборки/упаковки (CMake/qmake/Meson; PyInstaller/Briefcase; windeployqt/macdeployqt); особенности отечественных ОС/дистрибутивов. Умеет проектировать и реализовывать модульные настольные приложения; подключать и конфигурировать библиотеки визуализации; применять

	<p>многопоточность/асинхронность без блокировки UI; интегрировать Python-модули с C++-ядром.</p> <p>Владеет разработческой средой (Qt Creator/VS Code), системами сборки, дебаггерами/профайлерами, менеджерами окружений (venv/conda), практиками кросс-платформенной сборки и запуском на отечественных ОС.</p>
ОПК-4	<i>Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла</i>
ОПК-4.2	<p>Знает принципы сбора и анализа информации, создания информационных систем на стадиях жизненного цикла</p> <p>Знает жизненный цикл desktop-ПО (от прототипа до поддержки); состав и стандарты пакета документов (ТЗ/ПМИ/РК, пользовательская/установочная документация, релиз-ноуты); требования к UI-гайдлайнам, локализации/доступности; основы подписывания кода и дистрибуции (MSI/DMG/AppImage/Flatpak/Snap).</p> <p>Умеет оформлять требования и приёмочные критерии; готовить пользовательскую/эксплуатационную документацию; формировать релизные артефакты под 2+ ОС; организовывать обратную связь (отчёты об ошибках/телеметрию) и трассировку требований→задач→релизов.</p> <p>Владеет «documentation-as-code» (Markdown/AsciiDoc, Sphinx/Doxygen), шаблонами релизных заметок/чейнджлогов, практиками версионирования/ветвления (Git-flow), инструментами локализации (Qt Linguist/gettext) и средствами публикации пакетов.</p>
ОПК-6	<i>Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности</i>
ОПК-6.1	<p>Аргументировано применяет современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения</p> <p>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.</p> <p>Умеет строить функциональные экраны/диалоги; подключать визуализацию (2D/3D) и экспорт в PNG/SVG/PLY; реализовывать неблокирующую обработку данных (поток/таски/async); собирать и пакетировать приложение под Windows/Linux/macOS.</p> <p>Владеет CMake/qmake/Meson, PyInstaller/Briefcase, windeployqt/macdeployqt; средствами профилирования/логирования; практиками автоматической сборки релизов (GitLab CI).</p>
ОПК-6.2	<p>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</p> <p>Знает практики сопровождения desktop-приложений (логирование/трассировка, автообновления, телеметрия, обратная совместимость), подходы к тестам GUI (smoke/regression).</p>

	<p>Умеет настраивать сбор телеметрии/журналов, локализацию и переключение языка в рантайме; внедрять автообновление и стратегию релизов; писать дымовые/регрессионные тесты GUI.</p> <p>Владеет инструментами pytest-qt/dogtail (или аналогами), системами доставки обновлений и публикации артефактов, приёмами поддержания качества (код-ревью, линт, статанализ).</p>
PL-1 П	Способен применять язык программирования Python для решения задач в области ИИ
PL-1.1	<p>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</p> <p>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</p>
PL-1.2	<p>Осуществляет выбор инструментов разработки на Python, приемлимых для создания прикладной системы обработки научных данных, машинного обучения и визуализации с заданными требованиями</p> <p>Оптимизирует код с использованием библиотек для научных вычислений</p> <p>Знает и применяет библиотеки машинного обучения, в том числе глубокого обучения, такие как scikit-learn, PyTorch и TensorFlow</p>

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 4 семестре (очная форма)

№	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				

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
Общая трудоемкость по дисциплине		72				

Примечание: Л – лекции, КСР – контрольные и самостоятельные работы, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Курсовые проекты или работы.

Не предусмотрены учебным планом

Вид аттестации: ЛР, проект по кейсами индустриальных партнеров, зачет.

Автор В.И.Шиян, ст. преп. КВТ

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