

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
Государственного университета

А.Д.Король

26 мая 2025 г.

Регистрационный № 3180/б.



ПРОГРАММИРОВАНИЕ МОБИЛЬНЫХ И ВСТРАИВАЕМЫХ СИСТЕМ

Учебная программа учреждения образования по учебной дисциплине для
специальности:

6-05-0533-11 Прикладная информатика

2025 г.

Учебная программа составлена на основе ОСВО 6-05-0533-11-2023, учебных планов БГУ: № 6-5.3-59/03 от 15.05.2023, № 6-5.3-59/04 от 15.05.2023, № 6-5.3-59/05 от 15.05.2023, № 6-5.3-59/11ин. от 31.05.2023, № 6-5.3-59/12ин. от 31.05.2023, № 6-5.3-59/13ин. от 31.05.2023.

СОСТАВИТЕЛЬ:

М.И.Давидовская, старший преподаватель кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета

РЕЦЕНЗЕНТЫ:

И.А.Король, начальник лаборатории, заместитель директора Государственного предприятия «Центр систем идентификации» Национальной академии наук Беларуси, кандидат физико-математических наук, доцент;

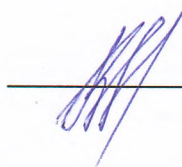
Т.В.Соболева, доцент кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета, кандидат физико-математических наук, доцент

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

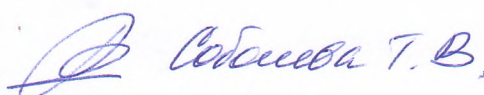
Кафедрой технологий программирования БГУ
(протокол № 17 от 15.05.2025)

Научно-методическим советом БГУ
(протокол № 10 от 22.05.2025)

Заведующий кафедрой



А.Н.Курбацкий



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины – подготовка специалиста по проектированию и разработке платформ-зависимых и кроссплатформенных мобильных приложений для операционных систем на основе ядра Linux и приложений для встраиваемых систем. Формирование и развитие системы знаний по теоретическим и практическим аспектам проектирования и разработки мобильных приложений и встраиваемых систем, получаемых будущими специалистами в процессе учёбы и необходимых им в дальнейшем для успешной профессиональной деятельности в области промышленной разработки.

Задачи учебной дисциплины:

1. Изучить инструментарий для проектирования и разработки мобильных приложений для ОС Android и других мобильных ОС на основе ядра Linux;
2. Изучить инструментарий для проектирования, моделирования и разработки приложений для встраиваемых систем.
3. Изучить синтаксис языка программирования Kotlin и Dart;
4. Научить эффективно применять полученные навыки проектирования и разработки приложений в профессиональной деятельности;
5. Сформировать навыки проектирования архитектуры мобильных приложений и встраиваемых систем.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к модулю «Прикладное программирование» компонента учреждения образования.

Организационные, управленческие и производственные процессы в любой отрасли насыщены вычислительными устройствами широкого спектра и автоматизируются за счёт применения многокомпонентных программных систем. Как следствие, повышаются требования к уровню компетентности выпускников вузов в области разработки программного обеспечения. Современное программное обеспечение – это сложная система, состоящая из взаимосвязанных программ, которые могут быть представлены мобильными приложениями, встраиваемыми системами, веб-приложениями и др. Без теоретических знаний и практических навыков разработки мобильных и встраиваемых программ специалисту по разработке программного обеспечения невозможно эффективно решать профессиональные задачи. Наиболее востребованными навыками являются проектирование архитектуры, разработка мобильных и встраиваемых приложений, применение современных библиотек и SDK для мобильных и встраиваемых операционных систем, сборка и тестирование приложений, моделирование и симуляция встраиваемых программных средств.

Связи с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др. Содержание учебной программы соответствует уровню подготовки студентов к изучению дисциплины «Программирование мобильных и

встраиваемых систем» и основывается на следующих учебных дисциплинах: «Основы и методология программирования», «Разработка кросс-платформенных приложений», «Промышленное программирование», «Операционные системы», «Технологии программирования для мобильных приложений».

Знания, умения и навыки, полученные при изучении дисциплины «Программирование мобильных и встраиваемых систем», будут востребованы при освоении дисциплин «Проектирование человеко-машинных интерфейсов», «Непрерывное интегрирование и сборка программного обеспечения» и других дисциплин модуля «Прикладное программирование», дисциплин профилизации «Информационная безопасность мобильных приложений», «Моделирование и программирование робототехнических устройств».

Требования к компетенциям

Освоение учебной дисциплины «Программирование мобильных и встраиваемых систем» должно обеспечить формирование следующих универсальных, базовых профессиональных и специализированных компетенций:

Универсальные компетенции:

Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации.

Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

Базовые профессиональные компетенции:

Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного подходов и иных парадигм программирования.

Специализированные компетенции:

Использовать программные средства и технологии для создания прикладного программного обеспечения.

В результате освоения учебной дисциплины студент должен:

знать:

- методы проектирования архитектуры мобильных и встраиваемых программных систем;

- методы проектирования интерфейса пользователя мобильных и встраиваемых приложений;

- методы разработки мобильных и встраиваемых программных средств;

- методы тестирования мобильных и встраиваемых приложений;

уметь:

- анализировать требования, предъявляемые к мобильным приложениям и встраиваемым программным системам;
 - составлять техническое задание на разработку мобильного или встраиваемого приложения;
 - проектировать интерфейс пользователя мобильных и встраиваемых приложений;
 - проектировать тесты для оценки качества мобильных и встраиваемых программ;
 - моделировать процессы во встраиваемых системах;
 - разрабатывать мобильные и встраиваемые приложения;
- иметь навык:
- создания мобильных приложений для ОС Android, ОС Аврора и других ОС на основе ядра Linux;
 - разработки приложений для встраиваемых вычислительных систем;
 - проектирования архитектуры, пользовательского интерфейса мобильных и встраиваемых приложений.

Структура учебной дисциплины

Дисциплина изучается в 6 семестре. В соответствии с учебным планом всего на изучение учебной дисциплины «Программирование мобильных и встраиваемых систем» отведено для **очной формы** получения высшего образования – 108 часов, в том числе 68 аудиторных часов, лекции – 34 часа, лабораторные занятия – 34 часа. **Из них:**

– лекции – 34 часа, лабораторные занятия – 30 часов, управляемая самостоятельная работа (УСР) – 4 часа.

Трудоёмкость учебной дисциплины составляет 3 зачётные единицы.

Форма промежуточной аттестации – зачёт, экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Программирование и проектирование мобильных приложений для ОС Android

Тема 1.1. Введение в разработку мобильных приложений для ОС Android

Операционная система Android, её возможности. Структура и архитектура ОС Android.

Технологии мобильной разработки на языке программирования Java. Android SDK. Android NDK. Java-интерфейсы. Язык программирования Kotlin для мобильной разработки. Технология Kotlin Multiplatform для платформ-зависимой разработки.

Инструментарий для мобильной разработки. Запуск и отладка приложений. Эмуляторы. Форматы пакетов приложений. Запуск Android-приложений в окружении других операционных систем.

Классификация мобильных приложений. Архитектура мобильных Android-приложений и основные компоненты.

Тема 1.2. Проектирование Android-приложений: ресурсы и многоязычность

Структура проекта и его различные представления. Манифест приложения. Элементы манифеста и их атрибуты.

Иерархия основных классов Android SDK. Понятие контекста. Представление View и его назначение. Класс Intent. Передача сообщений (намерений). Виды намерений и их характеристики. Операция (Activity), её жизненный цикл и методы жизненного цикла. Состояния операции и отслеживание изменений. Сервисы и их типы. Класс Service, основные методы и жизненный цикл. Класс ContentProvider для управления доступом к хранилищу и его методы. Приёмники широковещательных сообщений. Трансляция. Класс BroadcastReceiver и его методы.

Ресурсы, их создание и применение. Типы ресурсов и их определение. Проектирование меню приложения. Альтернативные ресурсы. Локализация приложения. Компиляция ресурсов.

Разметка экрана мобильного приложения и способы построения пользовательского интерфейса (User Interface – UI). Императивный и декларативный способ создания UI. Декларативный способ создания пользовательского интерфейса в ОС Android. Классические макеты XML, их виды и характеристики. Библиотека Jetpack Compose и её возможности для проектирования мобильного приложения. Слои и компоненты Jetpack Compose для организации структуры экрана.

Тема 1.3. Основы многозадачного мобильного интерфейса

Навигация в приложениях Android. Стек операций и задачи. Управление режимами запуска операции с помощью атрибутов в файле манифеста и флагов намерений. Обработка стека.

Фрагменты (Fragments), динамическая работа и взаимодействие Fragments с Activity. Жизненный цикл фрагментов. Менеджер фрагментов.

Рекомендации по проектированию адаптивного интерфейса мобильного приложения. Основные элементы ввода и управления. Многооконные приложения. Работа с диалоговыми окнами. Стандартные классы диалоговых окон и списков. Обработка событий и использование адаптеров. Особенности разработки приложения, содержащего несколько активностей.

Средства проектирования многооконного интерфейса в библиотеке Jetpack Compose. Управление окнами. Навигация между экранами. Модальные диалоговые окна. Виды меню и их проектирование. Динамические списки.

Многопоточность и библиотека `java.util.concurrent`. Параллелизм в ОС Android. Потокбезопасность в приложениях Android. Обработка параллелизма с использованием обратных вызовов. Организация многопоточного выполнения с использованием Kotlin Coroutines. Каналы и передача потоков данных. Асинхронные потоки данных Kotlin Flows. Сравнение механизмов реализации потоков данных в Kotlin.

Тема 1.4. Тестирование мобильных приложений

Введение в тестирование мобильных приложений. Особенности экосистемы Android как объекта тестирования. Основные типы тестирования Android-приложений. Инструменты и среды тестирования. Облачные сервисы для тестирования (Firebase Test Lab, BrowserStack, Sauce Labs). Сборка и автоматизация тестирования приложения. Процесс тестирования Android-приложения. Типичные проблемы и лучшие практики тестирования мобильных приложений.

Тема 1.5. Android I/O: от жестов до позиционирования

Отличительные особенности смартфонов. Сенсорное (touch) управление. Касания, жесты и их обработка. Обработка множественных касаний.

Инструменты 2D и 3D графики в Android. Работа с изображениями и мультимедиа. Камера. Использование системного приложения для получения изображения.

Сенсоры и датчики в Android. Акселерометр, датчик ориентации и компас: регулировка и программные функции. Обработка поворота устройства. Взаимодействие с системами позиционирования. Использование глобальных систем позиционирования и беспроводных сетей для определения местоположения. Использование картографических сервисов.

Тема 1.6. Работа с данными

Хранение данных в Android. Сохранение состояния и настроек приложения. Работа с Shared Preferences и DataStore. Внутреннее и внешнее хранилище. Права доступа к файлам. Работа с файлами. Использование

статических файлов как ресурсов. Управление базами данных в Android. Хранение данных в SQLite. Метод query. Условие, сортировка, группировка. Запросы и их типы. Использование контент-провайдера. Метод rawQuery. Транзакции в SQLite. Обновление базы данных в SQLite. Использование баз данных Room и Realm в мобильной разработке.

Библиотеки для клиент-серверного взаимодействия. Форматы обмена данными. Синхронизация данных и службы синхронизации.

Раздел 2. Проектирование приложений для мобильных ОС Linux

Тема 2.1. Введение в разработку приложений для мобильных ОС Linux

Мобильные операционные системы на основе ядра Linux. Операционная система Аврора, её история и основные характеристики. Архитектура и структура ОС Аврора. Установка и развёртывание ОС. Основные подходы к разработке мобильных приложений для ОС Аврора. Инструментарий разработчика и кроссплатформенные технологии.

Тема 2.2. Технологии QT и основы QML для разработки мобильных приложений

Технологии QT. Концепции QT. Модули QT. QT API для мобильных устройств. Аврора SDK, состав и установка. Структура проекта приложения и его типы. Архитектура пользовательского интерфейса приложений и декларативное описание с помощью QML. Инструменты сборки и запуска. Дополнительные возможности Аврора SDK.

Технология разработки пользовательского интерфейса QT Quick. Язык разметки QML и основы вёрстки пользовательского интерфейса. Интерпретатор QML. Базовые и сенсорные компоненты QTQuick. Базовые и основные визуальные типы QML. Позиционирование элементов. Контейнеры и разметки. Программирование логики и обработка событий. Преобразование элементов и анимация.

Тема 2.3. Проектирование пользовательского интерфейса в ОС Аврора и работа с сервисами

Интерфейс ОС Аврора. Окно приложения. Текстовые элементы и элементы управления. Контейнеры, меню и диалоговые окна. Рекомендации по разработке. Локализация приложения.

Объектная модель QT. Механизм сигналов и слотов. Экспорт C++ классов в QML. Системные API. Хранение данных и работа с базой данных SQLite. Графика и мультимедиа. Определение местоположения и карты. Работа с сетью и датчиками. Тестирование и валидация приложения.

Тема 2.4. Программное средство Flutter для разработки мобильных приложений

Язык программирования Dart. Типы данных и основные инструкции. Функции и классы. Операторы. Асинхронное программирование.

Программное средство Flutter, его возможности и характеристика. Архитектура и основные компоненты Flutter. Установка Flutter и расширений. Структура проекта и экосистема пакетов. Концепции виджетов и состояний. Основные виджеты и разметки. Темы, навигация и управление состояниями. Работа с данными. Графика и анимация. Flutter SDK для ОС Аврора и проектирование мобильных приложений.

Тема 2.5. Мультиплатформенные решения в мобильной разработке

Классификация решений для мультиплатформенной мобильной разработки. Кроссплатформенная разработка на примере Kotlin Multiplatform (Compose Multiplatform). Архитектура Compose Multiplatform. Проектирование мобильных приложений для ОС Аврора на основе Kotlin Multiplatform.

Раздел 3. Проектирование, моделирование и реализация приложений для встраиваемых систем

Тема 3.1. Введение во встраиваемые вычислительные системы

Система реального времени и её виды. Особенности систем реального времени. Встроенная (встраиваемая) система. Категории встраиваемых систем. Технические средства встраиваемых систем. Классификация технических средств и элементной базы. Микропроцессор, микроконтроллер и программируемая логическая интегральная схема, их виды и характеристика. Модульный принцип организации встраиваемых вычислительных систем (ВВС). Типовая структура ВВС, ещё основные подсистемы, модули и интерфейсы.

Предпосылки к появлению операционных систем для ВВС и их эволюция. Обзор и характеристика встраиваемых операционных систем. Встраиваемые операционные системы на основе ядра Linux.

Тема 3.2. Проектирование, моделирование, реализация и тестирование встраиваемой системы с использованием ОС Raspberry PI и Node-RED

Подходы к проектированию встраиваемых систем интернета вещей. Имитационное моделирование и его использование для проектирования ВВС. Характеристика средств для имитационного моделирования встраиваемых систем. Операционная система Raspberry PI и инструментальные средства для моделирования. Командный интерфейс и набор системных утилит BusyBox. Возможности Node-RED, установка и настройка. Набор виртуальных сенсоров Sence Nat и их настройка в Node-RED.

Парадигма потокового программирования и применение для проектирования встраиваемых систем. Структура и архитектура встраиваемой системы и её моделирование с использованием ОС Raspberry PI и Node-RED. Имитационная модель окружающей среды. Обработка данных в режиме

реального времени с Node-RED. Хранения и визуализация данных с InfluxDB и Grafana. Тестирование и оценка моделируемой системы.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная (дневная) форма получения высшего образования с применением дистанционных образовательных технологий
(ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Программирование и проектирование мобильных приложений для ОС Android	16			16		2	
1.1	Введение в разработку мобильных приложений для ОС Android	2			2			Отчет
1.2	Проектирование Android-приложений: ресурсы и многоязычность	4			4			Отчет
1.3	Основы многозадачного мобильного интерфейса	4			2		2	Отчет, контрольная работа
1.4	Тестирование мобильных приложений	2			2			Отчет
1.5	Android I/O: от жестов до позиционирования	2			2			Отчет, контрольная работа
1.6	Работа с данными	2			4			Отчет, проект
2	Проектирование приложений для мобильных ОС Linux	12			10			

2.1	Введение в разработку приложений для мобильных ОС Linux	1						Собеседование
2.2	Технологии QT и основы QML для разработки мобильных приложений	1						Собеседование
2.3	Проектирование пользовательского интерфейса в ОС Аврора и работа с сервисами	4			4			Отчет
2.4	Программное средство Flutter для разработки мобильных приложений	4			4			Отчет, контрольная работа
2.5	Мультиплатформенные решения в мобильной разработке	2			2			Отчет, коллоквиум
3	Проектирование, моделирование и реализация приложений для встраиваемых систем	6			4		2	
3.1	Введение во встраиваемые вычислительные системы	2						Собеседование
3.2	Проектирование, моделирование, реализация и тестирование встраиваемой системы с использованием ОС Raspberry PI и Node-RED	4			4		2	Отчет, контрольная работа

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Прохоренок, Н. А. Qt 6. Разработка оконных приложений на C++ / Николай Прохоренок. – Санкт-Петербург : БХВ-Петербург, 2022. – 512 с. – URL: <https://ibooks.ru/bookshelf/386517>.
2. Скин, Дж. Kotlin. Программирование для профессионалов / Джош Скин, Дэвид Гринхол, Эндрю Бэйли ; [пер. с англ. Е. Матвеев]. – 2-е изд. – Санкт-Петербург; Москва; Минск: Питер, 2023. – 558 с. – [URL: https://ibooks.ru/bookshelf/387733](https://ibooks.ru/bookshelf/387733).
3. Смалюк, А. Ф. Разработка мобильных приложений Android : учебно-методическое пособие для обучающихся учреждений высшего образования / А. Ф. Смалюк, Ю. Н. Силкович ; М-во образования Республики Беларусь, БГУ, Институт бизнеса БГУ. – Минск : Ин-т бизнеса БГУ, 2022. – 75 с.

Дополнительная литература

1. Васильев, Н. П. Введение в гибридные технологии разработки мобильных приложений: учебное пособие / Н. П. Васильев, А. М. Заяц. — Санкт-Петербург; Москва; Краснодар: Лань, 2020. — 157 с. — URL: <https://e.lanbook.com/book/230387>
2. Deven, Joshi. Building Cross-platform Apps with Flutter and Dart / Joshi Deven. — BPB Online, 2023, — 378 p.
3. Japon, B. Learn IoT Programming Using Node-RED / B. Japon. — BPB Online, 2022, — 251 p.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Для диагностики компетенций могут использоваться следующие средства текущей аттестации: контрольная работа; коллоквиум; опрос; творческая работа (проект); отчёт о выполнении лабораторной работы.

Формой промежуточной аттестации по дисциплине «Программирование мобильных и встраиваемых систем» учебным планом предусмотрен зачёт и экзамен.

Для формирования итоговой отметки по учебной дисциплине используется модульно-рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущей и промежуточной аттестации студентов по учебной дисциплине.

Формирование итоговой отметки в ходе проведения контрольных мероприятий текущей аттестации (примерные весовые коэффициенты,

определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации):

коллоквиум – 10%;

отчёт по лабораторной работе и устный опрос – 65%;

проект – 15 %.

контрольная работа – 10%

Итоговая отметка по дисциплине рассчитывается на основе итоговой отметки текущей аттестации (модульно-рейтинговой системы оценки знаний) 40% и экзаменационной отметки 60%.

Примерный перечень заданий для управляемой самостоятельной работы

Тема 1.3. Основы многозадачного мобильного интерфейса (2 часа)

Примерный перечень вопросов:

1. Элементы управления для создания многооконных приложений.
3. Навигация в приложениях для ОС Android и стек переходов назад.
4. Фрагменты и менеджер фрагментов.

Примерный перечень заданий:

1. Разработать мобильное приложение с реализацией многооконного пользовательского интерфейса и применением фрагментов.
2. Сформулировать нефункциональные требования для приложения согласно индивидуальному заданию.
3. Спроектировать архитектуру и структуру приложения согласно индивидуальному заданию.

Форма контроля: отчёт по лабораторной работе.

Тема 3.2. Проектирование, моделирование, реализация и тестирование встраиваемой системы с использованием ОС Raspberry PI и Node-RED (2 часа)

Примерный перечень вопросов:

1. Возможности Node-RED для моделирования встраиваемых систем.
2. Установка и настройка Node-RED в ОС Raspberry PI.
3. Набор виртуальных сенсоров Sence Hat и их настройка.
4. Моделирование процесса получения данных от виртуальных сенсоров.

Примерный перечень заданий:

1. Установить и настроить Node-RED в ОС Raspberry PI.
2. Спроектировать процесс получения данных от виртуальных сенсоров согласно индивидуальному заданию.
3. Протестировать полученную модель, автоматизировать сбор данных и визуализацию.

Форма контроля: отчёт по лабораторной работе.

Рекомендуемая тематика коллоквиума

1. «Технологии проектирования приложений для мобильной ОС».

Примерная тематика контрольных работ

Контрольная работа 1. Разработка мобильного приложения с адаптивным интерфейсом.

Контрольная работа 2. Мультисенсорное взаимодействие в мобильных приложениях.

Контрольная работа 3. Реализация мобильного приложения для ОС Аврора.

Контрольная работа 4. Моделирование и симуляция встраиваемых систем.

Рекомендуемая тематика проекта

1. «Разработка мобильного приложения для ОС Android с применением инструментов командной разработки».

Примерный перечень лабораторных занятий

1. Лабораторная работа 1. Создание мобильного приложения и настройка среды разработки.

2. Лабораторная работа 2. Разработка адаптивного приложения с поддержкой локализации.

3. Лабораторная работа 3. Многопоточность в разработке мобильных интерфейсов.

4. Лабораторная работа 4. Проектирование тестов и тестирование мобильных приложений.

5. Лабораторная работа 5. Мультисенсорное взаимодействие в мобильных интерфейсах.

6. Лабораторная работа 6. Хранение и обработка данных в мобильных приложениях.

7. Лабораторная работа 7. Разработка мобильного приложения для ОС Аврора с использованием QT.

8. Лабораторная работа 8. Реализация мобильного приложения для ОС Аврора с использованием Flutter SDK.

9. Лабораторная работа 9. Проектирование мультиплатформенных приложений.

10. Лабораторная работа 10. Проектирование, моделирование и реализация приложения для встраиваемых систем.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса большинства занятий используется практико-ориентированный подход, который предполагает освоение содержания образования через решение практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности.

При организации образовательного процесса используется метод группового обучения, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

При проведении занятий в компьютерном классе основной формой работы является работа студентов над заданиями под руководством и контролем преподавателя. При этом должно обеспечиваться достаточное количество вариантов каждого задания. В основном, предполагается, что вариант задания является индивидуальным, т. е., рассчитанным на выполнение одним студентом, в некоторых случаях, например, проектных работах, задание может выполняться небольшой группой студентов (2-3 студента). Преподаватель должен оперативно консультировать выполнение заданий и принимать выполненное задание (оценивать результаты его выполнения) посредством визуальной проверки полученных результатов и собеседованием со студентом (группой студентов).

В силу различного уровня готовности студентов к восприятию новых понятий, на занятиях по дисциплине рекомендуется при необходимости проводить дополнительные консультации в малых группах студентов для объяснения и закрепления сложного материала.

Методические рекомендации по организации самостоятельной работы

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные технологии.

Разместить в сетевом доступе на образовательном портале БГУ комплекс учебных и учебно-методических материалов: учебно-программные комплексы, учебные издания для теоретического изучения дисциплины, презентации лекций, методические указания к лабораторным занятиям, электронные версии домашних заданий, материалы текущего контроля и текущей аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательного стандарта высшего образования и учебно-программной документации, в том числе вопросы для подготовки к экзамену, задания и вопросы для самоконтроля, список рекомендуемой литературы, информационных ресурсов и др.

Примерный перечень вопросов к зачёту

1. Архитектура ОС Android и Android API.
2. Архитектура Android-приложения и основные компоненты.
3. Структура проекта Android-приложения: характеристика.
4. Файл манифеста AndroidManifest.xml: элементы и структура.
5. Разметка (макеты) страниц на примере FrameLayout.
6. Разметка (макеты) страниц на примере LinearLayout.
7. Разметка (макеты) страниц на примере RelativeLayout.
8. Разметка (макеты) страниц на примере ConstraintLayout.
9. Разметка (макеты) страниц на примере TableLayout.
10. Разметка (макеты) страниц на примере GridLayout.
11. Разметка (макеты) страниц на примере ListView.
12. Разметка (макеты) страниц на примере GridView.
13. Ресурсы Android-приложения на примере строковых и числовых ресурсов.
14. Ресурсы Android-приложения на примере размеров, визуальных стилей и тем.
15. Ресурсы Android на примере ресурсов drawable и отрисовываемых объектов.
16. Ресурсы Android-приложения на примере необработанных объектов.
17. Намерения в Android-приложении: класс Intent и методы класса.
18. Операции в Android-приложении: класс Activity и методы класса.
19. Операция (activity), задача (task) и стек переходов назад.
20. Фрагменты и управление фрагментами в Android-приложении.
21. Сервисы в Android-приложении: класс Service и методы класса.
22. Контент-провайдеры в Android-приложении. Класс ContentProvider и методы класса.
23. Приёмники широковещательных сообщений (Broadcast Receivers) в Android-приложении: основные классы и методы.
24. Сенсорное управление в смартфонах под управлением ОС Android, основные классы и методы.
25. Хранение данных в ОС Android на примере внутреннего и внешнего хранилища: основные классы и методы.
26. Хранение данных в ОС Android на примере SharedPreferences
27. Хранение данных в ОС Android на примере DataStore.
28. Основные подходы разработки Android-приложений с использованием SQLite.
29. Классы и основные методы для работы с SQLite в Android.
30. Разработка мобильных приложений с использованием библиотеки Room.
31. Анимация в ОС Android и типы анимации.
32. Использование встроенной камеры и работа с мультимедиа в Android-приложениях.

33. Взаимодействие с системами позиционирования в Android-приложениях.
34. Многопоточность и параллелизм в ОС Android с применением механизмов языка Java.
35. Многопоточность и параллелизм в ОС Android с применением механизмов языка Kotlin.
36. Клиент-серверное взаимодействие и синхронизация данных в Android-приложении.
37. Декларативный способ создания пользовательского интерфейса с использованием Jetpack Compose.
38. Технологии для кроссплатформенной разработки мобильных приложений.
39. Архитектура ОС Аврора и инструментарий разработки мобильных приложений.
40. Язык разметки QML и основы вёрстки пользовательского интерфейса.
41. Базовые и основные визуальные типы QML.
42. Проектирование пользовательского интерфейса и библиотека QT Quick.
43. Объектная модель QT.
44. Базовый синтаксис языка программирования Dart.
45. Функции в Dart.
46. Классы и объекты в Dart.
47. Проектирование приложения для ОС Аврора на основе Flutter SDK.
48. Рекомендации по проектированию встраиваемых систем интернета вещей.
49. Имитационное моделирование и его использование для проектирования встраиваемых систем.
50. Структура и архитектура встраиваемой системы и ее моделирование с использованием ОС Raspberry PI и Node-RED.

Примерный перечень вопросов к экзамену

1. Операционная система Android, её возможности и история развития.
2. Структура и архитектура ОС Android и Android API.
3. Технологии мобильной разработки на языке программирования Java и Kotlin для ОС Android.
4. Инструментарий мобильной разработки для ОС Android.
5. Классификация мобильных приложений для ОС Android.
6. Архитектура Android-приложения и основные компоненты.
7. Структура проекта Android-приложения: характеристика.
8. Файл манифеста AndroidManifest.xml: элементы и структура.
9. Характеристика разметок (макетов) страниц, их типы и основные элементы UI в приложениях для ОС Android.
10. Разметка (макеты) страниц на примере FrameLayout и LinearLayout в приложениях для ОС Android.

11. Разметка (макеты) страниц на примере RelativeLayout и ConstraintLayout в приложениях для ОС Android.
12. Разметка (макеты) страниц на примере TableLayout и GridLayout в приложениях для ОС Android.
13. Разметка (макеты) страниц на примере ScrollView, ListView и GridView в приложениях для ОС Android.
14. Ресурсы Android-приложения: строковые, булевы, числовые, меню, ресурсы разметки.
15. Ресурсы Android-приложения: цветовые, размеров, визуальных стилей и тем, drawable-ресурсы.
16. Ресурсы Android-приложения: отрисовываемых объектов, мipmap, анимации, необработанных объектов.
17. Ресурсы Android-приложения: создание псевдонимов ресурсов, компиляция ресурсов, доступ из кода.
18. Иерархия классов в Android SDK.
19. Задачи и стек переходов назад в мобильных приложениях для ОС Android.
20. Фрагменты и управление фрагментами в приложениях для ОС Android.
21. Намерения в Android-приложении: класс Intent и методы класса.
22. Операции в Android-приложении: класс Activity и методы класса.
23. Сервисы в Android-приложении: класс Service и методы класса.
24. Контент-провайдеры. Класс ContentProvider и методы класса в приложениях для ОС Android.
25. Приёмники широковещательных сообщений (Broadcast Receivers): основные классы и методы в приложениях для ОС Android.
26. Рекомендации по разработке и проектированию интерфейсов мобильных приложений в приложениях для ОС Android.
27. Основы разработки многооконных приложений для ОС Android.
28. Сенсорное управление в смартфонах в приложениях для ОС Android.
29. Хранение данных в ОС Android на примере SharedPreferences.
30. Хранение данных в ОС Android на примере DataStore.
31. Хранение данных в ОС Android на примере внутреннего и внешнего хранилища: основные классы и методы.
32. Основные подходы разработки Android-приложений с использованием SQLite.
33. Основные классы и методы для работы с SQLite в Android.
34. Разработка мобильных приложений с использованием библиотеки Room.
35. 2D и 3D графика в мобильных приложениях для ОС Android.
36. Анимация в ОС Android и типы анимации.
37. Использование встроенной камеры и работа с мультимедиа в Android-приложениях.
38. Классификация сенсоров и датчиков для ОС Android и их характеристики.

39. Взаимодействие с системами позиционирования в приложениях для ОС Android.
40. Многопоточность и параллелизм в ОС Android с применением механизмов языка Java.
41. Многопоточность и параллелизм в ОС Android с применением механизмов языка Kotlin.
42. Клиент-серверное взаимодействие и синхронизация данных в приложениях для ОС Android.
43. Декларативный способ создания пользовательского интерфейса с использованием Jetpack Compose в приложениях для ОС Android.
44. Кроссплатформенная разработка с использованием инструментария Kotlin Multiplatform.
45. Язык программирования Dart, его концепция и базовый синтаксис.
46. Функции, классы и объекты в Dart.
47. Кроссплатформенная разработка с использованием инструментария Flutter.
48. Мобильные операционные системы на основе ядра Linux.
49. Операционная система Аврора.
50. Подходы и инструментарий разработки мобильных приложений для ОС Аврора.
51. Язык разметки QML и основы вёрстки пользовательского интерфейса в приложениях для ОС Аврора.
52. Базовые и основные визуальные типы QML.
53. Интерфейс ОС Аврора и элементы для проектирование пользовательского интерфейсы.
54. Объектная модель QT.
55. Характеристика системных API в приложениях для ОС Аврора.
56. Хранение данных и работа с базой данных SQLite в приложениях для ОС Аврора.
57. Проектирование приложения для ОС Аврора на основе Flutter SDK.
58. Системы реального времени, их виды и особенности.
59. Встраиваемые системы и их категории.
60. Технические средства встраиваемых систем.
61. Встраиваемые операционные системы на основе ядра Linux.
62. Имитационное моделирование и его использование для проектирования встраиваемых система.
63. Операционная система Raspberry PI и инструментальные средства для моделирования.
64. Парадигма потокового программирования и моделирование окружающей среды.
65. Инструмент потокового программирования Node-RED, характеристика, установка и основные компоненты
66. Структура и архитектура встраиваемой системы и ее моделирование с использованием ОС Raspberry PI и Node-RED.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Учебная дисциплина не требует согласования			

Заведующий кафедрой технологий
программирования
доктор технических наук, профессор



А.Н.Курбацкий

15.05.2025

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УО

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

УТВЕРЖДАЮ
Декан факультета
