

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

 О.Г. Прохоренко
«05» июля 2023 г.

Регистрационный № УД-12567/уч.



Программирование мобильных и встраиваемых систем

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

**1-31 03 07 Прикладная информатика (по направлениям)
направления специальности**

1-31 03 07-01 Прикладная информатика (программное обеспечение
компьютерных систем)

2023 г.

Учебная программа составлена на основе ОСВО 1-31 03 07-2021, типового учебного плана № G 31- 1- 030/пр-тип., утверждённого 01.07.2021, учебных планов БГУ №G31-1-034/уч. от 23.07.2021, №G31-1-023/уч. ин. от 09.08.2021, №G 31-1-216/уч. от 22.03.2022 г., №G31-1-224/уч. ин. от 27.05.2022.

СОСТАВИТЕЛИ:

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

РЕЦЕНЗЕНТЫ:

И. А. Король, заместитель директора республиканского унитарного предприятия «Национальный центр электронных услуг», 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

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования
(протокол № 16 от 18 мая 2023 г.);

Научно-методическим советом БГУ
(протокол № 9 от 29 июня 2023 г.)

Заведующий кафедрой



А.Н. Курбацкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины — подготовка специалиста по проектированию и разработке мобильных приложений на языках Java, Kotlin и C++ 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;
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.

Дисциплина «Программирование мобильных и встраиваемых систем» 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 приложений специалисту по разработке программного обеспечения невозможно эффективно решать профессиональные задачи. Наиболее востребованными навыками являются проектирование архитектуры программных систем, управление ИТ-проектами, разработка мобильных и встраиваемых приложений, применение современных библиотек и SDK для

мобильных и встраиваемых операционных систем, сборка и тестирование приложений, моделирование и симуляция встраиваемых систем.

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», «Системное программирование».

Изучение дисциплины «Программирование мобильных и встраиваемых систем» будет востребовано при изучении дисциплины «Проектирование человеко-машинных интерфейсов» и других дисциплин модуля «Прикладное программирование», так же дисциплин специализации «Актуальные решения в сфере мобильной разработки», «Информационная безопасность мобильных приложений», «Проектирование приложений под Linux» и при выполнении курсовых и дипломных работ.

Требования к компетенциям

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

универсальные компетенции:

УК-1. Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации.

УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

УК-4. Работать в команде, толерантно воспринимать социальные, этнические, конфессиональные, культурные и иные различия.

УК-5. Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

УК-6. Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

базовые профессиональные компетенции:

БПК-2. Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного подходов и иных парадигм программирования.

БПК-4. Применять знания в области принципов функционирования, архитектур и программных реализация операционных систем, структурной организации компьютеров и компьютерных систем, методах обработки данных для выбора вычислительных средств решения практических задач.

специализированные компетенции:

СК-5. Использовать программные средства и технологии для создания прикладного программного обеспечения.

В результате освоения учебной дисциплины студент должен:

знать:

- методы проектирования мобильных и встраиваемых программных систем;
- методы проектирования интерфейса пользователя мобильных и встраиваемых приложений;
- методы разработки программных продуктов с использованием различных инструментальных средств;
- методы проектирования баз данных;
- основные понятия и методы тестирования программного обеспечения;

уметь:

- анализировать требования, предъявляемые к мобильным приложениям и встраиваемым программным системам, и составлять техническое задание;
- моделировать процессы во встраиваемых системах программное обеспечение;
- создавать различные виды тестов для оценки качества программ и тестировать их;
- проектировать интерфейс пользователя приложений;
- разрабатывать мобильные приложения и встраиваемые программные комплексы с использованием различных инструментальных средств;

владеть:

- навыками создания мобильных приложений с применением Android SDK и Augora SDK;
- навыками программирования с использованием языка программирования Java;
- навыками программирования с использованием языка программирования Kotlin;
- навыками разработки приложений для встраиваемых вычислительных систем;
- навыками анализа и проектирования архитектуры приложений.

Структура учебной дисциплины

Дисциплина изучается в 6 семестре. 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, из них: лекции – 34 часов, лабораторные занятия – 30 часов, управляемая самостоятельная работа – 4 часа.

Трудоёмкость учебной дисциплины составляет 3 зачётные единицы.

Форма текущей 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. Архитектура приложений для Android. Технологии мобильной разработки на языке программирования Java. Android SDK. Android NDK. Java-интерфейсы. Язык программирования Kotlin для мобильной разработки. Форматы пакетов приложений. Инструментарий для мобильной разработки. Запуск и отладка приложений. Эмуляторы. Запуск Android-приложений в окружении других операционных систем. Классификация мобильных приложений по видам.

Тема 1.2. Структура приложения, его компоненты, макеты, ресурсы и локализация

Основные компоненты мобильных приложений в ОС Android. Структура проекта и его различные представления. Манифест приложения. Элементы манифеста и их атрибуты. Разметка экрана мобильного приложения и способы построения пользовательского интерфейса (User Interface – UI). Императивный и декларативный способ создания UI. Основные виды разметок и их характеристики. Jetpack Compose как пример декларативного определения элементов UI.

Ресурсы и создание ресурсов. Типы ресурсов (простые значения, визуальные стили и темы, изображения, разметка, анимация, меню и др.). Определение и использование ресурсы. Альтернативные ресурсы. Локализация приложения. Компиляция ресурсов.

Тема 1.3. Основные компоненты, проектирование пользовательского интерфейса и тестирование

Основные компоненты для создания пользовательского интерфейса приложения. 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.

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 диалогов: TimePickerDialog, DatePickerDialog, ProgressDialog, AlertDialog. DialogFragment – класс диалога на основе фрагментов. Список ListView. Одиночный и множественный выбор в ListView. Обработка событий и использование адаптеров. Особенности разработки приложения, содержащего несколько активностей. Перелистывание (Swipe). Меню в Android. Дочерние и контекстные меню. Описание меню с помощью XML.

Сборка и тестирование приложения. Проектирование модульных тестов и тестов пользовательского интерфейса.

Тема 1.4. Обработка жестов, графики, медиа данных и позиционирование

Отличительные особенности смартфонов. Сенсорное (touch) управление. Короткие нажатия и обработка с помощью OnClickListener(). Одиночные касания (нажатие, движение, отпускание) в обработчике onTouchListener(). MultiTouch – обработка множественных касаний.

Инструменты 2D и 3D графики в Android. Работа с изображениями и мультимедиа. Класс MediaPlayer – аудио/видео плеер, основные возможности. Запись звука с помощью MediaRecorder и AudioRecorder. 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 Google.

Тема 1.5. 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.

Тема 1.6. Многопоточность, синхронизация и клиент-серверное взаимодействие

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.

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

Тема 1.7. Декларативные программные средства и кроссплатформенные решения для создания мобильных приложений

Декларативный способ создания пользовательского интерфейса. Библиотека Jetpack Compose. Архитектура мобильного приложения на основе Jetpack Compose. Основные слои и компоненты. Технология Kotlin Multiplatform для платформозависимой разработки.

Решения для кроссплатформенной разработки на примере Kotlin Multiplatform (Compose Multiplatform) и Flutter. Архитектура Compose Multiplatform. Архитектура и основные компоненты Flutter.

Раздел 2. Подходы к проектированию и реализации приложений для мобильной ОС Аврора

Тема 2.1. Введение в разработку приложений для ОС Аврора

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 в ОС Аврора

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

Программное средство Flutter, его возможности и характеристика. 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 для ОС Аврора.

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

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		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	18			18		2	
1.1	Введение в разработку мобильных приложений для ОС Android	2			2			Собеседование Отчет по лабораторной работе
1.2	Структура приложения, его компоненты, макеты, ресурсы и локализация	4			2			Собеседование Отчет по лабораторной работе
1.3	Основные компоненты, проектирование пользовательского интерфейса и тестирование	4			2		2	Собеседование Отчет по лабораторной работе
1.4	Обработка жестов, графики, медиа данных и позиционирование	2			4			Собеседование Отчет по лабораторной работе
1.5	Работа с данными	2			4			Собеседование Отчет по лабораторной работе
1.6	Многопоточность и сетевое	2			4			Собеседование

	взаимодействие							Проект Контрольная работа
1.7	Декларативные программные средства и кроссплатформенные решения для создания мобильных приложений	2						Собеседование
2	Подходы к проектированию и реализации приложений для мобильной ОС Аврора	10			8			
2.1	Введение в разработка приложений для ОС Аврора	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			Собеседование Отчет по лабораторной работе Коллоквиум
2.4	Программное средство Flutter для разработки мобильных приложений в ОС Аврора	4			4			Собеседование Отчет
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	Собеседование Отчет по лабораторной работе Контрольная работа

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

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. Гриффитс, Д. Head First. Программирование для Android / Дэвид Гриффитс, Дон Гриффитс; [пер. с англ. Е. Матвеев]. — 2-е изд. — Санкт-Петербург [и др.]: Питер, 2018. — 909 с. — URL: <https://ibooks.ru/bookshelf/358143>.
3. Прохоренок, Н. А. Qt 6. Разработка оконных приложений на C++ / Николай Прохоренок. — Санкт-Петербург : БХВ-Петербург, 2022. — 512 с.
4. 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>.
5. Смалюк, А. Ф. Разработка мобильных приложений Android : учебно-методическое пособие для обучающихся учреждений высшего образования, осваивающих образовательную программу высшего образования I степени 1-26 03 01 "Управление информационными ресурсами" / А. Ф. Смалюк, Ю. Н. Силкович ; М-во образования Республики Беларусь, БГУ, Институт бизнеса БГУ. — Минск : Ин-т бизнеса БГУ, 2022. — 75 с.

Перечень дополнительной литературы

1. Дейтел, П. Android для разработчиков / П. Дейтел, Х. Дейтел, А. Уолд. 3-е изд. — СПб.: Питер, 2016. — 512 с.: ил.
2. Льюис, Ш. Нативная разработка мобильных приложений: перекрестный справочник для iOS и Android / Шон Льюис, Майк Данн; пер. с англ. яз. А. Н. Киселева. — Москва: ДМК Пресс, 2020. — 375 с. — URL: <https://ibooks.ru/bookshelf/372097>
3. Шрёдер, К. Linux. Книга рецептов. Все необходимое для администраторов и пользователей / Карла Шрёдер; [пер. с англ. А. Киселев]. — 2-е изд. — Санкт-Петербург; Москва; Минск: Питер, 2022. — 588 с. — URL: <https://ibooks.ru/bookshelf/385988>.
4. Deven, Joshi. Building Cross-platform Apps with Flutter and Dart / Joshi Deven. — BPB Online, 2023, — 378 p.
5. 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 в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование.
2. Письменная форма: коллоквиум, контрольная работа.
3. Устно-письменная форма: отчёт по лабораторным работам, проект с устной защитой и оценением на основе проектного метода.

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

Итоговая отметка формируется на основе

1. Правил проведения аттестации студентов, курсантов, слушателей при освоении содержания образовательных программ высшего образования (Постановление Министерства образования Республики Беларусь № 53 от 29 мая 2012 г.);

2. Положения о рейтинговой системе оценки знаний обучающихся по учебной дисциплине в БГУ (Приказ ректора БГУ от 31.03.2020 № 189-ОД);

3. Критериев оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале (Письмо Министерства образования Республики Беларусь от 28.05.2013 г. № 09-10/53-ПО).

Текущий контроль самостоятельной работы студентов по усвоению знаний по учебной дисциплине «Программирование мобильных и встраиваемых систем» рекомендуется осуществлять преподавателем на лабораторных занятиях в виде проверки выполнения индивидуальных заданий, а также в процессе собеседования со студентом в ходе такой проверки. Такое совмещение проверки, собеседования и индивидуальной консультации позволяет наиболее полно оценить уровень знаний и практических навыков студента по рассматриваемой теме. Однако оно возможно при проведении занятий в небольших группах, предъявляет требования к организации учебного процесса и высокому уровню мотивации и отдачи преподавателя.

При формировании итоговой отметки используется рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний в итоговую отметку:

- коллоквиум – 10%;
- отчёт по лабораторной работе – 65%;
- проект – 15 %.
- контрольная работа – 10%

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей успеваемости (рейтинговой системы оценки знаний) и экзаменационной отметки с учётом их весовых коэффициентов. Вес отметки по текущей успеваемости составляет 40 %, экзаменационной отметки – 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема № 1.3. Основные компоненты, проектирование пользовательского интерфейса и тестирование (2 часа)

Примерный перечень вопросов:

1. Основные компоненты для проектирования мобильных приложений для ОС Android.
2. 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: собеседование.

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: собеседование.

Рекомендуемая тематика коллоквиума

1. «Проектирование и реализация приложений для мобильной ОС».

Примерная тематика контрольных работ

Контрольная работа 1. Разработка мобильного приложения с адаптивным интерфейсом для ОС Android.

Контрольная работа 2. Реализация мобильного приложения для ОС Aurora.

Рекомендуемая тематика проекта

1. «Проектирование и разработка мобильного приложения для ОС Android с применением инструментов командной разработки».

Примерная тематика лабораторных занятий

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

Лабораторная работа 2. Разработка адаптивного приложения с поддержкой локализации.

Лабораторная работа 3. Разработка интерфейсов мобильных приложений и основы тестирования.

Лабораторная работа 4. Разработка мобильных приложений с функциями распознавания жестов.

Лабораторная работа 5. Разработка мобильных приложения с определением местоположения.

Лабораторная работа 6. Разработка мобильных приложений с хранением информации в базе данных и в файле настроек.

Лабораторная работа 7. Многопоточность и клиент-серверное взаимодействие.

Лабораторная работа 8. Разработка мобильного приложения для ОС Автора с использованием QT.

Лабораторная работа 9. Реализация мобильного приложения для ОС Автора с использованием Flutter SDK.

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 обучающихся

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 зачету

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. Фрагменты и управление фрагментами.
18. Намерения в Android-приложении: класс Intent и методы класса.
19. Операции в Android-приложении: класс Activity и методы класса.
20. Сервисы в Android-приложении: класс Service и методы класса.
21. Контент-провайдеры. Класс ContentProvider и методы класса.
22. Приёмники широковещательных сообщений (Broadcast Receivers): основные классы и методы.
23. Сенсорное управление в смартфонах, основные классы и методы.
24. Хранение данных в ОС Android на примере SharedPreferences и DataStore.
25. Хранение данных в ОС Android на примере внутреннего и внешнего хранилища: основные классы и методы.
26. Основы работы с базами данных SQLite. Запросы в sqlite.
27. Классы и основные методы для работы с SQLite.
28. Анимация в ОС Android и типы анимации.
29. Использование встроенной камеры и работа с мультимедиа в Android-приложениях.
30. Взаимодействие с системами позиционирования: права доступа, основные классы и методы.
31. Многопоточность и параллелизм в ОС Android с применением механизмов языка Java.
32. Многопоточность и параллелизм в ОС Android с применением механизмов языка Kotlin.
33. Клиент-серверное взаимодействие и синхронизация данных.
34. Декларативный способ создания пользовательского интерфейса с использованием Jetpack Compose.
35. Язык разметки QML и основы верстки пользовательского интерфейса.
36. Базовые и основные визуальные типы QML.
37. Элементы для проектирования пользовательского интерфейса QT Quick.
38. Объектная модель QT.
39. Проектирование приложения для ОС Аврора на основе Flutter SDK.

40. Рекомендации по проектированию встраиваемых систем интернета вещей.
41. Имитационное моделирование и его использование для проектирования ВВС.
42. 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.
4. Инструментарий мобильной разработки.
5. Классификация мобильных приложений.
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.
10. Разметка (макеты) страниц на примере FrameLayout и LinearLayout.
11. Разметка (макеты) страниц на примере RelativeLayout и ConstraintLayout.
12. Разметка (макеты) страниц на примере TableLayout и GridLayout.
13. Разметка (макеты) страниц на примере ScrollView, ListView и GridView.
14. Ресурсы Android-приложения: строковые, булевые, числовые, меню, ресурсы разметки.
15. Ресурсы Android-приложения: цветовые, размеров, визуальных стилей и тем, drawable.
16. Ресурсы Android-приложения: отрисовываемых объектов, mipmap, 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.
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.
25. Приёмники широковещательных сообщений (Broadcast Receivers): основные классы и методы.

26. Рекомендации по разработке и проектированию интерфейсов мобильных приложений.
27. Основы разработки многооконных приложений для ОС Android.
28. Сенсорное управление в смартфонах.
29. Хранение данных в ОС Android на примере SharedPreferences и DataStore.
30. Хранение данных в ОС Android на примере внутреннего и внешнего хранилища: основные классы и методы.
31. Основы работы с базами данных SQLite. Запросы в sqlite.
32. Классы и основные методы для работы с SQLite.
33. 2D и 3D графика в мобильных приложениях для ОС Android.
34. Анимация в ОС Android и типы анимации.
35. Использование встроенной камеры и работа с мультимедиа в Android-приложениях.
36. Классификация сенсоров и датчиков для ОС Android и их характеристики.
37. Взаимодействие с системами позиционирования: права доступа, основные классы и методы.
38. Многопоточность и параллелизм в ОС Android с применением механизмов языка Java.
39. Многопоточность и параллелизм в ОС Android с применением механизмов языка Kotlin.
40. Клиент-серверное взаимодействие и синхронизация данных.
41. Декларативный способ создания пользовательского интерфейса с использованием Jetpack Compose.
42. Кроссплатформенная разработка с использованием инструментария Kotlin Multiplatform.
43. Кроссплатформенная разработка с использованием инструментария Flutter.
44. Мобильные операционные системы на основе ядра Linux.
45. Операционная система Аврора.
46. Подходы и инструментарий разработки мобильных приложений для ОС Аврора.
47. Язык разметки QML и основы верстки пользовательского интерфейса.
48. Базовые и основные визуальные типы QML.
49. Интерфейс ОС Аврора и элементы для проектирования пользовательского интерфейса.
50. Объектная модель QT.
51. Характеристика системных API.
52. Хранение данных и работа с базой данных SQLite.
53. Проектирование приложения для ОС Аврора на основе Flutter SDK.
54. Системы реального времени, их виды и особенности
55. Встраиваемые системы и их категории.
56. Технические средства встраиваемых систем.
57. Встраиваемые операционные системы на основе ядра Linux.

58. Рекомендации по проектированию встраиваемых систем интернета вещей.
59. Имитационное моделирование и его использование для проектирования ВВС.
60. Структура и архитектура встраиваемой системы и ее моделирование с использованием ОС Raspberry PI и Node-RED.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

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)
1. Проектирование человеко-машинных интерфейсов	Кафедра технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 16 от 18.05.2023)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

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
