

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

 О. Г. Прохоренко

«08» июля 2022 г.

Регистрационный № УД – 11361/уч.

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ ДЛЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

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

1-31 03 07 Прикладная информатика (по направлениям)

Направление специальности:

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

2022 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

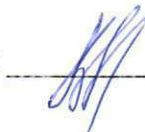
Т. В. Соболева, доцент кафедры многопроцессорных систем и сетей факультета прикладной математики и информатики Белорусского государственного университета, кандидат физико-математических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования
Белорусского государственного университета
(протокол № 13 от 19 мая 2022 г.).

Научно-методическим Советом Белорусского государственного университета
(протокол № 6 от 29.06.2022 г.).

Заведующий кафедрой
технологий программирования



А. Н. Курбацкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины — подготовка специалиста по проектированию и разработке приложений на языках C, Objective-C, Swift, включая мобильные приложения для операционной системы iOS. Формирование и развитие системы знаний по теоретическим и практическим аспектам проектирования и разработки приложений, получаемых будущими специалистами в процессе учёбы и необходимых им в дальнейшем для успешной работы при разработке мобильных приложений.

Задачи учебной дисциплины:

1. Изучить инструментарий для проектирования и разработки консольных и мобильных приложений;
2. Изучить синтаксис языков программирования Objective-C и Swift;
3. Научить эффективно применять полученные навыки проектирования и разработки приложений в профессиональной деятельности;
4. Сформировать навыки проектирования архитектуры и разработки мобильных приложений.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Дисциплина «Технологии программирования для мобильных приложений» относится к модулю «Программирование» государственного компонента. Одним из факторов успешного развития образования, здравоохранения, промышленности, сельского хозяйства и других отраслей является применение автоматизированных программных систем для управления производственным циклом. Следовательно, повышаются требования к уровню компетентности выпускников вузов в области разработки программного обеспечения. Современное программное обеспечение представляют собой комплексную систему, состоящую из взаимосвязанных программ с различным назначением. Одной из категорий программ выступают мобильные приложения. Их применение обосновано доступностью и возможностями современных мобильных устройств. Без теоретических знаний и практических навыков разработки мобильных приложений специалисту по разработке программного обеспечения невозможно эффективно решать профессиональные задачи. Наиболее востребованными навыками являются управление исходным кодом проектов с помощью систем контроля версий, проектирование, разработка и сборка многофайловых проектов, разработка приложений с хранением данных в базе данных, проектирование требований и спецификации с помощью унифицированного языка моделирования (Unified Modelling Language — UML), управление ИТ-проектами, разработка мобильных приложений с использованием библиотек и SDK в зависимости от мобильной операционной системы, сборка и тестирование приложений.

Связи с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Содержание учебной программы соответствует уровню подготовленности студентов к изучению данной дисциплины и основывается на базовых учебных дисциплинах I степени высшего образования, включая «Основы и методология программирования», «Разработка кросс-платформенных приложений», «Промышленное программирование».

Изучение дисциплины «Технологии программирования для мобильных приложений» будет востребовано при изучении дисциплины «Системное программирование», дисциплины «Программирование мобильных и встраиваемых систем» и других дисциплин модуля «Прикладное программирование», так же дисциплин специализации и при выполнении курсовых и дипломных работ.

Требование к компетенциям

Освоение учебной дисциплины «Технологии программирования для мобильных приложений» 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-5. Быть способным к саморазвитию и совершенствованию 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.

специализированные компетенции:

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

В результате освоения учебной дисциплины студент должен:

знать:

- методы проектирования программных систем;
- методы проектирования интерфейса пользователя;
- 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 программным системам, составлять техническое задание, моделировать программное обеспечение, а также проводить его тестирование;
- проводить оценку качества программного обеспечения;
- проектировать интерфейс пользователя приложений;
- разрабатывать программные комплексы с использованием различных инструментальных средств;
- управлять версиями приложений;

владеть:

- навыками создания приложений с использованием баз данных;
- навыками программирования с использованием языка программирования Objective-C;
- навыками программирования с использованием языка программирования Swift;
- средствами для решения задач проектирования и разработки мобильных приложений;
- навыками анализа и проектирования архитектуры приложений.

Структура учебной дисциплины

Дисциплина изучается в 4 семестре. Всего на изучение учебной дисциплины «Технологии программирования для мобильных приложений» отведено:

– для очной формы получения высшего образования – 216 часов, в том числе 72 аудиторных часа, из них: лекции – 36 часов, лабораторные занятия – 30 часов, управляемая самостоятельная работа – 6 часов.

Трудоёмкость учебной дисциплины составляет 6 зачётных единицы.

Форма текущей аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Технологии управления, проектирования и разработки программного обеспечения

Тема 1.1. Пользовательское окружение macOS, командный интерфейс и файловая система macOS для управления ИТ-проектами

Эволюция операционных систем. История развития проекта Unix. Современные операционные системы. Операционные системы семейства BSD. История развития и концепция macOS. Технологии виртуализации и запуск macOS в виртуальной машине.

Архитектура и экосистема macOS. Пакетный менеджер и способы установки приложений.

Командный интерфейс, понятие консоли и терминала. Синтаксис командной строки. Командный интерпретатор на примере bash и zsh. Базовые команды, права доступа. Справочные системы.

Иерархия файловой системы. Файлы и дерево каталогов. Типы файлов. Домены. Файловые системы HFS+ (Mac OS Extended), APFS и уровни доступа.

Тема 1.2. Управление кодом и документирование проекта в продуктовой разработке

Системы контроля версий (СКВ) для управления исходным кодом приложений. Типы СКВ. Распределенная система контроля версий и управления кодом git. Установка git. Создание и инициализация репозитория. Клонирование репозитория. Состояния файлов под управлением git. Запись и фиксация изменений в локальный репозиторий. Внешние репозитории, подключение и настройка. Публикация изменений во внешний репозиторий и получение изменений из репозитория. Управление локальными и внешними ветками. Консольный клиент git. Графические клиентские приложения управления версиями.

Язык разметки Markdown. Документирование проекта с помощью файла README. Документирование проекта в wiki. Github Pages и другие сервисы документирования проектов.

Тема 1.3. Многофайловое программирование, сборка и непрерывная интеграция приложений

Разработка приложений на языке C и сборка консольных приложений. Структура проекта по модели «Клиент-Интерфейс-Сервер» (КИС). Файл Makefile, основные понятия и синтаксис. Утилита make и другие утилиты сборки. Современные решения для автоматической сборки и распространения. Онлайн-сервисы для разработки, управления и сборки проектов на примере Repl.it и других.

Технологии непрерывной интеграции: характеристики и возможности. Основные понятия, файлы конфигурации и характеристики Github Actions.

Другие решения для непрерывной интеграции и их интеграция с системой контроля версий.

Тема 1.4. Разработка требований и спецификация ПО с применением UML

Унифицированный язык моделирования (Unified Modelling Language – UML) для проектирования ПО. Классификация диаграмм UML. Диаграммы вариантов использования, деятельности, состояний, последовательности, объектов, классов, пакетов, компонентов и размещения и др., их элементы и связи между элементами. Правила обозначения для различных видов диаграмм. Виды связей в зависимости от типа диаграммы. Правила составления спецификаций и сценариев.

Функциональные и нефункциональные требования. Спецификация ПО и его структура.

Тема 1.5. Разработка приложений и хранение данных в базе данных SQLite

Основы SQLite. Типы данных SQLite. Типы соответствия данных SQLite и правила определения соответствия типов. Консоль SQLite и основные команды управления базой данных (БД).

Стандарт SQL и синтаксис команд SQL в SQLite. Выборки и параметрические запросы.

Разработка приложений на языке C и хранение данных в БД SQLite. Объектная модель SQLite C API. Объект-соединение и методы взаимодействия с базой данных SQLite. Создание и подключение к базе данных. Создание таблиц и удаление таблиц. Вставка, выборка, обновление и удаление данных. Транзакции в SQLite.

Тема 1.6. Управление ИТ-проектами

Введение в управление ИТ-проектами. Базовые термины проектного управления. Гибкие методологии разработки программного обеспечения. Принципы и ценности Agile. Обзор популярных гибких методологий на примере Scrum, Lean, Kanban и др. Применение Kanban для управления проектами. Онлайн-сервисы для управления проектами.

Раздел 2

Технологии разработки мобильных приложений для ОС iOS

Тема 2.1. Введение в разработку мобильных приложений для операционной системы iOS

Введение в операционную систему iOS. Архитектура iOS. Типы API в macOS и iOS. Springboard как сервис, отвечающий за пользовательский интерфейс в iOS, и Backboard как сервис, отвечающий за обработку сообщений от оборудования. Этапы загрузки iOS. Слои абстракции iOS:

CoreOS, Core Services, Media и Cocoa Touch. Стандартные системные библиотеки и SDK в iOS.

Инструменты для разработки iOS-приложений. Возможности IDE Xcode. Конфигурирование и настройка проекта, файл списка свойств формата .plist. Возможности и настройка симулятора. Редактирование, документирование и комментирование кода. Управление версиями, подключение локальных и внешних репозиториях. Отладка в Xcode. Правила публикации приложений в AppStore.

Тема 2.2. Базовые средства языка программирования Objective-C

Характеристики Objective-C. Токены, идентификаторы и зарезервированные слова в Objective-C.

Базовые средства языка C в Objective-C. Объекты и синтаксис отправки сообщений. Типы данных, квалификаторы, константы, выражения и переменные в Objective-C. Числовые типы данных. Перечислимый тип данных. Операторы Objective-C.

Ввод и вывод в Objective-C. Комментирование кода в Objective-C.

Строки и управление строковым типом данных. Структуры и указатели в Objective-C. Коллекции в Objective-C и их характеристики. Массивы в Objective-C и их методы. Словари в Objective-C и их методы. Наборы в Objective-C и их методы.

Условные операторы и циклы в Objective-C.

Тема 2.3. Расширенные средства языка программирования Objective-C

Понятие объекта и класса. Объявление и реализация классов. Объявление, выделение памяти, инициализация экземпляра класса и механизмы освобождения памяти.

Синтезируемые методы доступа классов. Области видимости класса и экземпляра класса. Концепция наследования классов в Objective-C.

Структура проектов на языке Objective-C. Определение интерфейсов классов, их реализации и блока программы согласно модели КИС.

Функции в Objective-C. Объявление функций. Области действия функций и спецификатор static. Переменные в функциях.

Протоколы в Objective-C и их концепция. Методы протокола. Наследование протоколов. Поддержка протокола классом.

Тема 2.4. Базовые возможности языка Swift

Язык программирования Swift, особенности и базовые понятия. Переменные и константы в Swift. Правила объявления переменных и констант. Области видимости переменных.

Ввод и вывод информации в языке Swift. Комментирование кода в Swift.

Числовые типы данных в Swift. Копирование типа. Операторы (арифметические, соответствия, сравнения и диапазона) в языке Swift. Приведение числовых типов. Псевдонимы типов.

Текстовые типы данных в Swift. Строковый литерал. Пустой строковый литерал. Многострочный строковый литерал. Приведение к строковому типу данных. Объединение и сравнение строк. Кодировка Unicode в строковых типах данных. Методы строковых типов данных.

Логический тип данных и логические операторы.

Управляющие конструкции. Утверждения. Оператор условия if и оператор ветвления switch. Циклы и операторы.

Тема 2.5. Контейнерные и опциональные типы данных в языке Swift

Контейнерные типы данных и их характеристики. Понятие последовательности и коллекции.

Кортежи: литерал кортежа, объявление кортежа, тип данных кортежа, явное и неявное указание типа, доступ к элементам кортежа и операции над кортежами.

Диапазоны: полуоткрытый и закрытый диапазон, бинарная, префиксная и постфиксная формы диапазона, свойства и методы диапазонов.

Массивы в Swift, их свойства и методы. Словари в Swift, их свойства и методы. Наборы в Swift, их свойства и методы.

Опциональные типы данных. Опциональный тип данных в кортежах. Извлечение опционального значения. Проверка значения опционального типа данных. Опциональное связывание.

Тема 2.6. Объектно-ориентированное программирование и расширенные возможности языка Swift

Функции. Входные параметры и возвращаемое значение. Функциональный тип. Функция в качестве входного и возвращаемого значений.

Объектные типы данных: структуры, перечисления и классы. Перечисления: синтаксис, параметры, свойства и методы. Структуры, их синтаксис, свойства и методы.

Классы, их синтаксис, свойства и методы. Требования к классам и типы свойств. Наследование классов.

Протоколы, их свойства и методы. Наследование протоколов.

Структура проектов на языке Swift. Определение функций и их вызов, объявление интерфейсов классов и их реализация в многофайловых проектах на языке Swift.

Тема 2.7. Проектирование интерфейса, проектирование архитектуры и тестирование мобильных приложений

Структура проекта мобильного приложения на языке Objective-C и Swift. Жизненный цикл мобильного приложения.

Проектирование интерфейса мобильного приложения в Interface Builder и библиотека UIKit. Основные элементы управления и их виды. Представления View и их типы. Привязка элементов интерфейса и обработка событий. ViewController и его жизненный цикл. Адаптация элементов интерфейса под разные устройства с помощью AutoLayout. Использование ресурсов приложения (изображения, прочие данные). Понятие делегирования. Обработка ввода текста. Проектирование макетов и навигации на примере UITableView, UICollectionView и других представлений. Интерфейсные элементы навигации (TabBarController и UINavigationController).

Проектирование интерфейса мобильного приложения и библиотека SwiftUI. Ключевые понятия и элементы интерфейса SwiftUI. Управление свойствами представлений и переходами между ними.

Подходы к проектированию архитектуры мобильного приложения для iOS. Концепция архитектурного шаблона MVC для проектов мобильных приложений на языке Objective-C и на языке Swift. Архитектурный шаблон MVVM и структура проекта мобильного приложения.

Модульное тестирование. Преимущества и недостатки модульного тестирования. Разработка модульных тестов. Тестирование пользовательского интерфейса (User Interface — UI) и разработка UI-тестов в Xcode.

Тема 2.8. Графика, анимация, жесты и их обработка в мобильных приложениях

Библиотека Core Graphics и её основные возможности. Геометрические типы данных. Графические примитивы и двухмерная графика. Цвета и шрифты. Рисование. Использование камеры и фотоархива. Селектор изображений и галерея изображений.

Анимация представлений и анимация слоёв. Возможности библиотеки Core Animation. Анимация и анимируемые свойства UIView. Анимация с помощью CALayer и его наследников.

Жесты и виды жестов. Обработка жестов. Реагирующий элемент и цепочка реагирующих элементов. Мультисенсорная архитектура. Методы уведомления о касаниях. Обработка одного и нескольких жестов. Распознавание жестов и UIGestureRecognizer.

Тема 2.9. Интернационализация, локализация и определение местоположения в мобильных приложениях

Интернационализация и локализация. Локализация интерфейса и файлы локализации. Локализация макетов storyboard. Интернационализация изображений и форматов данных. Словарь конфигурации спецификатора спецификации формата (поддержка множественного числа). Добавление нового языка. Локализация иконки приложения.

Библиотека Core Location. Определение местоположения, основные подходы и методы. Разрешение на использование служб определения

местоположения. Диспетчер местоположения CLLocationManager. Запуск и использование диспетчера местоположения. Требуемая точность определения местоположения и фильтры расстояний. Обновление координат и определение расстояния между объектами CLLocation.

Библиотека MapKit. Визуализация карты. Понятие маркера и MKAnnotation. MKDirections (направления) и MKOverlay («наложения»). Методы делегата MKMapViewDelegate. Автоматическая группировка маркеров при масштабировании.

Тема 2.10. Хранение данных в приложениях для iOS

Механизмы хранения данных в файловой системе iOS: списки свойств, архивы объектов, SQLite3, Core Data. Каталоги приложения (Documents, Library и tmp), определение местоположения и права доступа. Стратегии хранения файлов их преимущества и недостатки.

Списки свойств. Запись и считывание. Сериализация списка свойств. Управление настройками приложения и класс UserDefaults (NSUserDefaults).

SQLite3 и способы подключения библиотеки в проект. Открытие и закрытие базы данных. Создание и наполнение таблиц БД. Извлечение данных из БД. Связанные переменные.

Библиотека Core Data. Сущности и управляемые объекты. Свойства сущностей: атрибуты, связи и извлекаемые свойства (fetched properties). Извлечение значений из атрибута и запись в атрибут. Контекст и использование контекста. Постоянное (persistence store) и резервное (backing data store) хранилища. Контекст управляемых объектов и менеджер отмены. Создание и извлечение управляемых объектов. Разработка модели данных.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дневная форма получения образования с применением электронных средств обучения (ДО)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1.	Технологии управления, проектирования и разработки программного обеспечения	14			12		2	
1.1	Пользовательское окружение macOS, командный интерфейс и файловая система macOS для управления ИТ-проектами	2			2			Отчет
1.2	Управление кодом и документирование проекта в продуктовой разработке	2			2			Отчет
1.3	Многофайловое программирование, сборка и непрерывная интеграция приложений	2			4			Отчет
1.4	Разработка требований и спецификация ПО с применением UML	4					2	Собеседование

1.5	Разработка приложений и хранение данных в базе данных sqlite	2			4			Проект, контрольная работа
1.6	Управление ИТ-проектами	2						Собеседование
2.	Технологии разработки мобильных приложений для ОС iOS	22			18		4	
2.1	Введение в разработку мобильных приложений для операционной системы iOS	2						Собеседование
2.2	Базовые средства языка программирования Objective-C	2			2			Отчет
2.3	Расширенные средства языка программирования Objective-C	2			2			Отчет
2.4	Базовые возможности языка Swift	2			2			Отчет
2.5	Контейнерные и опциональные типы данных в языке Swift	2			2			Отчет, коллоквиум
2.6	Объектно-ориентированное программирование и расширенные возможности языка Swift	2			2			Отчет
2.7	Проектирование интерфейса, проектирование архитектуры и тестирование мобильных приложений	2					4	Собеседование
2.8	Графика, анимация, жесты и их обработка в мобильных приложениях	2			2			Отчет

2.9	Интернационализация, локализация и определение местоположения в мобильных приложениях	2			2			Отчет, контрольная работа
2.10	Хранение данных в приложениях для iOS	4			4			Проект
Итого		36			30		6	

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Казанский А. А. Разработка приложений на Swift 5.1 и SwiftUI с нуля. / А. А. Казанский. — СПб.: БХВ-Петербург, 2021. — 381 с.: ил.
2. Маран, М. М. Программная инженерия : учебное пособие для вузов / М. М. Маран. — 3-е изд., стер. — Санкт-Петербург: Лань, 2022. — 196 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/189470>.
3. Паттон Дж. Пользовательские истории. Искусство гибкой разработки ПО. / Дж. Паттон. — СПб.: Питер, 2018. — 286 с.: ил.
4. Усов. В. Swift. Основы разработки приложений под iOS, iPadOS и macOS. / В. Усов. — 6-ое изд., доп. и перераб. — СПб.: Питер, 2022. — 544 с.: ил.
5. Шоттс У. Командная строка Linux. Полное руководство. / У. Шоттс — 2-ое межд. изд. — СПб.: Питер, 2022. — 544 с.: ил.

Перечень дополнительной литературы

1. Вейн Ч. Swift подробно. / Ч. И. Вейн. — М: ДМК, 2019. — 422 с.: ил.
2. Коберн А. Современные методы описания функциональных требований к системам. / А. Коберн. — М: Лори, 2017. — 288 с.: ил.
3. Кон М. Agile. Оценка и планирование проектов. / М. Кон. — М: Альпина Паблицер, 2022. — 424 с.: ил.
4. Кон М. Пользовательские истории. Гибкая разработка программного обеспечения. / М. Кон. — М: Вильямс, 2019. — 256 с.: ил.
5. Лёве Д. Совершенный софт. / Д. Лёве. — СПб.: Питер, 2021. — 480 с.: ил.
6. Нойбург М. Программирование для iOS 7. Основы Objective-C, Xcode и Cocoa.: Пер. с англ. / М. Нойбург — М.: Вильямс, 2014. — 384 с.: ил.
7. Усов. В. Swift. Разработка приложений под iOS на основе фреймворка UIKit. / В. Усов. — Москва, 2021. — 492 с.: ил.
8. Черткова Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для академического бакалавриата, для студ. вузов, обуч. по инженерно-техническим напр. / Е. А. Черткова; "Высшая школа экономики", национальный исследовательский университет. - 2-е изд., испр. и доп. - Москва : Юрайт, 2017. - 168 с. : ил.
9. Allen G. The Definitive Guide to SQLite. / G. Allen, M. Owens. — Second Edition. — Apress, 2010. — 347 p.
10. A Swift Tour. [Электронный ресурс] / Apple Inc. — 2022. — Режим доступа: <https://docs.swift.org/swift-book/GuidedTour/GuidedTour.html>. — Дата доступа: 16.06.2022.
11. Chakon Sc., Straub B. Pro Git. [Электронный ресурс] / Apress. — Second Edition — 2022. — Режим доступа: <https://git-scm.com/book/ru/v2>. — Дата доступа: 20.06.2022.

12. Objective-C 2.0 Essentials. [Электронный ресурс] / Payload Media. — Second Edition — 2016. — Режим доступа: https://www.techotopia.com/index.php/Objective-C_2.0_Essentials. — Дата доступа: 14.06.2022.

13. SQL As Understood By SQLite. [Электронный ресурс] / SQLite Consortium. — 2022. — Режим доступа: <https://www.sqlite.org/lang.html>. — Дата доступа: 18.06.2022.

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: отчет по лабораторным работам с их устной защитой и оцениванием на основе проектного метода, проект.

Формой текущей аттестации по дисциплине «Технология программирования для мобильных приложений» учебным планом предусмотрен 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 «Технологии программирования для мобильных приложений» 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 отдаче 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%;
- отчет по лабораторной работе – 60%;
- собеседование – 5%;
- проект – 15 %.
- контрольная работа – 10%

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

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема № 1.4. Разработка требований и спецификация ПО с применением UML (2 часа)

Перечень вопросов:

1. Применение языка UML для проектирования спецификации ПО.
2. Поведенческие диаграммы.
3. Структурные диаграммы.
4. Проектирование функциональных и нефункциональных требований.
5. Структура спецификации программного обеспечения.

Примерный перечень заданий:

1. Разработать функциональные требования для приложения согласно индивидуальному заданию.
2. Сформулировать нефункциональные требования для приложения согласно индивидуальному заданию.
3. Спроектировать архитектуру и структуру приложения согласно индивидуальному заданию.

Форма контроля: собеседование.

Тема № 2.7. Проектирование интерфейса, проектирование архитектуры и тестирование мобильных приложений (4 часа)

Перечень вопросов:

1. Проектирование интерфейса мобильного приложения в Interface Builder и библиотека UIKit.

2. Применение AutoLayout для адаптации интерфейса под разные устройства.
3. Проектирование навигационных элементов интерфейса.
4. Проектирование интерфейса мобильного приложения с использованием библиотеки SwiftUI.
5. Применение архитектурного шаблона MVC для мобильных приложений.

Примерный перечень заданий:

1. Спроектировать интерфейс мобильного приложения с учетом размеров и ориентации устройств.
2. Реализовать многооконный интерфейс мобильного приложения.
3. Спроектировать архитектуру мобильного приложения на основе шаблона MVC.

Форма контроля: собеседование.

Рекомендуемая тематика коллоквиума

1. Коллоквиум «Технологии управления, проектирования и разработки приложений для ОС macOS и iOS».
2. Коллоквиум «Технологии проектирования и командной разработки мобильных приложений»

Примерная тематика контрольных работ

Контрольная работа 1. Управление кодом, документирование, проектирование и разработка приложений

Контрольная работа 2. Разработка приложений на языке Objective-C и Swift

Примерная тематика лабораторных работ

Лабораторная работа 1. Знакомство с командной строкой macOS и основные консольные команды.

Лабораторная работа 2. Управление кодом с помощью git и документирование проекта.

Лабораторная работа 3. Консольные приложения и их сборка.

Лабораторная работа 4. Проектирование и разработка консольных приложений с хранением данных в базе данных.

Лабораторная работа 5. Проектная работа. Проектирование требований и спецификации ПО и разработка приложения.

Лабораторная работа 6. Разработка консольных приложений на языках Objective-C и Swift.

Лабораторная работа 7. Проектирование интерфейса и разработка мобильных приложений на языке Swift и Objective-C.

Лабораторная работа 8. Разработка приложений для обработки графики и жестов на языке Swift и Objective-C.

Лабораторная работа 9. Разработка приложений с локализацией и применением Core Location и Core Data на языке Swift и Objective-C.

Лабораторная работа 10. Проектная работа. Проектирование, разработка и тестирование мобильного приложения для ОС iOS.

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 образовательных стандартов 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 экзамену

1. История развития и архитектура macOS.
2. Файловая система macOS и ее структура.
3. Системы контроля версий, их виды и базовые команды git.
4. Ветвление в git, внешние репозитории и метки.
5. Многофайловое программирование на C, модель КИС и сборка приложений
6. Библиотека SQLite, консоль и команды управления базой данных.
7. Библиотека SQLite, классификация запросов и их синтаксис.
8. Разработка приложений и хранение данных в sqlite.
9. Структурные диаграммы UML.
10. Поведенческие диаграммы UML.
11. Гибкие методологии управления проектами.
12. Системы управления проектами в стиле Kanban.
13. История развития iOS, архитектура и слои абстракции.
14. Язык программирования Objective-C: базовый синтаксис языка, числовые и строковые объекты.
15. Язык программирования Objective-C: коллекции на примере словарей и массивов, основные методы.
16. Язык программирования Objective-C: классы, область видимости, протоколы.
17. Язык программирования Objective-C: структура проекта согласно модели КИС на примере проекта.
18. Язык программирования Swift: базовые операторы, переменные и константы, правила объявления переменных и констант.
19. Язык программирования Swift: числовые и текстовые типы данных, приведение типов, операции и свойства.
20. Язык программирования Swift: логические значения и операторы, логический тип данных, псевдонимы типов, операторы сравнения и диапазона.
21. Язык программирования Swift: контейнерные типы данных, их классификация, характеристика и особенности.
22. Язык программирования Swift: кортежи, объявление кортежа, тип данных и взаимодействие с элементами кортежа.
23. Язык программирования Swift: диапазоны, операторы диапазона, базовые свойства и методы.

24. Язык программирования Swift: объявление массивов, типы данных, явное указание типа, доступ к элементам массива.
25. Язык программирования Swift: сравнение и слияние массивов, многомерные массивы, базовые свойства и методы массивов.
26. Язык программирования Swift: наборы, доступ и модификация набора, эквивалентность наборов.
27. Язык программирования Swift: словари, взаимодействие с элементами словаря, базовые свойства и методы словарей.
28. Язык программирования Swift: опциональный тип данных и извлечение опционального значения.
29. Язык программирования Swift: операторы if, switch, for, while, repeat while.
30. Язык программирования Swift: функции и объявление функций, аргументы функции и возвращаемое значение, возможности и функциональный тип.
31. Библиотека UIKit Framework: возможности, UI компоненты, примеры, их особенности, классы и свойства.
32. Библиотека UIKit Framework: табличное представление и представление-коллекция, классы и свойства.
33. Структура проекта Single View Application и согласно шаблону MVC.
34. Разработка интерфейса приложения в Interface Builder, макеты и автоматическая компоновка AutoLayout.
35. Разработка интерфейса приложения с применением SwiftUI.
36. Библиотека Core Graphics, геометрические типы данных и основные классы.
37. Анимация в iOS, анимация представлений и слоев, типы анимации
38. Интернационализация и локализация приложений для платформы iOS
39. Обработка различных видов жестов и касаний.
40. Использование списков свойств для хранения данных, файл настроек проекта и класс UserDefaults
41. Использование библиотеки SQLite в проектах на языке Swift
42. Использование библиотеки CoreData, сущности и управляемые объекты, использование контекста
43. Определение местоположения, диспетчер местоположения и делегат диспетчера местоположения
44. Проектирование, разработка и тестирование мобильных приложений в Xcode.

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)
Системное программирование	Кафедра технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 13 от 19.05.2022)
Программирование мобильных и встраиваемых систем	Кафедра технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения (протокол № 13 от 19.05.2022)

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ

на ____ / ____ учебный год

№№ Пп	Дополнения и изменения	Основание

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

Заведующий кафедрой

(ученая степень, звание)

(подпись)

(И. О. Фамилия)

УТВЕРЖДАЮ

Декан факультета

(ученая степень, звание)

(подпись)

(И. О. Фамилия)