

Легендарный курс разработки iOS приложений. 1-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Введение и переменные

- Что такое Swift, его история и преимущества, где применяется;
- Знакомство со средой разработки Xcode Playground;
- Переменные в Swift и основные типы данных;
- Понятия объявления и инициализация переменной;
- Отличия основных типов данных, получение информации о занимаемой памяти;
- Получение минимальных, максимальных и случайных значений;
- Вычисления в переменных, приведение типов и ошибки в вычислениях с числами с плавающей запятой.

Результат занятия: изучили переменные, узнали об основных типах данных.

Практическое задание: решение задач с использованием переменных.

День второй

Хранимые и вычисляемые переменные, геттеры и сеттеры

- Понятие хранимой переменной;
- Геттеры и сеттеры в хранимых переменных;
- Понятие вычисляемой переменной;
- Организация реактивного подхода при помощи вычисляемых переменных;
- Отличия между хранимыми и вычисляемыми переменными.

Результат занятия: изучили понятия хранимая и вычисляемая переменная, отличия между ними.

Практическое задание: решение задач с вычисляемыми переменными.

День третий

Фреймворк для построения интерфейса SwiftUI: объект Text

- Знакомство со средой разработки Xcode, создаем учебное приложение;
- Изучение объекта Text;
- Настройка базового внешнего вида текста (шрифт, размер, цвет, фон и т.п.);
- Инструменты расположения текста на экране, понятия стеков;
- Обработка динамического изменения размеров шрифта;
- Взаимодействие с дневным и ночным режимами.

Результат занятия: создали первое приложение, изучили работу объекта Text.

Практическое задание: создание экрана входа в приложение с использованием объекта Text.

День четвертый

Получение данных от пользователя: объект TextField в SwiftUI

- Изучение объекта TextField;
- Настройка внешнего вида TextField стандартными, не стандартными средствами;
- Перехват значений TextField и их передача другим объектам;
- Обертка State в переменных и ее особенности работы.

Результат занятия: изучили объект TextField, научились перехватывать его значения.

Практическое задание: создание экрана входа в приложение с использованием объекта TextField.

Легендарный курс разработки iOS приложений. 2-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Объект TextField: профессиональное использование, работа "под капотом"

- Анализ и корректировка данных на лету;
- Фокусирование: определяем активный TextField;
- Выбор экранной клавиатуры в зависимости от передаваемого типа данных;
- Объект SecureField для получения скрытых данных;
- Условия в Swift, конструкция if - else.

Результат занятия: углубленно изучили TextField и способы управления им.

Практическое задание: внедрение на экран входа проверки корректности пароля.

День второй

Кнопки в SwiftUI: объект Button

- Создание простой кнопки;
- Стандартные способы изменения внешнего вида кнопки;
- Создание кнопки со сложным пользовательским дизайном;
- Переменные состояний и взаимодействие с ними действий кнопки;
- Использование переключателей у переменных с логическим типом.

Результат занятия: изучили объект Button, научились создавать кнопки со сложным дизайном.

Практическое задание: внедрение кнопки на экран входа в приложение и проверка введенных данных по тапу.

День третий

Фигуры в SwiftUI и их использование для построения интерфейса

- Знакомство с геометрическими фигурами круг, прямоугольник, эллипс;
- Оформление внешнего вида фигур их комбинация в сложных объектах;
- Понятие "тень" и ее устройство, работа с тенями и размытием;
- Создание пользовательских фигур;
- Использование фигур в интерфейсе в качестве маски объекта.

Результат занятия: изучили работу фигур в интерфейсах SwiftUI.

Практическое задание: практика создания сложных комбинированных объектов.

День четвертый

Использование изображений в SwiftUI: объект Image

- Использование системных изображений и их оформление;
- Использование внешних растровых изображений и их настройка;
- Применение эффектов к изображениям;
- Объединение объектов с фигурами и изображениями;
- Использование векторных изображений.

Результат занятия: изучили работу с изображениями разного рода в SwiftUI.

Практическое задание: создание экрана с комбинированным использованием растровых изображений и фигур.

Легендарный курс разработки iOS приложений. 3-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Использование градиентов в SwiftUI

- Виды градиентов: линейные, радиальные и угловые градиенты;
- Mesh Gradient в iOS 18: понятие устройства;
- Создание простого линейного градиента. Распределение цветов;
- Использование Mesh Gradient в интерфейсах SwiftUI;
- Совмещение градиентов с другими модификаторами вида: прозрачность, размытие и т.п.

Результат занятия: познакомились с понятием градиента и инструментами для его создания в фреймворке SwiftUI.

Практическое задание: создание пользовательского интерфейса с плавными цветными переходами.

День второй

Практическое занятие: создание учебного приложения с профессиональным дизайном интерфейса

- Основные принципы верстки с использованием Figma;
- Наброски домашнего экрана на основе готового дизайна в Figma;
- Создание View, выполняющего работу TextField с пользовательским интерфейсом;
- Приемы для индивидуального поведения пользовательского TextField;
- Создание кнопок авторизации с пользовательским интерфейсом.

Результат занятия: Создали основу для первого экрана в учебном приложении с профессиональным дизайном.

Практическое задание: работа с версткой экрана приложения из Figma.

День третий

Практическое занятие: оптимизация кода View: улучшение кода домашнего экрана

- Приемы для сокращения объема основного кода View;
- Простая функция с параметрами в Swift;
- Переменные и функции в качестве View в SwiftUI;
- Создание и оптимизация View для кнопок авторизации;
- Улучшение TextField с пользовательским интерфейсом.

Результат занятия: изучили подходы для оптимизации кода интерфейса.

Практическое задание: продолжение работы с экраном учебного приложения на основе дизайна в Figma.

День четвертый

Практическое занятие: создание экрана авторизации

- Наброски экрана авторизации на основе готового дизайна в Figma;
- Улучшение TextField для ввода пароля с возможностью отображения скрытых символов в заданных ситуациях;
- Создание универсальной кнопки для перехода к следующему экрану.

Результат занятия: создали экран авторизации в практическом учебном приложении.

Практическое задание: внедрение на экран авторизации учебного приложения кастомизированного универсального TextField.

Легендарный курс разработки iOS приложений. 4-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Навигация в приложении: переходы между экранами и передача данных

- Навигация с UINavigationController в SwiftUI;
- Создание простой кнопки для перехода между экранами;
- Использование переменных состояний для перехода между экранами;
- UINavigationController в iOS 18 и выше;
- Передача данных между экранами, обертка свойства @Binding.

Результат занятия: научились создавать переходы между экранами и передавать данные между ними.

Практическое задание: внедрение перехода между экранами в приложении.

День второй

Визуализация навигации в SwiftUI: TabView и Toolbar

- Создание вкладок стандартными средствами, при помощи TabView;
- Настройка TabView и переходы между вкладками;
- Создание пользовательских вкладок с индивидуальным дизайном;
- Размещение кнопок на панели инструментов, объект Toolbar;
- Внедрение навигации и панели инструментов в приложение на экран авторизации.

Результат занятия: научились создавать вкладки для перехода между экранами и использовать Toolbar для размещения кнопок.

Практическое задание: создание приложения с несколькими экранами и переходами между ними при помощи TabView.

День третий

Классы в Swift: основная информация

- Что такое класс, объявление класса, понятие Self;
- Свойства и методы класса;
- Устройство инициализатора (конструктора) класса;
- Использование нескольких инициализаторов;
- Наследование классов, обязательные инициализаторы;
- Использование родительского инициализатора.

Результат занятия: познакомились с основными возможностями класса в Swift.

Практическое задание: решение задач для закрепления темы классов.

День четвертый

Классы в Swift: расширенный уровень, парадигма ООП

- Деинициализатор в классах;
- Понятия ООП: наследование, инкапсуляция, полиморфизм;
- Уровни (модификаторы) доступа в классах;
- Использование геттеров и сеттеров для управления доступом к свойствам класса;
- Использование одного и того же интерфейса для разных типов объектов.

Результат занятия: познакомились с парадигмой ООП и ее реализацией при помощи классов в Swift.

Практическое задание: решение задач для закрепления темы ООП.

Легендарный курс разработки iOS приложений. 5-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Структуры в Swift, отличия структур от классов

- Объявление структур и их инициализация;
- Понятия value type и reference type;
- Отличия структуры от класса;
- Циклические ссылки при использовании классов.

Результат занятия: изучили структуры в Swift и определили разницу с классами.

Практическое задание: решение задач для закрепления темы структур.

День второй

Паттерн проектирования MVVM

- Для чего применяются паттерны проектирования;
- Понятие паттерна MVVM;
- Классическая реализация MVVM при помощи класса;
- Реализация MVVM с использованием реактивного подхода в SwiftUI;
- Обертки ObservableObject, StateObject, ObservableObject, Published;
- Observable и ObservationTracking в iOS 17 и выше.

Результат занятия: научились использовать паттерн MVVM в SwiftUI.

Практическое задание: решение задач на тренировку реализации MVVM.

День третий

Практическое занятие: рефакторинг учебного приложения, реализация паттерна MVVM

- Рефакторинг учебного проекта;
- Создание ViewModel для домашнего экрана и экрана авторизации;
- Передача действий между View с использованием безымянной функции;
- Разнесение свойства и логику в соответствии с правилами MVVM.

Результат занятия: произвели рефакторинг учебного проекта под паттерн MVVM.

Практическое задание: рефакторинг собственного проекта под паттерн MVVM.

День четвертый

Коллекции в Swift: Массивы

- Объявление и инициализация массивов в Swift;
- Доступ к элементам массива;
- Добавление, удаление и изменение элементов;
- Основные свойства массивов;
- Заполнение массивов повторяющимися данными;
- Сортировка массивов.

Результат занятия: изучили базовую информацию о массивах в Swift.

Практическое задание: решение задач для закрепления темы массивов.

Легендарный курс разработки iOS приложений. 6-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Циклы for, while и forEach

- Знакомство с синтаксисом и принципом работы цикла for;
- Цикл для перебора диапазона чисел и коллекций;
- Цикл while;
- Перебор коллекций в SwiftUI для генерации View при помощи цикла forEach.

Результат занятия: изучили работу цикла for в Swift и forEach в SwiftUI.

Практическое задание: решение задач для закрепления понимания циклов.

День второй

Перечисления в Swift: сила и мощь перечислений enums

- Что такое перечисления и их преимущества в Swift;
- Определение и использование перечислений;
- Перебор членов перечисления: конструкция switch;
- Ассоциированные значения в enum;
- Различные типы кейсов, raw values;
- Переменные и функции в перечислениях.

Результат занятия: изучили перечисления в Swift и их возможности.

Практическое задание: решение задач для закрепления темы перечислений.

День третий

Практическое занятие: рефакторинг учебного приложения

- Создание Model в учебном приложении;
- Написание структуры компании для кнопок авторизации;
- Использование перечислений для хранения иконок для кнопок авторизации;
- Создание универсального View для генерации кнопки авторизации;
- Оптимизация кода приложения, покрытие универсальными View.

Результат занятия: завершили рефакторинг проекта под паттерн MVVM.

Практическое задание: рефакторинг учебного приложения.

День четвертый

Анимации в SwiftUI

- Выполнение анимации при изменении свойств объекта;
- Easy анимации;
- Spring анимации;
- Анимации переходов.

Результат занятия: изучили работу анимаций в SwiftUI.

Практическое задание: создание экрана входа в приложение с применением анимации объектов на старте.

Легендарный курс разработки iOS приложений. 7-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Практическое занятие:
экран создания учетной записи в учебном приложении

- Проектирование нового экрана по принципам MVVM;
- Создание простой базы данных учетных записей;
- Настройка авторизации на основе сопоставления данных;
- Внедрение анимации.

Результат занятия: завершили создание первого учебного приложения.

Практическое задание: внедрение возможности авторизации по имеющимся записям в учебное приложение.

День второй

Использование жестов в SwiftUI

- Обработка жеста касания;
- Жест "щипок" и обработка его событий;
- Жест "поворота" объекта и обработка его событий.

Результат занятия: научились использовать разные жесты в приложениях, использующих SwiftUI.

Практическое задание: создание экрана, реализующего аналог ленты соцсети с возможностью просмотра фотографий при помощи жестов.

День третий

Производительные сетки: LazyVGreed, lazyHGreed

- Понятие Lazy;
- Размещение объектов на экране при помощи LazyVGreed и LazyHGreed;
- Анимация поворота объектов;
- Кастомные модификаторы в SwiftUI.

Результат занятия: научились работать с производительными сетками LazyVGreed и LazyHGreed и анимировать повороты объектов по тапу.

Практическое задание: Создание экрана с массивом данных, размещенным при помощи LazyVGreed.

День четвертый

Подготовка нового приложения:
игра "Крестики-нолики" с искусственным интеллектом

- Создание нового проекта на основе паттерна MVVM;
- Создание игрового поля на главном экране;
- Реализация обработки тапов по клеткам;
- Анимация поворота клетки по тапу.

Результат занятия: подготовили домашний экран игры крестики-нолики.

Практическое задание: обновление внешнего вида клетки по тапу.

Легендарный курс разработки iOS приложений. 8-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Опциональные типы данных в Swift

- Что такое опционалы и для чего они необходимы;
- Методы извлечения значения из опционала;
- Оператор объединения nil;
- Примеры использования опционалов.

Результат занятия: изучили опционалы и методы их извлечения.

Практическое задание: решение задач для закрепления понимания опционалов.

День второй

Реализация игры "Крестики-нолики": часть 2

- Обработка тапов по полю;
- Индивидуальная анимация для хода игрока и для хода компьютера;
- Создание функции, определяющей конец игры;
- Создание функции, определяющей победу.

Результат занятия: реализовали анимации по тапу и функции, определяющие завершение игры.

Практическое задание: размышление над логикой определения лучшего хода.

День третий

Работа уведомлений в SwiftUI

- Вызов уведомления (Alert) по требованию;
- Настройка внешнего вида Alert;
- Alert с индивидуальным дизайном.

Результат занятия: Изучили работу уведомлений в SwiftUI.

Практическое задание: Практика использования уведомлений в приложении.

День четвертый

Рекурсивные функции в Swift

- Основная идея рекурсивной функции;
- Вычисление факториала числа при помощи рекурсивной функции;
- Вычисление суммы элементов массива при помощи рекурсивной функции;

Результат занятия: изучили метод рекурсии в функциях.

Практическое задание: решение задач для усвоения темы рекурсии.

Легендарный курс разработки iOS приложений. 9-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Теория игр: алгоритм мини-макс

- Немного о теории игр: максимизатор и минимизатор, игровые деревья;
- Принципы работы алгоритма мини-макс, рекурсивное дерево игры;
- Основные этапы работы алгоритма мини-макс и связь с ИИ.

Результат занятия: познакомились с алгоритмом мини-макс, узнали об игровых деревьях и применении его в ИИ.

Практическое задание: решение задач для усвоения темы.

День второй

Реализация игры "Крестики-нолики": часть 3

- Реализация рекурсивной функции на основе алгоритма мини-макс;
- Создание функции для нахождения лучшего хода;
- Внедрение уведомления об окончании игры;
- Создание анимации для начала новой игры.

Результат занятия: завершили реализацию логики поиска ходов, добавили уведомления и дополнительную анимацию.

Практическое задание: индивидуальная анимация для начала новой игры, после уведомления об окончании игры.

День третий

Реализация игры "Крестики-нолики": часть 4

- Задачи по доработке интерфейса;
- Добавление возможности выбора символа игрока;
- Реализация подсветки выигрышной комбинации;
- Счетчик очков выигрышей и кнопки отмены хода и начала новой игры.

Результат занятия: завершили создание игры крестики нолики с искусственным интеллектом.

Практическое задание: доработка игры системой ачивок за достижения в игре.

День четвертый

Введение в UIKit: знакомство со Storyboard

- Знакомство с фреймворком для построения интерфейса UIKit;
- Базовые сведения о Storyboard и ViewController;
- Размещаем на экране TextField, Label и Button;
- Привязка объектов к ViewController.

Результат занятия: Познакомились со Storyboard, узнали о размещении объектов и их привязке к ViewController.

Практическое задание: Создание собственного проекта с использованием label, textField и button.

Легендарный курс разработки iOS приложений. 10-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Протоколы в Swift

- Абстракция в ООП;
- Протоколы с изменяемыми свойствами;
- Требования к методам, протоколы и классы;
- Протоколы с ассоциированными типами.

Результат занятия: изучили работу протоколов и их применимость в Swift.

Практическое задание: решение задач для понимания протоколов.

День второй

Расширения в Swift

- Для чего нужны расширения и что можно с их помощью добавить;
- Добавление методов и вычисляемых свойств;
- Добавление инициализаторов;
- Расширения и вложенные типы;
- Конформность к протоколам.

Результат занятия: изучили, что такое расширения в Swift и как их применять.

Практическое задание: решение задач для понимания расширений.

День третий

UIKit: Навигация между экранами и передача данных

- UINavigationController и перемещение между ViewController;
- Настройка перехода в UIStoryboardSegue;
- Передача данных при помощи делегата;
- Передача данных при помощи клоужера, базовое понятие клоужера.

Результат занятия: Узнали как произвести переход между ViewController, как передавать между ними данные.

Практическое задание: Создание приложения с несколькими экранами и переходами между ними.

День четвертый

UIKit: Управление объектами в коде, AutoLayout

- Что такое констрейнт в Storyboard;
- Управление констрейнтами при помощи кода;
- Работа с размерами текущего View при помощи кода;
- Основные методы работы с текстовыми данными.

Результат занятия: научились управлять объектами на экране в Storyboard и при помощи кода.

Практическое задание: создание экрана с использованием разных объектов и управление ими при помощи констрейнтов.

Легендарный курс разработки iOS приложений. 11-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Введение в UIKit: Создание объектов на экране без помощи Storyboard

- Основные принципы создания объектов на экране без помощи Storyboard;
- Объекты Label, Button, Constraint;
- Создание объектов при помощи кода;
- Настройка свойств объектов при помощи кода.

Результат занятия: изучили, как размещать объекты на экране кодом, без использования Storyboard.

Практическое задание: создание экрана без помощи Storyboard.

День второй

Введение в UIKit: TableView часть 1

- Для чего нужны таблицы в UIKit, объект TableView;
- Настройка TableView при помощи Storyboard;
- Передача данных в ячейку для разных версий iOS;
- Функция TableView.

Результат занятия: изучили, как использовать таблицы, созданные при помощи Storyboard.

Практическое задание: экран с использованием таблицы и передача данных в нее при помощи textField.

День третий

Введение в UIKit: TableView часть 2

- Создание TableView без помощи Storyboard;
- Настройка вида таблицы при помощи кода;
- Использование свайпов в TableView;
- Обработка свайпов, вызов и настройка простого Alert.

Результат занятия: изучили, как создать таблицу при помощи кода и настроить использование свайпов.

Практическое задание: создание экрана, копирующего три секции меню настроек iOS.

День четвертый

Введение в UIKit: TableView часть 3

- Передача данных в TableView созданный кодом на другой экран;
- Ячейка TableView с индивидуальным дизайном;
- Верстка ячейки с индивидуальным дизайном при помощи кода.

Результат занятия: изучили, как производить верстку ячеек с индивидуальным дизайном при помощи кода.

Практическое задание: создание дополнительных секций на экране, копирующем настройки iOS.

Легендарный курс разработки iOS приложений. 12-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Экраны запуска в UIKit и SwiftUI: Готовим приложение "Тетрис" с классическим дизайном

- Для чего необходимы экраны запуска;
- Создание Launch Screen при помощи Storyboard;
- Создание Launch Screen в проектах на SwiftUI;
- Подходы для использования анимации в Launch Screen в SwiftUI.

Результат занятия: изучили, как создать и анимировать экран запуска в SwiftUI.

Практическое задание: создание экрана запуска в готовящемся приложении "Тетрис".

День второй

Матрицы в Swift: часть 1

- Матрицы и их применение в программировании;
- Доступ и изменение элементов матрицы;
- Объединение матриц, сложение матриц;
- Сложение матриц разного размера.

Результат занятия: научились создавать, объединять и складывать матрицы разного размера в Swift.

Практическое задание: решение задач, для закрепления понимания матриц.

День третий

Матрицы в Swift: часть 2

- Транспонирование матрицы;
- Поворот матрицы по часовой стрелке и против;
- Создание фигуры тетриса на основе матриц.

Результат занятия: научились поворачивать, транспонировать матрицы.

Практическое задание: решение задач, для закрепления понимания матриц.

День четвертый

Сабскрипты в Swift

- Для чего нужны сабскрипты, синтаксис сабскриптов;
- Сабскрипты с несколькими параметрами;
- Сабскрипты только для чтения;
- Сабскрипты и параметры по умолчанию;
- Использование сабскриптов для доступа к элементам матрицы.

Результат занятия: изучили, что такое сабскрипты и как их использовать.

Практическое задание: решение задач, для закрепления понимания сабскриптов.

Легендарный курс разработки iOS приложений. 13-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Словари в Swift

- Что такое словари в Swift;
- Создание словаря и добавление в него элементов;
- Доступ к элементам словаря и итерация по ним;
- Свойства и методы словарей.

Результат занятия: изучили, что такое словарь и как его использовать в Swift.

Практическое задание: решение задач для закрепления понимания словарей.

День второй

Кортежи в Swift

- Что такое кортежи в Swift;
- Создание кортежа и обращение к его элементам;
- Создание кортежа из функции;
- Методы кортежей.

Результат занятия: изучили, что такое кортеж и как его использовать в Swift.

Практическое задание: решение задач для закрепления понимания кортежей.

День третий

Реализация игры "Тетрис" на SwiftUI: часть 1

- Создание шаблона проекта игры на основе MVVM;
- Создание типа фигуры и отрисовка универсального блока;
- Проектирование фигур и колодеца на основе матриц;
- Программирование функции, объединяющей матрицы колодца и фигуры.

Результат занятия: создали новое приложение, которое будет реализовывать игру "Тетрис", добавили функции отрисовки фигур на основе их матриц.

Практическое задание: организация движения фигуры по тапу на кнопку.

День четвертый

Создание и использование таймера в SwiftUI

- Для чего используются таймеры в программировании;
- Создание, настройка таймера и управление состоянием интерфейса;
- Таймер с остановкой и перезапуском;
- Использование Timer.publish, базовое понимание даты в Swift и использование календаря;
- Таймер с обратным отсчетом счетчика и к дате.

Результат занятия: научились создавать разные таймеры в интерфейсе SwiftUI .

Практическое задание: продумать варианты движения фигуры используя таймер.

Легендарный курс разработки iOS приложений. 14-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Реализация игры "Тетрис" на SwiftUI: часть 2

- Кодирование спуска фигуры по таймеру;
- Создание скриншотов View программно в SwiftUI;
- Реализация функции, создающую скриншот колодца;
- Оптимизация работы алгоритмов отрисовки фигур при помощи скриншотов.

Результат занятия: научили фигуру двигаться согласно заданного таймера, оптимизировали скорость выполнения программы.

Практическое задание: продумать, каким образом можно запрограммировать управление работой фигуры.

День второй

Безымянные функции (клоужеры) в Swift: часть 1

- Что такое клоужер (замыкание), основной синтаксис;
- Простой клоужер с параметрами и без;
- Клоужер возвращающий значение, обращение к параметрам клоужера по индексу;
- Захват значений клоужером из окружающего контекста.

Результат занятия: изучили, базовые понятия клоужера и как его применять в Swift.

Практическое задание: решение задач, для закрепления понимания клоужеров.

День третий

Безымянные функции (клоужеры) в Swift: часть 2

- Клоужер как аргумент функции;
- Возврат клоужера из функции;
- Автозамыкания;
- Убегающие клоужеры.

Результат занятия: изучили, что такое убегающий клоужер, использование клоужера в качестве результата функции и в качестве ее аргумента.

Практическое задание: решение задач, для закрепления понимания клоужеров.

День четвертый

Функции высшего порядка в Swift. Гибкая работа с коллекциями данных: часть 1

- Фильтрация элементов коллекции по условию;
- Преобразование каждого элемента коллекции в новое значение;
- Объединение всех элементов коллекции в единое значение;
- Сортировка элементов коллекции по определенному критерию.

Результат занятия: изучили несколько популярных функций высшего порядка в Swift.

Практическое задание: решение задач, для закрепления понимания изученных функций высшего порядка.

Легендарный курс разработки iOS приложений. 15-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Функции высшего порядка в Swift. Гибкая работа с коллекциями данных: часть 2

- Преобразование элементов коллекции и удаление nil;
- Преобразование элементов коллекции и объединение результатов в одну коллекцию;
- Функции поиска заданных элементов в коллекции;
- Применение указанного действия к каждому элементу коллекции.

Результат занятия: изучили еще несколько важных функций высшего порядка.
Практическое задание: решение задач, для закрепления понимания изученных функций высшего порядка.

День второй

Создание интерактивных интерфейсов в SwiftUI: DragGesture - отслеживание движения пальца

- Добавление жеста в SwiftUI View;
- Отслеживание изменений и завершения жеста;
- Жесты с несколькими действиями, одновременное отслеживание нескольких жестов;
- Динамическое обновление состояния.

Результат занятия: изучили работу и использование DragGesture в SwiftUI.
Практическое задание: Создание экрана с объектами, использующими DragGesture.

День третий

Реализация игры "Тетрис" на SwiftUI: часть 3

- Программирование перемещения фигуры за пальцем;
- Реализация поворота фигуры по тапу;
- Программирование падения фигуры по свайпу вниз;
- Создание функции рандомного выбора фигуры.

Результат занятия: внедрили в игру отслеживание движений пальцем для перемещения фигуры по экрану.
Практическое задание: Удаление лишних вспомогательных элементов на экране.

День четвертый

Реализация игры "Тетрис" на SwiftUI: часть 4

- Определение логики поиска рядов для очистки;
- Создание функции для поиска заполненных рядов;
- Реализация функции удаления заполненного ряда из матрицы колодца.

Результат занятия: добавили в игру функционал, удаляющий заполненные фигурами ряды.
Практическое задание: создание функции, определяющей конец игры.

Легендарный курс разработки iOS приложений. 16-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Многопоточность. Фреймворк Grand Central Dispatch: часть 1

- Что такое очередь, необходимость многопоточности в приложениях;
- Очереди (Dispatch Queues) в Grand Central Dispatch (GCD);
- Асинхронные и синхронные задачи;
- Потоки (Threads).

Результат занятия: изучили, как использовать очереди в синхронных и асинхронных потоках в Swift .

Практическое задание: решение задач для закрепления понимания основ многопоточности.

День второй

Многопоточность. Фреймворк Grand Central Dispatch: часть 2

- Пользовательские очереди;
- Выполнение задач с задержкой;
- Группировка задач;
- Использование GCD в интерфейсе SwiftUI.

Результат занятия: научились создавать пользовательские очереди, узнали, как выполнять задачи с задержкой и как их группировать.

Практическое задание: решение задач для закрепления понимания основ многопоточности.

День третий

Механизмы конкурентности в Swift 5.5: часть 1

- Асинхронные функции (Async Functions);
- Вызов асинхронных функций;
- Конкурентные функции (Concurrent Functions);
- Обработка ошибок.

Результат занятия: научились создавать и работать с синхронными и асинхронными функциями.

Практическое задание: решение задач для закрепления основ механизма конкурентности.

День четвертый

Механизмы конкурентности в Swift 5.5: часть 2

- Подход к организации асинхронного кода (Structured Concurrency);
- Решение проблем синхронизации при помощи Actors;
- Основные отличия между Class и Actor;
- Аннотация @MainActor, как гарантия безопасности главного потока.

Результат занятия: изучили подход к организации асинхронного кода, что такое Actor и в каких случаях применять аннотацию @MainActor.

Практическое задание: решение задач для закрепления основ механизма конкурентности.

Легендарный курс разработки iOS приложений. 17-й модуль

Цель курса: Получить глубокое понимание языка программирования Swift, научиться создавать сложные интерфейсы для iOS приложений при помощи фреймворка SwiftUI

Программа курса:

День первый

Реализация игры "Тетрис" на SwiftUI: часть 5

- Определение принципы анимации динозавра на заставке;
- Определение рандомного порядка для поворота динозавра и движения лапами;
- Создание функции анимации динозавра, с использованием асинхронных подходов;
- Внедрение анимированного динозавра на экран, до запуска игры.

Результат занятия: разместили анимированного динозавра на главный экран до момента старта игры.

Практическое задание: продумать логику очистки колодца, при помощи анимации с использованием механизма конкуренции, после окончания игры.

День второй

Реализация игры "Тетрис" на SwiftUI: часть 6

- Определение принципов и логики очистки колодца после окончания игры;
- Создание функцию для анимированного заполнения и очистки колодца рядами блоков, используя механизмы конкуренции;
- Внедрение анимации очистки колодца при наступлении окончания игры.

Результат занятия: создали анимацию для выполнения очистки колодца по окончанию игры.

Практическое задание: определение логики набора очков и повышения уровней.

День третий

Работа со звуком. Фреймворк AVKit

- Преимущества использования высокоуровневого фреймворка Apple AVKit;
- Основные компоненты AVKit для работы с аудио;
- Создание аудиосессии и аудиоплеера и их использование в SwiftUI;
- Воспроизведение звуков из сети;
- Использование таймера для отображения текущего времени аудио.

Результат занятия: научились воспроизводить звуки в приложениях на SwiftUI.

Практическое задание: создание плеера для воспроизведения звуков в приложении.

День четвертый

Реализация игры "Тетрис" на SwiftUI: часть 7

- Реализация воспроизведения звуков при движении фигуры и при старте игры;
- Реализация возможности подсчета очков игры;
- Внедрение отображения уровня и функции ускорения игры при его повышении;
- Отображение кнопки паузы игры.

Результат занятия: завершили реализацию игры тетрис с классическим дизайном.

Практическое задание: внедрение отображения времени на игровом экране.