

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет Радиотехнический
Кафедра РТ5

Курс «Программирование на основе классов и шаблонов»

Отчет по домашнему заданию
«Создание первичного To-Do листа на языке Swift»

Выполнил:

студент группы РТ5-31Б:
Топорин Б.Г

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю.Е.

Подпись и дата:

Москва, 2023

Описание задания

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транпилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ.
Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

Основная часть

Swift - это мощный, современный и выразительный язык программирования, созданный компанией Apple для разработки приложений под платформы iOS, macOS, watchOS и tvOS. Он был представлен в 2014 году и быстро стал одним из предпочтительных языков для мобильной разработки благодаря своей чистоте, эффективности и безопасности.

Вот несколько ключевых характеристик Swift:

1. Ясный и читаемый синтаксис: Swift разработан с упором на читаемость и ясность кода. Его синтаксис узнаваем и дружелюбен к разработчикам.

2. **Безопасность:** Swift внедряет множество функций безопасности, чтобы предотвратить ошибки программирования. Он предлагает нулевое значение для опциональных типов данных, обработку ошибок, контроль доступа и автоматическую управляемую память.
3. **Высокая производительность:** Swift был оптимизирован для работы на платформах Apple, что обеспечивает высокую производительность и быстрое выполнение кода.
4. **Интероперабельность:** Swift легко взаимодействует с Objective-C, что обеспечивает совместимость с существующими проектами на платформах Apple.
5. **Поддержка сообщества:** Swift - это open-source язык, что позволяет активному сообществу разработчиков вносить свой вклад в его развитие. Это способствует появлению множества библиотек, фреймворков и инструментов, которые обогащают экосистему Swift.
6. **Инструменты и IDE:** Swift использует Xcode в качестве основной среды разработки. Xcode предоставляет разработчикам богатый инструментарий для отладки, профилирования и создания интерфейсов.
7. **Универсальность:** Swift может быть использован для создания разнообразных приложений, начиная от мобильных приложений на iOS и Android (с использованием Kotlin Multiplatform), до серверных приложений и даже приложений для встраиваемых систем.

Swift олицетворяет прогрессивный подход к разработке программного обеспечения, учитывая современные требования к безопасности, производительности и удобству использования.

В качестве демонстрации возможностей языка я решил написать простенькое приложение для отслеживания задач. Его реализация представлена ниже:

```
import SwiftUI
```

```
struct ContentView: View {
```

```
    // Массив для хранения задач
```

```
    @State private var tasks: [String] = []
```

```
    // Переменная для хранения текста новой задачи
```

```
    @State private var newTask: String = ""
```

```
    var body: some View {
```

```
        // Определение основного вида приложения внутри NavigationView
```

```
        NavigationView {
```

```
            VStack {
```

```
                // Вид списка (List) для отображения задач
```

```
                List {
```

```
                    // Итерация по массиву задач для создания текстовых представлений
```

```
                    ForEach(tasks, id: \.self) { task in
```

```
                        Text(task)
```

```
                    }
```

```
                    // Возможность удаления задачи свайпом
```

```
                    .onDelete(perform: deleteTask)
```

```
                }
```

```
                // Вид для ввода новой задачи и кнопки для ее добавления
```

```
                HStack {
```

```
                    // TextField для ввода новой задачи
```

```
                    TextField("New Task", text: $newTask)
```

```
                    .textFieldStyle(RoundedBorderTextFieldStyle())
```

```

        // Кнопка для добавления задачи
        Button(action: addTask) {
            Text("Add")
        }
    }
    .padding()
}

// Заголовок навигационной панели
.navigationTitle("To-Do List")
}
}

// Функция для добавления новой задачи в массив
func addTask() {
    if !newTask.isEmpty {
        tasks.append(newTask)
        newTask = ""
    }
}

// Функция для удаления задачи из массива по индексу
func deleteTask(at offsets: IndexSet) {
    tasks.remove(atOffsets: offsets)
}

}

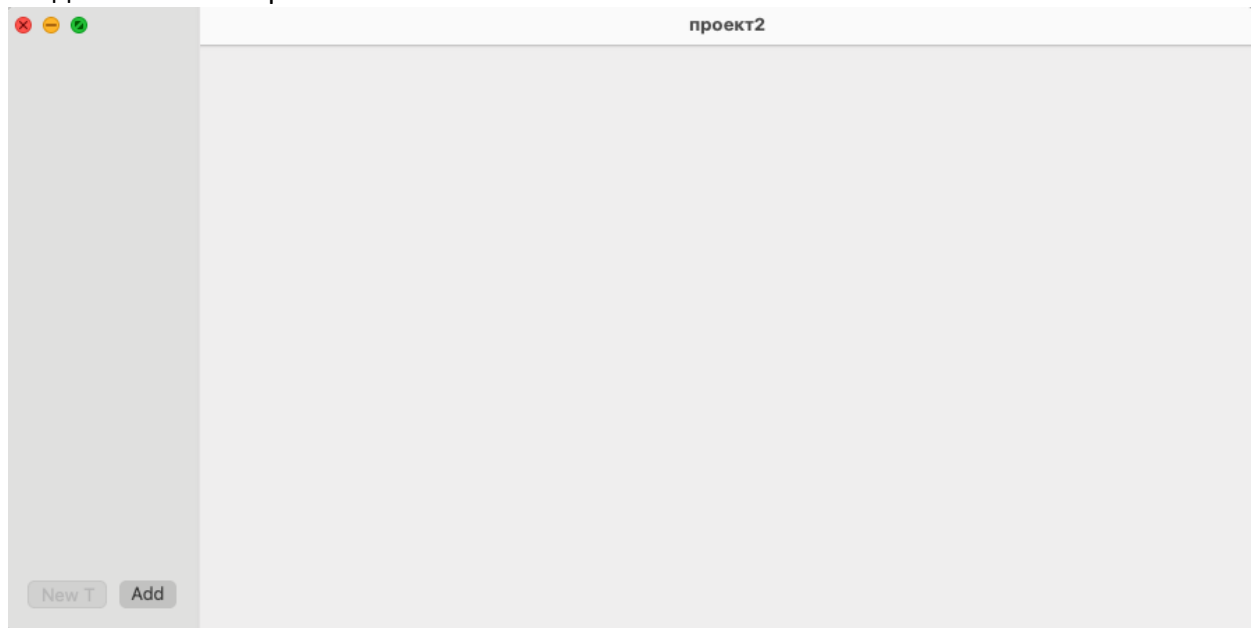
struct ContentView_Previews: PreviewProvider {
    // Вид предварительного просмотра
    static var previews: some View {
        ContentView()
    }
}

```

}
}

В результате получаем следующее:

1. Модальное окно приложения



2. Написание задачи и добавление ее в список задач

