

**Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики**

Инструментальные средства разработки ПО
Лабораторная работа №5
Xcode source code extension

Выполнил:
Лукьянов Кирилл М32051

2021г

Описание лабораторной работы:

<https://github.com/bigfirestart/ZvaroExt>

Создадим расширение, которое в данном файле поменяет устаревший синтаксис замыканий на более новый без круглых скобок

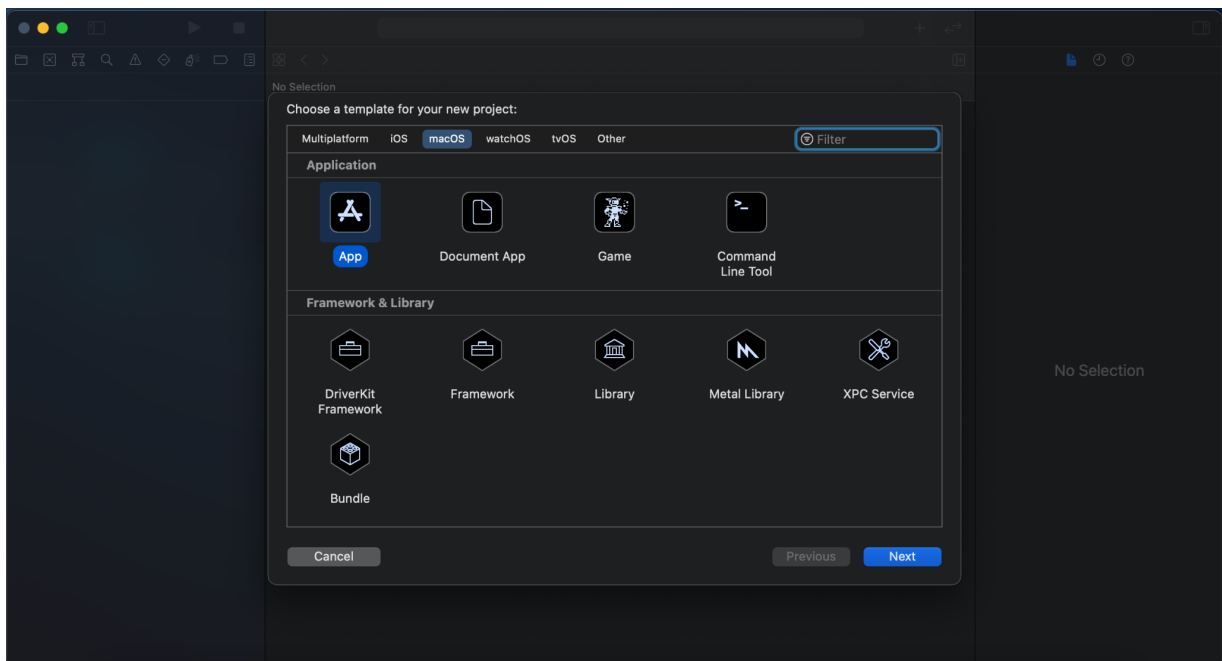
Расширения Xcode довольно ограничены и всё что они могут, это получить и изменить содержимое конкретного файла, так же они не могут работать в фоновом режиме, это накладывает множество ограничений на наши расширения

Далее начинается множество нюансов, связанных с крайне ограниченной базой знаний и документацией от Apple

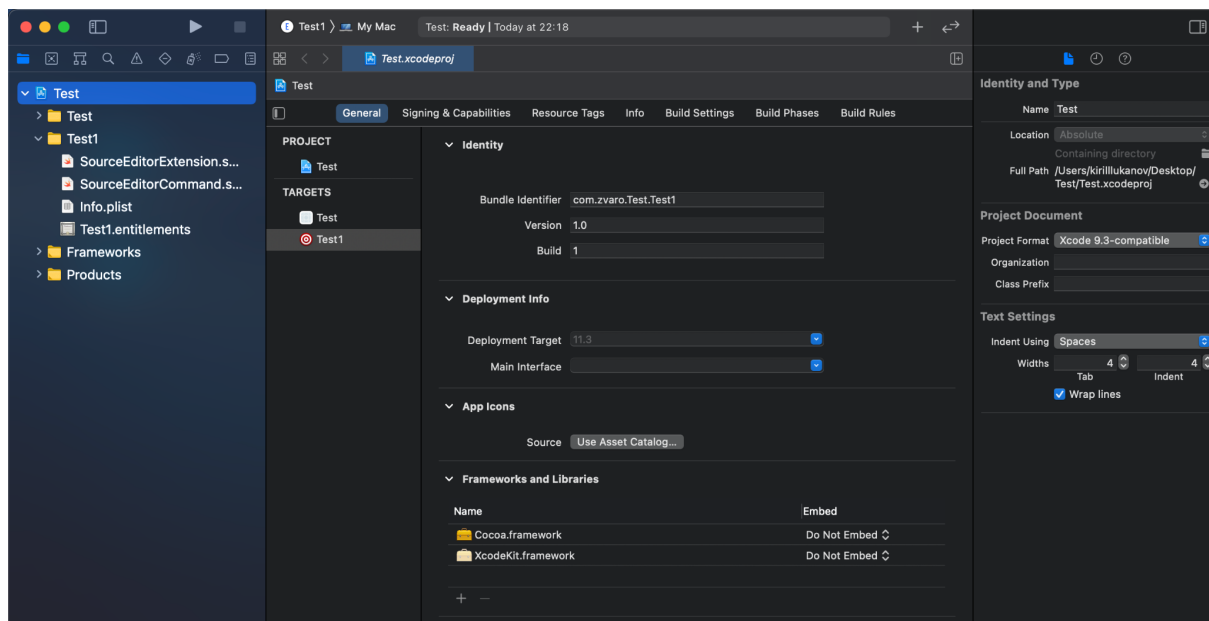
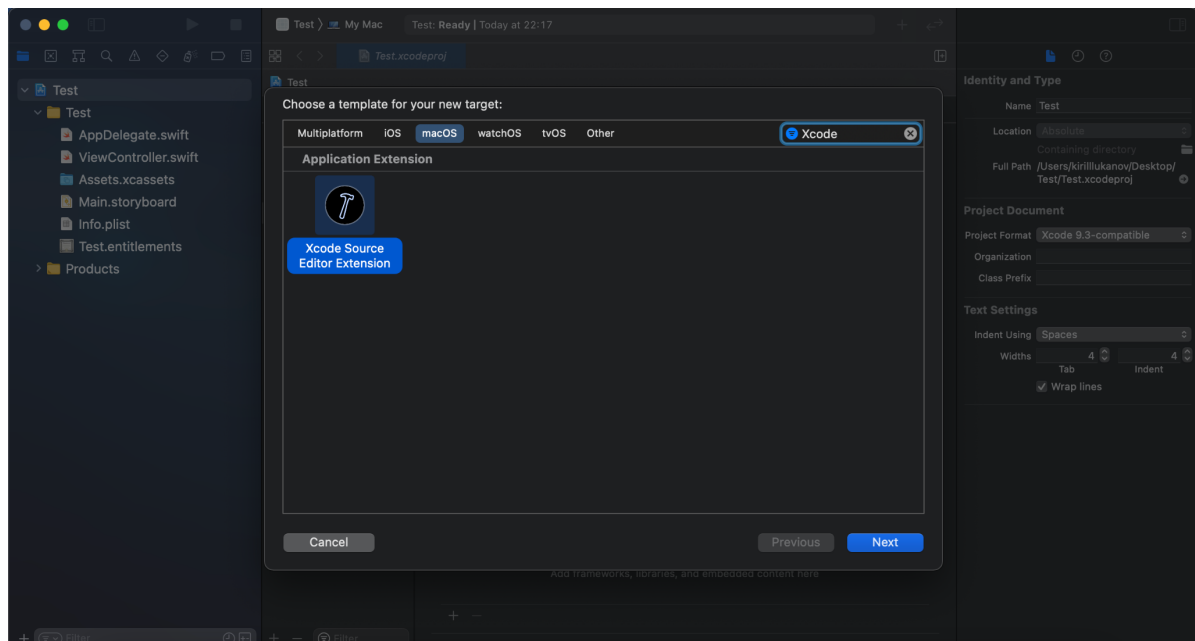
Лабораторная работы была выполнена на Xcode 12.5 (13.0 beta совершенно отказывается работать с кастомными расширениями и не видит их)

Ход работы

Создадим macOS приложение



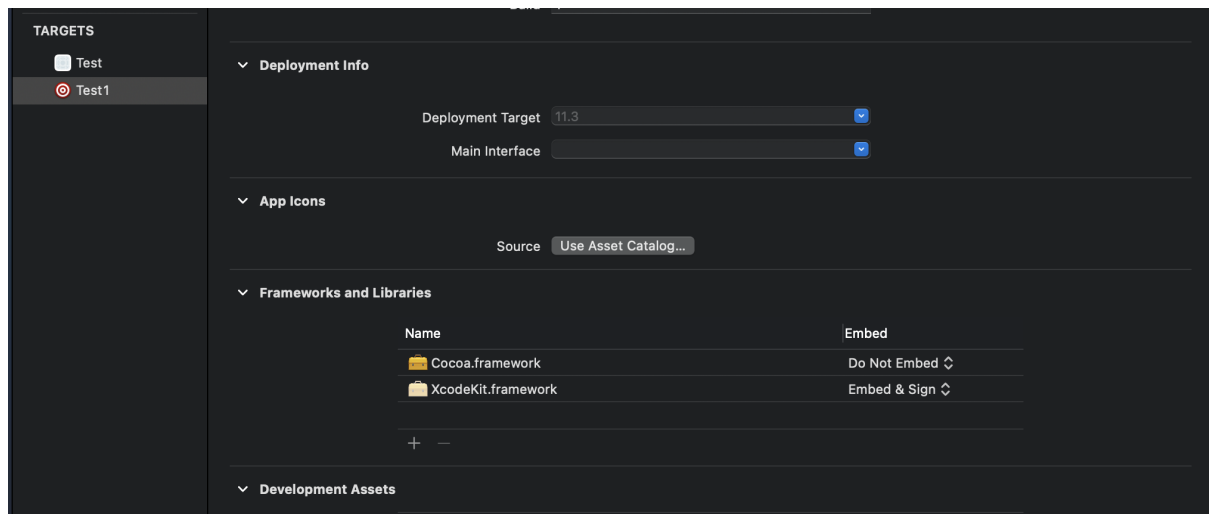
И добавим в него новый таргет Xcode Source Editor Extension



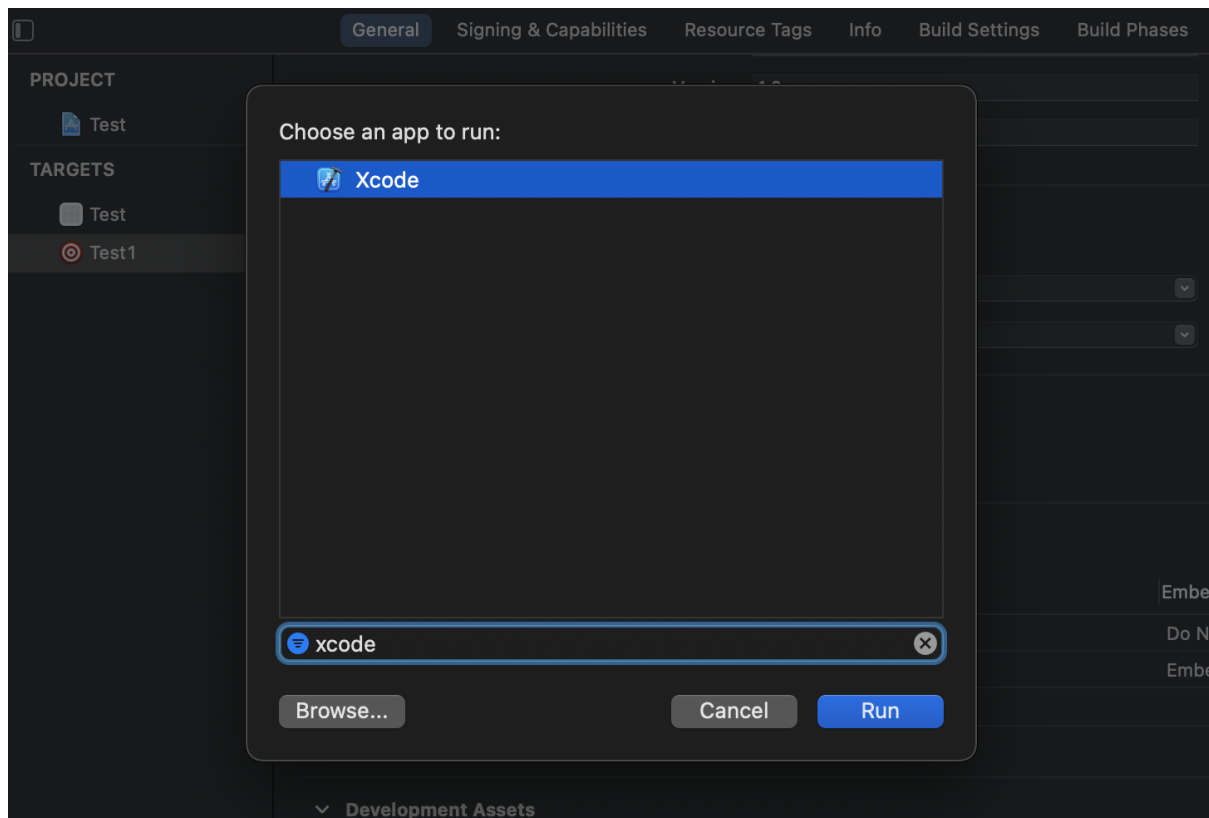
Xcode сгенерирует нам 4 новых файла

- SourceEditorExtension - Класс наследник XCSourceEditorCommand
- SourceEditorCommand - Класс дополнений расширения
- Info.plist - основные настройки расширения
- {PROJ_NAME}.entitlements - файл подписи

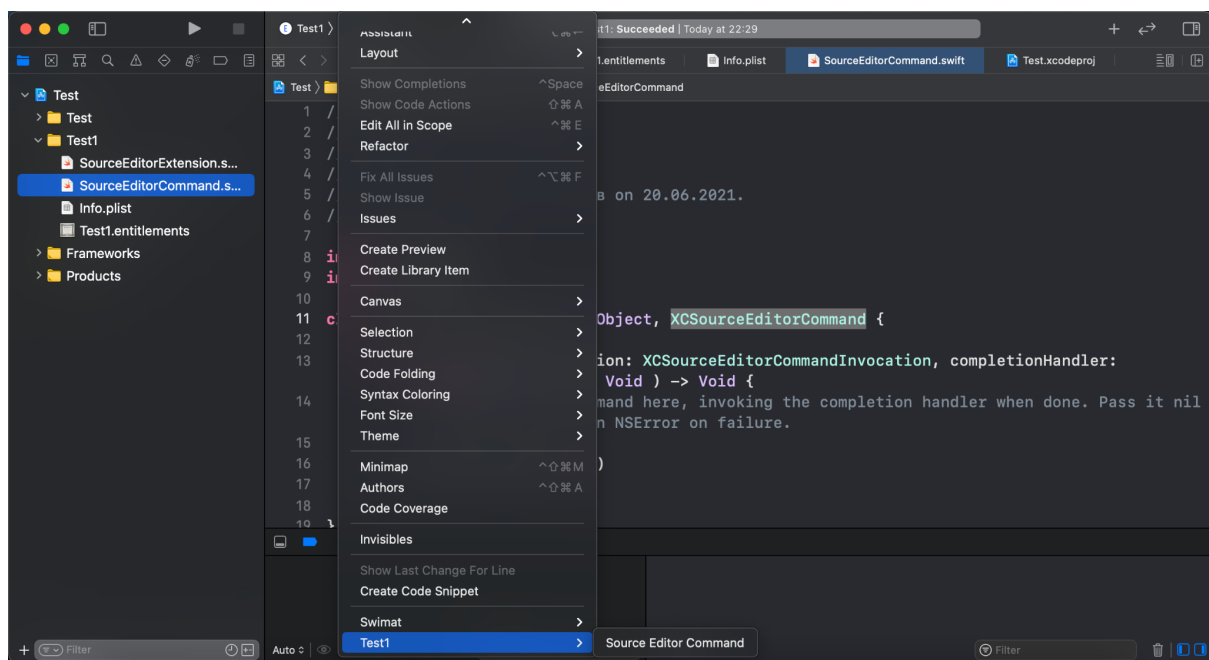
Перед запуском обязательно изменим XcodeKit на Embed & Sign



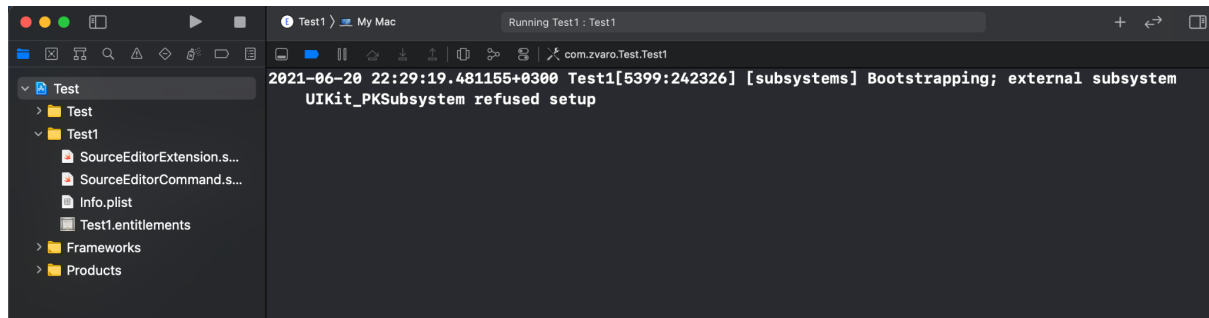
При запуске выберем цель запуска Xcode



Запустится изолированный инстанс xcode который можно отличить по серой иконке и в нём во вкладке Editor видим расширение и команды прописанные в Info.plist



В исходной вкладке можно мониторить дебаг консоль



Реализация

Наш основной метод это perform в классе наследнике XCSourceEditorCommand, параметр invocation содержит buffer в котором есть вся информация о нашем файле, курсоре и выделенном тексте, в completionHandler возвращаем ошибку или nil

```
class SourceEditorCommand: NSObject, XCSourceEditorCommand {  
  
    func perform(with invocation: XCSourceEditorCommandInvocation, completionHandler:  
        @escaping (Error?) -> Void ) -> Void {  
        // Implement your command here, invoking the completion handler when done. Pass it nil  
        // on success, and an NSError on failure.  
  
        completionHandler(nil)  
    }  
  
}
```

Реализуем наш метод

```
class SourceEditorCommand: NSObject, XCSourceEditorCommand {
    func perform(with invocation: XCSourceEditorCommandInvocation, completionHandler: @escaping (Error?) -> Void) -> Void {
        var updatedLineIndexes = [Int]()
        for lineIndex in 0 ..< invocation.buffer.lines.count {
            let line = invocation.buffer.lines[lineIndex] as! NSString
            do {
                let regex = try NSRegularExpression(pattern: "\\{.*\\}(\\.+\\.\\.)+in", options: .caseInsensitive)
                let range = NSRange(0 ..< line.length)
                let results = regex.matches(in: line as String, options: .reportProgress, range: range)
                - = results.map { result in
                    let cleanLine = line.remove(characters: ["(", ")"], in: result.range)
                    updatedLineIndexes.append(lineIndex)
                    invocation.buffer.lines[lineIndex] = cleanLine
                }
            } catch {
                completionHandler(error as NSError)
            }
        }

        if !updatedLineIndexes.isEmpty {
            let updatedSelections: [XCSourceTextRange] = updatedLineIndexes.map { lineIndex in
                let lineSelection = XCSourceTextRange()
                lineSelection.start = XCSourceTextPosition(line: lineIndex, column: 0)
                lineSelection.end = XCSourceTextPosition(line: lineIndex, column: 0)
                return lineSelection
            }
            invocation.buffer.selections.setArray(updatedSelections)
        }
        completionHandler(nil)
    }
}
```

Найдем в файле регехр и уберём () и обновим буффер. На выходе получим готовое расширение

Официальная документация Apple

https://developer.apple.com/documentation/xcodekit/creating_a_source_editor_extension