**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет прикладной математики и информатики

Кафедра технологий программирования

Жуковский Павел Сергеевич

Технологии Программирования

Отчет по лабораторной работе №6

«Разработка консольных приложений на Objective-C и Swift»

студента 2 курса 13 группы

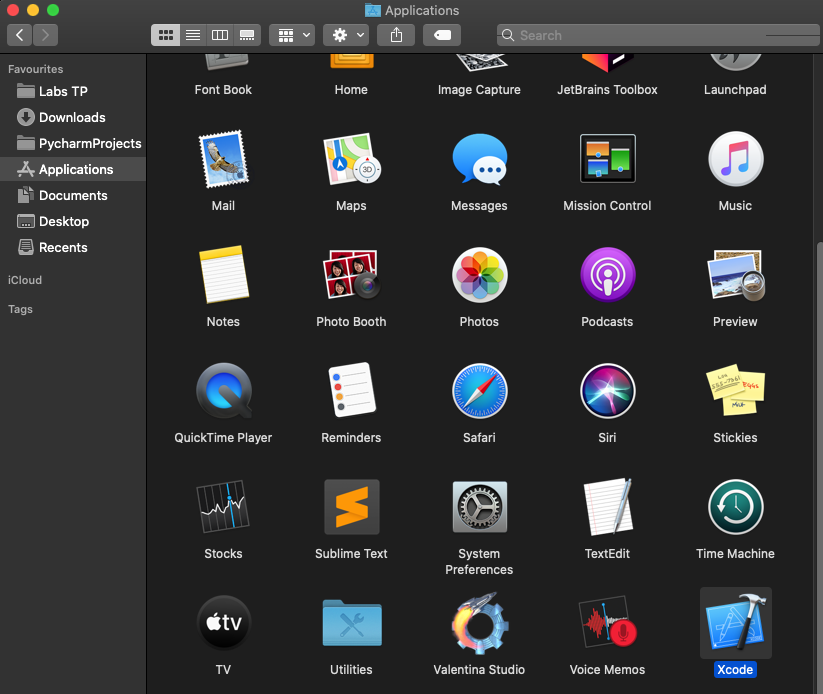
**Преподаватель**

**Давидовская Мария Ивановна**

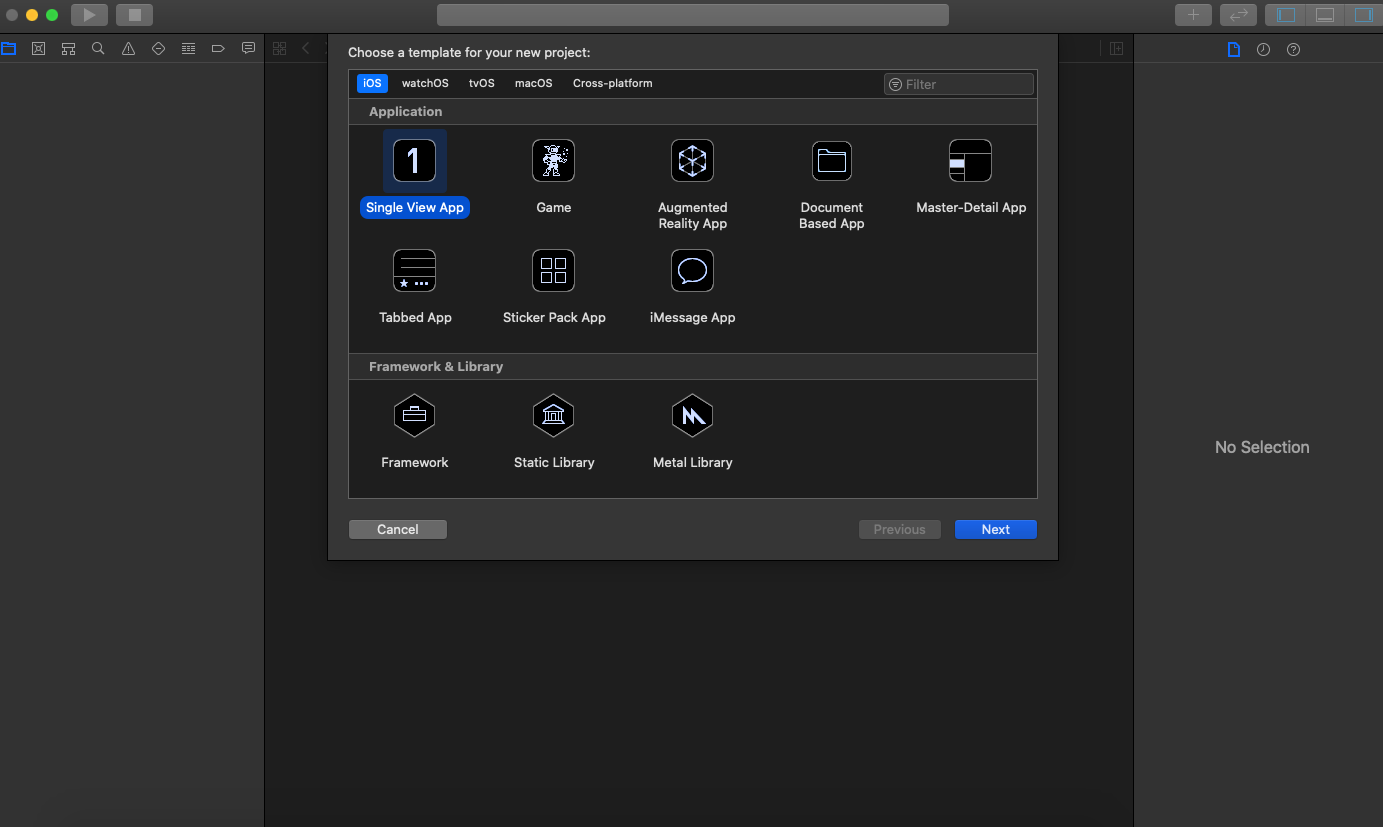
**Минск 2019**

**Задание №1**

Установил X-Code из AppStore:



Разобрался с созданием проектов в X-Code:



Контрольные вопросы:

1. Какие способы запуска симулятора iOS возможны?

Для начала, нужно убедится, что в всплывающем меню «Scheme» выбрано нужное устройство (например, iPhone 6). После этого нужно просто нажать кнопку запуска в верхнем левом углу панели X-Code. Также можно в меню Product выбрать и нажать кнопку Run для запуска. Для этого также можно нажать сочетание клавиш Command+R.

2. Как управлять симулятором?

Вообще, среда X-Code предоставляет множество функций и инструментов для работы с симулятором. Например, мы можем переключаться между различными устройствами (условно с iPhone 7 на iPhone 6 или даже на iPad), также мы можем менять версию iOS, поворачивать устройство и многое другое.

**Задание №2 (вариант 17)**

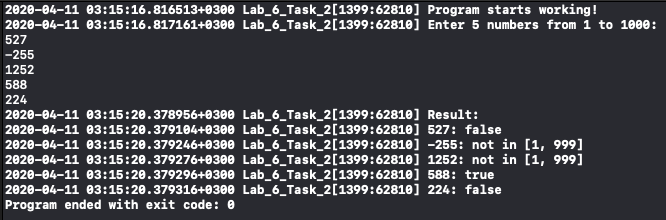
Для чисел от 1 до 1000, найти сотни, в которых есть внутренние повторение

(например, 122, 133, 144, 677 и т. д.).

Реализация на Objective-C:



Пример работы:



Протокол тестирования:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | 527  -255  1252  588  224 | 527: false  -255: not in [1, 999]  1252: not in [1, 999]  588: true  224: false | 527: false  -255: not in [1, 999]  1252: not in [1, 999]  588: true  224: false | Да |
| 2 | 911  244  588  944  255 | 911: true  244: true  588: true  944: true  255: true | 911: true  244: true  588: true  944: true  255: true | Да |
| 3 | -8247  909  112  757  556 | -8247: not in [1, 999]  909: false  112: false  757: false  556: false | -8247: not in [1, 999]  909: false  112: false  757: false  556: false | Да |

Контрольные вопросы:

1. Какие расширения файлов у следующих типов файлов?

• C language source file – \*.c

• C++ language source file – \*.cpp

• Header file – \*.h

• Objective-C source file – \*.m

• Objective-C++ source file – \*.mm

• Object (compiled) file – \*.obj

2. Способы вывода на экран в Objective-C.

1) С помощью СИ-шной функции printf(«string»);

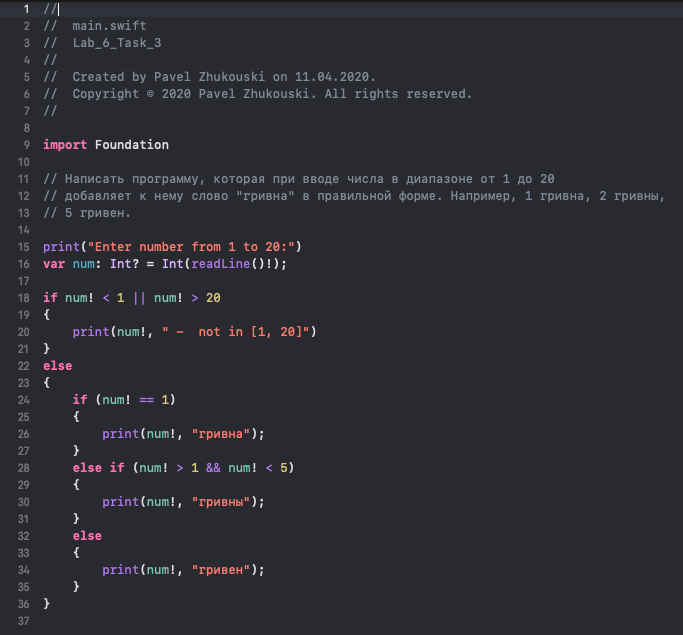
2) С помощью функции Objective-C под названием NSLog(@«string»), при чем здесь перед строкой обязателен символ ‘@’.

**Задание №3 (вариант 17)**

Написать программу, которая при вводе числа в диапазоне от 1 до 20

добавляет к нему слово "гривна" в правильной форме. Например, 1 гривна, 2 гривны, 5 гривен.

Реализация на Swift:



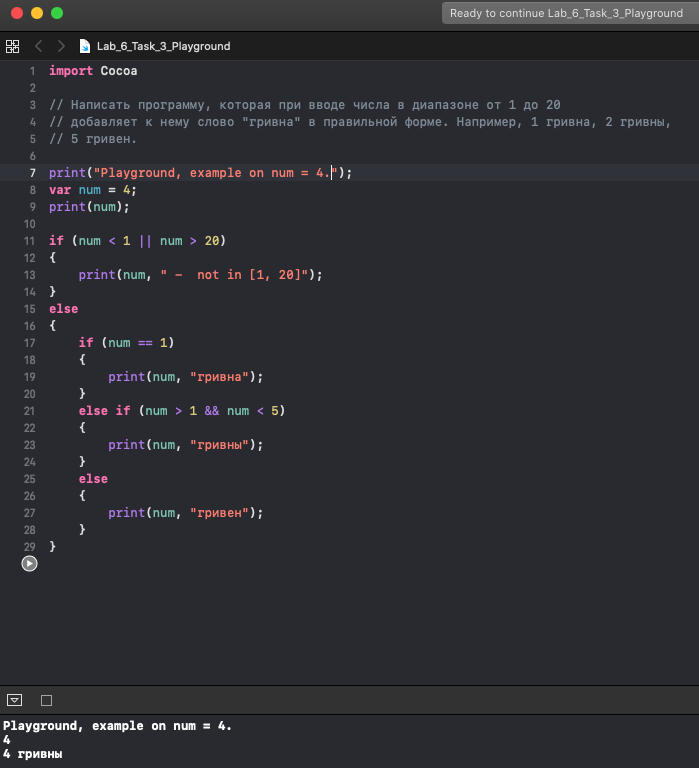
Пример работы:



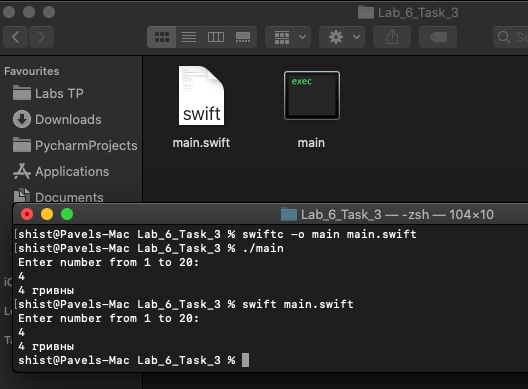
Протокол тестирования:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | 1 | 1 гривна | 1 гривна | Да |
| 2 | 4 | 4 гривны | 4 гривны | Да |
| 3 | 17 | 17 гривен | 17 гривен | Да |
| 4 | 25 | 25 – not in [1, 20] | 25 – not in [1, 20] | Да |

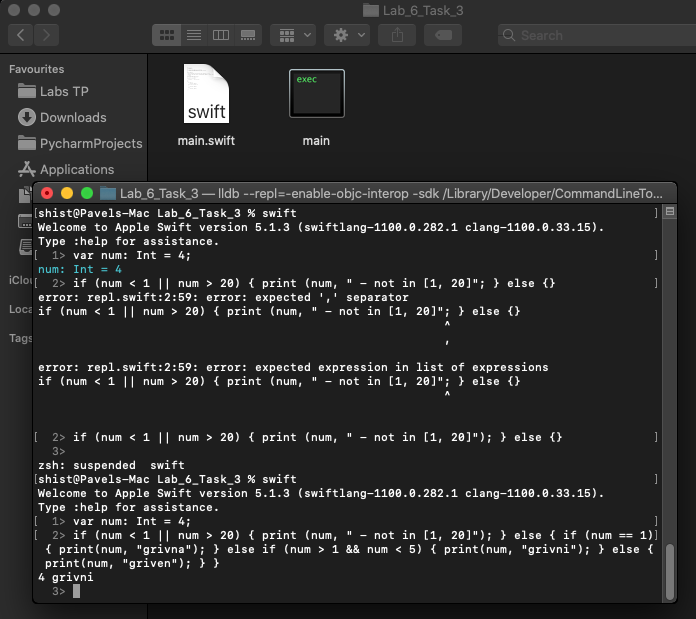
Запуск этого же приложения в Playground:



Компиляция приложения с помощью swiftc в консоли:



Разработка и запуск приложения в REPL:



Контрольные вопросы:

1. Что такое swift REPL и в каких случаях рекомендуется использовать?

Swift REPL (Read Eval Print Loop) – это возможность поэкспериментировать с приложением прямо в командной строке (для этого достаточно прописать команду swift). Данный функционал можно отнести к интерпретируемым языкам, что очень удобно, если нам нужно где-то поэкспериментировать с приложением и запустить тот или иной код немедленно.

2. Что такое playgraund? Какие задачи (проекты) можно создавать в

playground?

Playground (также известная как «песочница») – это еще один инструмент для экспериментирования с кодом. Она запускается в среде X-Code. Обычно в ней реализуют простые интерактивные примеры кода для приложений, например, на языке Swift.

3. Как компилировать приложение на языке swift в консоли bash?

Для компиляции приложения на языке swift в консоли bash достаточно прописать следующие команды в консоли:



Первая строчка отвечает за компиляцию приложения main.swift, а вторая за выполнение. Также, для дальнейшего выполнения достаточно прописывать:



**Задание №4**

Контрольные вопросы:

1. Как создать локальный git-репозиторий для проектов в Xcode?

Для того, чтобы создать локальный git-репозиторий для своего проекта в X-Code, достаточно при создании этого проекта подключить навигатор Source Control.

2. Как добавить внешний репозиторий в Xcode? Назвать два способа

подключения.

Для того, чтобы добавить в X-Code внешний репозиторий, можно: 1) зайти в навигатор Source Control, и в нём, нажав правой кнопкой мыши по имени проекта, выбрать пунктик «Create Remote on GitHub»; 2) в среде X-Code зайти в раздел Preferences 🡪 Accounts, после чего подключить свой аккаунт с GitGub.

3. Как создать ветку репозитория в Xcode?

Для того, чтобы создать ветку репозитория в X-Code, достаточно в навигаторе Source Control нажать на кнопочку в верхнем левом углу, где выбрать пунктик «Branch from «master»». Там можно ввести имя новой ветки и добавить ее, нажав «Create».

4. Как отменить commit в Xcode?

Для того, чтобы отменить commit в X-Code, требуется выбрать тот участок кода, который «закоммитили», после чего нажать на стрелочку рядом с номером его изменения. Среда X-Code выдаст меню, где нам следует выбрать пунктик «Don’t Commit».

5. Как слить ветки в Xcode?

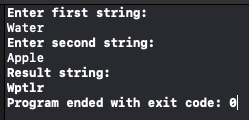
Если требуется слить ветки в X-Code, нужно нажать правой кнопкой мыши на ветку в навигаторе Source Control, после чего выбрать «Merge «development» into «master»», и в конце подтвердить это действие.

**Задание №5 (вариант 17)**

Реализация на Swift:



Пример работы:



Протокол тестирования:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| №  п/п | Входные данные | Ожидаемые выходные данные | Действительные выходные данные | Тест пройден |
| 1 | Apple  Pear | The amount of symbols in strings is different!!! | The amount of symbols in strings is different!!! | Да |
| 2 | Mouse  Keyboard | The amount of symbols in strings is different!!! | The amount of symbols in strings is different!!! | Да |
| 3 | Water  Apple | Wptlr | Wptlr | Да |
| 4 | Apple  Water | Aapee | Aapee | Да |

Контрольные вопросы:

1) Расскажите о режимах работы среды программирования X-Code.

Среда X-Code открывает нам огромный функционал с различными возможностями. В частности, он позволяет проработать общую схему приложения, а также его визуальную часть с помощью Interface Builder. Мы можем создавать код к приложению в редакторе, параллельно экспериментируя с отдельными его компонентами в песочнице playground. Также X-Code позволяет совершать отладку приложения, его тестирование, создание документации к нему, работу с GitHub и многое другое.

2) В чем отличия языка программирования Swift от C и С++.

Самое главное отличие, по моему мнению, – это то, что в языке Swift нам не обязательно указывать тип того или иного объекта, компилятор сам его определит. Также стоит отметить, что в Swift отсутствуют указатели, чего не скажешь о языках C/С++. Ну и напоследок можно добавить, что в Swift реализованы встроенные перечисления, которые могут сочетать в себе тот же функционал, что и классы со структурами в языках C/C++.

3) Какие способы отладки программ поддерживает среда X-Code?

В первую очень хотел бы отметить отладчик, которым мне уже приходилось пользоваться – это отладка в режиме Debug с помощью точек остановы. Для этого достаточно нажать напротив той строчки, где мы бы хотели остановиться и посмотреть, что происходит в программе, а потом запустить программу. Данный отладчик позволяет не только проходиться по программе пошагово, но и следить за тем, что хранится во всех переменных. Также, существует консольный отладчик LLDB, который позволяет совершать отладку в консоли.

4) Какие операторы языка swift Вы знаете?

В языке swift существуют следующие операторы:

Унарные: префиксные (например, -a или !b) и постфиксные (например, i++);

Бинарные: например, 2 + 3, 8 – 2, 8 \* 2;

Тернарные: например, a ? b : c (если a – правда, то возвращаем b, иначе – с);

5) Какие типы коллекций применяются в языке Swift и в чем их особенности?

Массив: Array

Тип Swift соединен с NSArray классом Foundation.

Хранит значения одного и того же типа в упорядоченном списке. Одно и то же значение может появляться в массиве несколько раз в разных позициях.

Набор: Set

Тип Swift соединен с NSSet классом Foundation .

Хранит различные значения одного и того же типа в сборе с не определяется упорядочением. Вы можете использовать набор вместо массива, когда порядок элементов не важен или когда вам нужно убедиться, что элемент появляется только один раз.

Словарь: Dictionary

Представляет собой хранилище, в котором каждый элемент имеет ключ и значение. Разные элементы не могут иметь одинаковые ключи. Все ключи в словаре должны быть уникальными. По ключу мы можем найти элемент в словаре изменить его или удалить.

6) Какие типы проектов X-Code Вы знаете?

* Empty Application – Совсем пустой шаблон, если нам нужно создать проект с нуля;
* Single View Application – Шаблон, в котором есть одно готовое окно;
* Tabbled Application – Шаблон проекта, в котором имеется несколько вкладок;
* Master-Detail Application – Шаблон приложения с развернутым меню;
* Utility Application – Шаблон, нужный для создания утилит гаджетов;
* SpriteKit Game – Шаблон для создания игр на основе SpriteKit;
* OpenGL Game – Шаблон для создания игр на основе OpenGL;
* Page-Based Application – Шаблон приложения, основанного на перелистывании страниц;