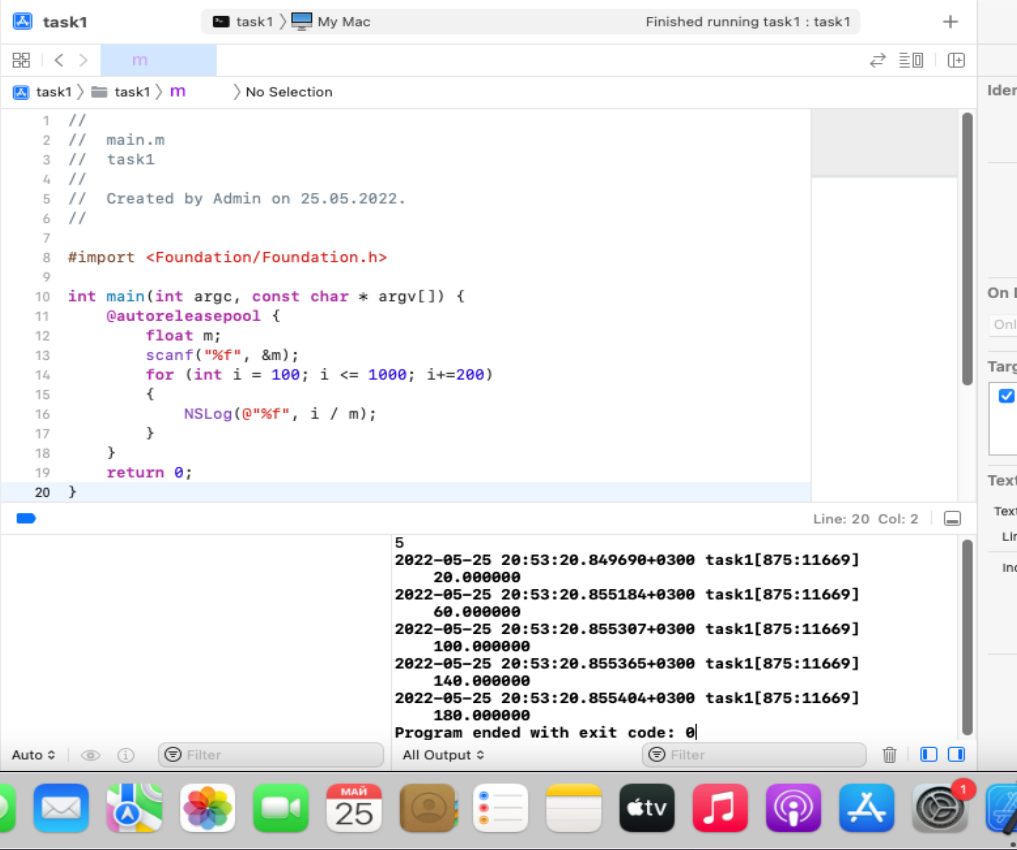
Задание 1

+

Задание 2

2) Реализовать приложение на Objective-C согласно варианту.



3) Опубликовать код приложения в git-репозиторий в каталоге task2 и в ветке feature-task2.

4) Предоставить протоколы тестов.

Тест Ожидаемый ответ Ответ

5 - 20 60 100 140 180

2 - 50 150 250 350 450

5) Ответить на вопросы.

ВОПРОСЫ

1. Какие расширения файлов у следуюших 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 .o

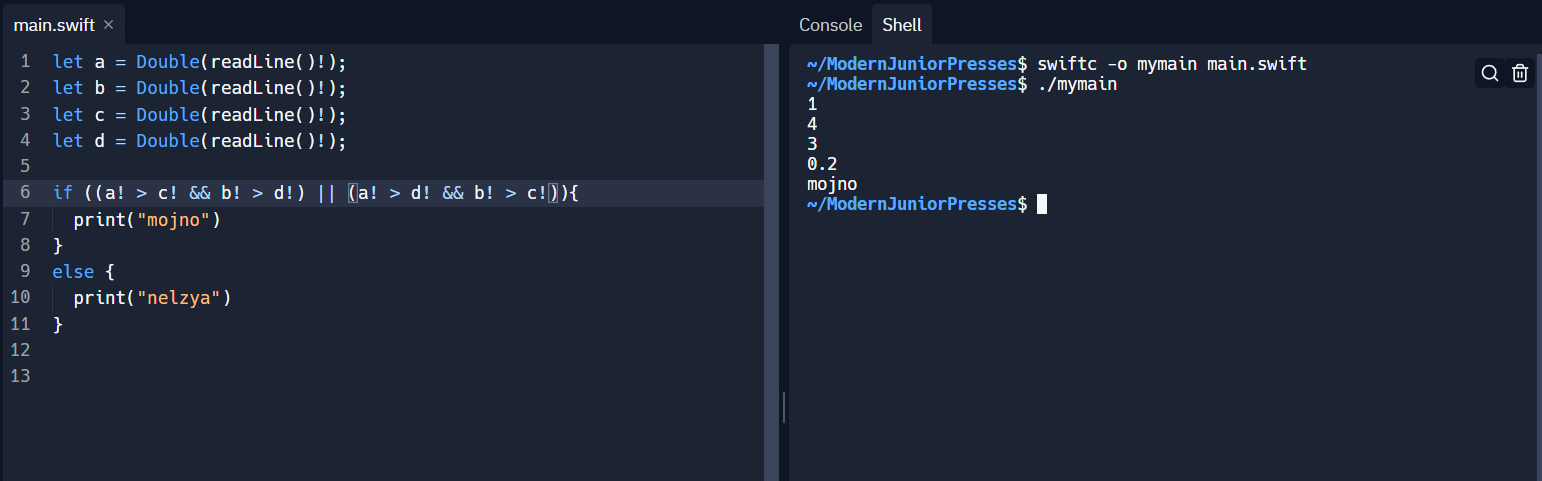
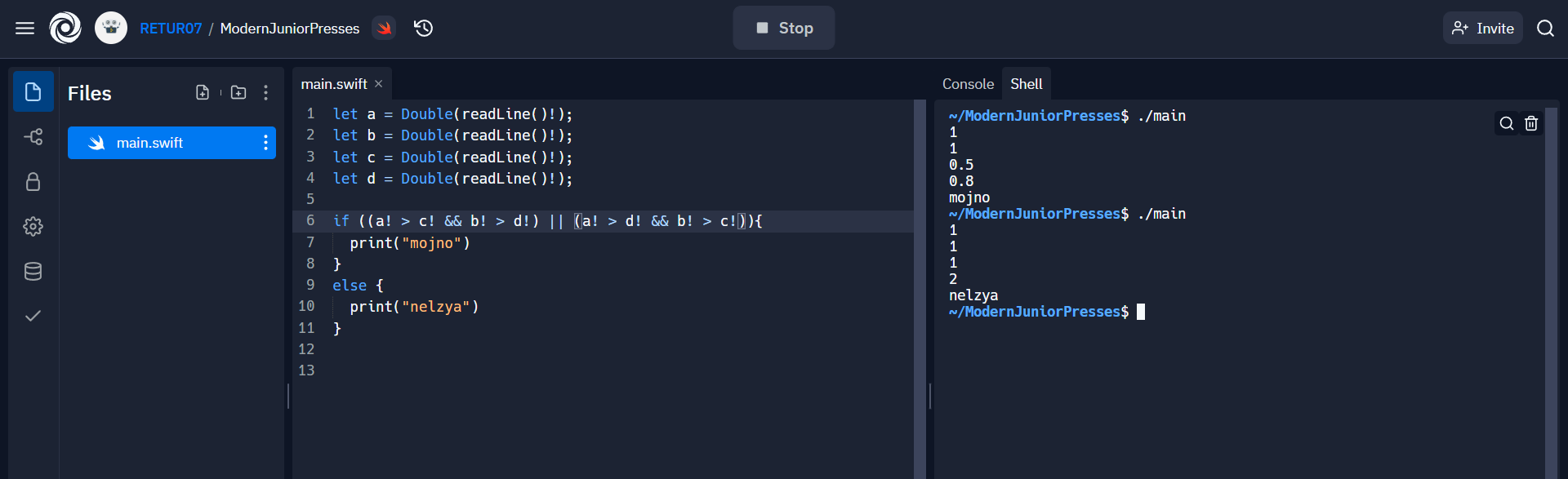
2. Способы вывода на экран в Objective-C

* NSLog
* printf

Задание 3

3. Продемонстрировать навыки работы с консолью Swift REPL, компиляцию

приложений с помощью swiftc



ВОПРОСЫ

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

Он позволяет нам сделать разбор написанного кода в режиме runtime. Удобно экспериментировать с небольшим по размеру кодом и выполнять debug.

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

Playground, или в переводе на русский «песочница», отлично подойдут для изучения Swift. Вы вводите строку кода и сразу видите результат. После того, как вы убедитесь, что написанный вами код дает нужный результат, вы можете просто взять и перенести его в свой проект.

Например, при помощи Playground вы можете решить следующие задачи: освоить синтаксис Swift поэкспериментировать с новыми API, понаблюдать за каждым шагом его выполнени.

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

% swiftc -o file file\_name.swift

% ./file\_name

4 задание

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

Когда запускаешь проект, необходимо выбрать “ создать репозиторий”

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

a) Зайти в source control , выбрать clone.

б) *Файл → Управление источником → Push...*

3. Как создать ветку репозитория в Xcode?

Выбрать Branch from master

4. Как отменить commit в Xcode?

Пишем git hist, получая список всех коммитов в ветке

Далее находим по комментарию нужный коммит и копируем 7 символов хэша этого коммита.

Далее git reset --hard имя\_коммита\_перед\_неверным

Таким образом мы сбросим ветку до последнего верного коммита.

5. Как слить ветки в Xcode?

Выбрать merge into Branch

Задание 5

ВОПРОСЫ

1) Расскажите о режимах работы среды программирования xCode.

**Interface Builder** – Это интерфейсный редактор, который позволяет проектировать и создавать полный пользовательский интерфейс мобильных приложений. Является составной частью IDE Xcode.

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

Отсутствуют указатели (то есть язык является более безопасным). Присутствуют именованные параметры, Swift имеет более богатую систему типов, чем C ++

3. Какие способы отладки программ поддерживает среда xCode?

Breakpoint, Breakpoint - это точка, которая позволяет остановить программу в некоторый момент времени и выполнить некоторые действия над запущенной программой.

LLDB - консольный отладчик

4. Какие операторы языка swift Вы знаете?

Унарные, бинарные, тернарные (условный оператор: выражение ? действие1 : действие2). Операторы сравнения, арифметические операторы, логические операторы.

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

Swift обеспечивает три основных типа коллекций - это Массивы, Множества и Словари для хранения коллекций значений.

6. Какие типы проектов xCode Вы знаете?

Single View Application – шаблон с одним готовым окном Empty Application – пустой шаблон для создания приложения с нуля Tabbled Application – для проекта с несколькими вкладками OpenGL Game – для игр на основе OpenGL.

**<https://github.com/fpmi-tp2022/lab6/>**

|  |
| --- |
| #import <Foundation/Foundation.h> |
| int main(int argc, const char \* argv[]) { |
| @autoreleasepool { |
| float m; |
| scanf("%f", &m); |
| for (int i = 100; i <= 1000; i+=200) |
| { |
| NSLog(@"%f", i / m); |
| } |
| } |
| return 0; |
| } |

|  |
| --- |
| let a = Double(readLine()!);  let b = Double(readLine()!); |
| let c = Double(readLine()!); |
| let d = Double(readLine()!); |
|  |
| if ((a! > c! && b! > d!) || (a! > d! && b! > c!)){ |
| print("mojno") |
| } |
| else { |
| print("nelzya") |
| } |

import Foundation

import Glibc

var flights : [String : Int] // init empty

var airports : [String : String] = ["a" : "b", "c" : "d"]

print(airports.count) // count

print("a")

print(airports["a"]!) // print key and value

for (name, city) in airports{

print("\(name) : \(city)")

} // print all elements

airports["e"] = "u" // add

airports["a"] = "t" // update

airports.removeValue(forKey : "c")

print()

for (name, city) in airports{

print("\(name) : \(city)")

}

for (key, \_) in airports{

airports.removeValue(forKey : key)

}

for (name, city) in airports{

print("\(name) : \(city)")

}

print(airports.count) // count