**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №4

по дисциплине: **«**Разработка приложений для *IPhone* и *IPad***»**

на тему: **Обобщения**

Выполнил: студент гр. ИТП-41

Солодков М.А

Приняла: Семенченя Т.С.

Гомель 2020

**Цель**: изучить основы языка программирования *Swift* (наследование и полиморфизм)

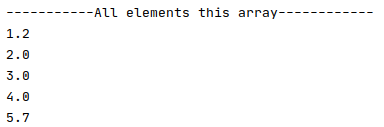
**Ход работы**

**Вариант 13**

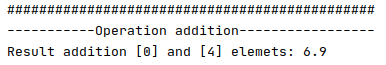
**Задание**

10. Создать объект класса Одномерный массив, используя класс Массив.

Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить)



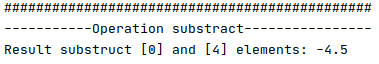
**Рисунок 1** – Результат создания и добавления элементов в массив



**Рисунок 2** – Результат операции сложения двух чисел



**Рисунок 3** – Результат операции умножения двух чисел



**Рисунок 4** – Результат операции вычитания двух чисел

**Вывод:** в ходе выполнения лабораторной работы №3 были изучены основы программирования на языке *Swift*. А так же ознакомлен c обобщениями и псевдонимами типов в языке программирования *Swift.*

**Листинг программы**

**Array.swift**

class Array<T>{

var array: [T];

init(){

self.array = [T]();

}

func operationAddElement(element: T){

self.array.append(element);

}

func printConsole(){

self.array.forEach({print($0)});

}

func operationAddition<T:Numeric>(firstElement: T, secondElement: T) -> T{

fatalError();

}

func operationSubstract<T:Numeric>(firstElement: T, secondElement: T) -> T{

fatalError();

}

func operationMultiply<T:Numeric>(firstElement: T, secondElement: T) -> T{

fatalError();

}

}

**OneDimensionalArray.swift**

class OneDimensionalArray<T>: Array<T>{

override init(){

super.init();

}

override func operationAddElement(element: T) {

super.operationAddElement(element: element)

}

override func operationAddition<T: Numeric>(firstElement: T, secondElement: T) -> T{

return firstElement + secondElement;

}

override func operationMultiply<T: Numeric>(firstElement: T, secondElement: T) -> T{

return firstElement \* secondElement;

}

override func operationSubstract<T: Numeric>(firstElement: T, secondElement: T) -> T{

return firstElement - secondElement;

}

}

**Main.swift**

var array = OneDimensionalArray<Double>()

array.operationAddElement(element: 1.2);

array.operationAddElement(element: 2);

array.operationAddElement(element: 3);

array.operationAddElement(element: 4);

array.operationAddElement(element: 5.7);

print("-----------All elements this array------------")

print(array.printConsole())

print("##############################################")

print("-----------Operation addition-----------------")

print("Result addition [0] and [4] elemets: \(array.operationAddition(firstElement: array.array[0], secondElement: array.array[4]))");

print("##############################################")

print("-----------Operation multiply-----------------")

print("Result multiply [0] and [4] elements: \(array.operationMultiply(firstElement: array.array[0], secondElement: array.array[4]))");

print("##############################################")

print("-----------Operation substract----------------")

print("Result substruct [0] and [4] elements: \(array.operationSubstract(firstElement: array.array[0], secondElement: array.array[4]))");