**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №3

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

на тему: **Наследование и полиморфизм**

Выполнил: студент гр. ИТП-41

Солодков М.А

Приняла: Семенченя Т.С.

Гомель 2020

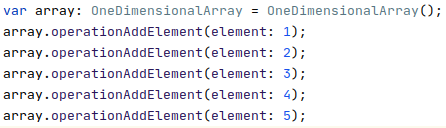
**Цель**: изучить основы языка программирования *Swift* (наследование и полиморфизм)

**Ход работы**

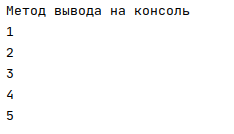
**Вариант 13**

**Задание**

13. Создать объект класса Одномерный массив, используя класс Массив. Методы: создать, вывести на консоль, выполнить операции (сложить, вычесть, перемножить)



**Рисунок 1** – Создание одномерного массива и добавление элементов



**Рисунок 2** – Результат вывода в консоль добавленных элементов



**Рисунок 3** – Результат выполнения операции сложения с 1 и 2 элементами массива



**Рисунок 4** – Результат выполнения операции вычитания со 2 и 3 элементами массива



**Рисунок 5 –** Результат выполнения операции умножения с 3 и 5 элементами массива

**Вывод:** в ходе выполнения лабораторной работы №3 были изучены основы программирования на языке *Swift*. А так же ознакомлен c наследованием классов в языке программирования *Swift*.

**Листинг программы**

**Array.swift**

class Array{

var array: [Int]

init(){

self.array = [Int]();

}

func operationAddElement(element: Int){

self.array.append(element);

}

func operationPrintConsole(){

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

}

func operationAddition(firstElement: Int, secondElement: Int) -> Int{

return 0;

}

func operationSubstract(firstElement: Int, secondElement: Int) -> Int{

return 0;

}

func operationMultiply(firstElement: Int, secondElement: Int) -> Int{

return 0;

}

}

**OneDimensionalArray.swift**

enum Operation{

case addition(firstElement: Int, secondElement: Int)

case substract(firstElement: Int, secondElement: Int)

case multiply(firstElement: Int, secondElement: Int)

func operation() -> Int{

switch self{

case .addition(let firstElement, let secondElement):

return firstElement + secondElement;

case .substract(let firstElement, let secondElement):

return firstElement - secondElement;

case .multiply(let firstElement, let secondElement):

return firstElement \* secondElement;

}

}

}

class OneDimensionalArray : Array{

override init(){

super.init();

}

override func operationAddElement(element: Int) {

super.array.append(element);

}

override func operationAddition(firstElement: Int, secondElement: Int) -> Int{

Operation.addition(firstElement: firstElement, secondElement: secondElement).operation();

}

override func operationSubstract(firstElement: Int, secondElement: Int) -> Int{

Operation.substract(firstElement: firstElement, secondElement: secondElement).operation();

}

override func operationMultiply(firstElement: Int, secondElement: Int) -> Int {

Operation.multiply(firstElement: firstElement, secondElement: secondElement).operation();

}

func description() {

super.operationPrintConsole()

}

}

**Main.swift**

var array: OneDimensionalArray = OneDimensionalArray();

array.operationAddElement(element: 1);

array.operationAddElement(element: 2);

array.operationAddElement(element: 3);

array.operationAddElement(element: 4);

array.operationAddElement(element: 5);

print("Метод вывода на консоль")

array.description();

print("-----------------------")

print("Результат операции сложения: \(array.operationAddition(firstElement: array.array[0], secondElement: array.array[1]))");

print("-----------------------")

print("Результат операции вычитания: \(array.operationSubstract(firstElement: array.array[1], secondElement: array.array[2]))");

print("-----------------------")

print("Результат операции умножения: \(array.operationMultiply(firstElement: array.array[2], secondElement: array.array[4]))");

print("-----------------------")