Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Современные языки программирования

Отчёт

По лабораторной работе №3

**Использование языка программирования Swift: структуры и классы, методы**

Вариант 5

Выполнила:

студент группы 310101

Блажеевич В. А.

Проверил:

Усенко Ф. В.

Минск 2024

**Цель:** Выполнить разработку приложения с использованием языка программирования Swift: структуры и классы, методы.

Напишите программу с классом Math. Создайте два атрибута – a и b. Напишите методы addition – сложение, multiplication – умножение, division – деление, subtraction – вычитание. При передаче в методы параметров a и b с ними нужно производить соответствующие действия и печатать ответ.

Результат выполнения работы программы представлен на рисунке 1.

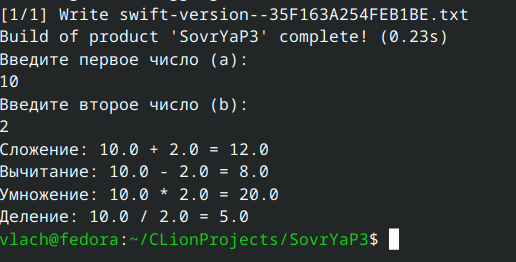


Рисунок 1 – Результат выполнения работы программы 1

**Вывод:** В ходе лабораторной работы разработано приложение на языке программирования Swift, выполнены задания на создание класса *Math* для реализации методов в нём. Осуществлена валидация вводимых значений.

**Ответы на вопросы**

3. Классы и структуры являются универсальными и гибкими конструкциями, которые станут строительными блоками для кода вашей программы. Для добавления функциональности в классах и структурах можно объявить свойства и методы, применив тот же синтаксис, как и при объявлении констант, переменных и функций.

В отличие от других языков программирования, Swift не требует создавать отдельные файлы для интерфейсов и реализаций пользовательских классов и структур. В Swift, вы объявляете структуру или класс в одном файле, и внешний интерфейс автоматически становится доступным для использования в другом коде.

Экземпляр *класса* традиционно называют *объектом*. Тем не менее, классы и структуры в Swift гораздо ближе по функциональности, чем в других языках, и многое в этой главе описывает функциональность, которую можно применить к экземплярам *и* класса, *и* структуры. В связи с этим, употребляется более общий термин – *экземпляр*.

5. *Тип значения* – это тип, значение которого копируется, когда оно присваивается константе или переменной, или, когда передается функции.

Все структуры и перечисления – типы значений в Swift. Это значит, что любой экземпляр структуры и перечисления, который вы создаете, и любые типы значений, которые они имеют в качестве свойств, всегда копируются, когда он передается по вашему коду.

В отличие от типа значений, *ссылочный тип не копируется*, когда его присваивают переменной или константе, или, когда его передают функции. Вместо копирования используется ссылка на существующий экземпляр.