**Отчет по самостоятельной работе №**3

**по дисциплине МДК 01.02 “Технология разработки программного обеспечения”.**

Выполнил: студент

группы 319

Меркушин Евгений Романович

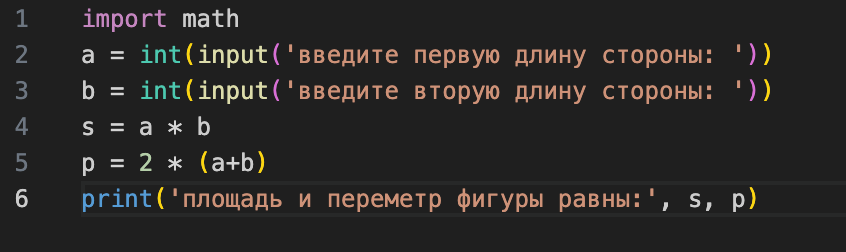
Дата 19.11.2024

**Цель работы**

Цель работы – изучить работу основных операторов языка Python, включая арифметические, логические и побитовые операции.

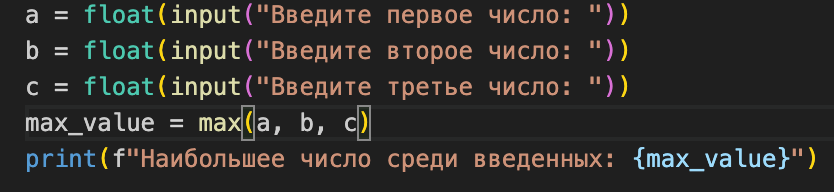
**Основная структура задания**

**Задание 1**



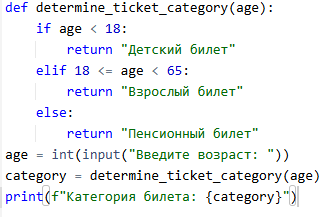
данная программа находит периметр и площадь фигуры

**Задание 2**



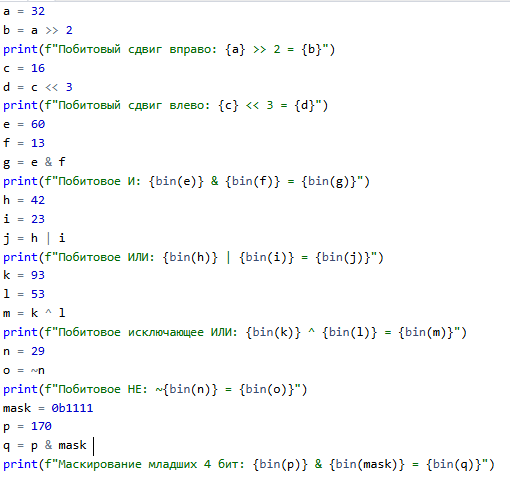
данная программа находит наибольшее число среди трех значений, вводимых пользователем

**Задание 3**



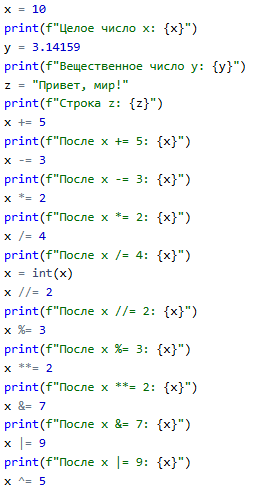
Данная программа осуществляет проверку возраста

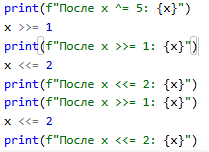
**Задание 4**



Эта программа использует побитовые операции для сдвига и маскирования битов

**Задание 5**





Эта программа демонстрирует работу операторов с различными типами данных

# Выводы работы

1. **Типы данных имеют значение**. В Python типы данных строго контролируются, и некоторые операции могут применяться только к определенным типам данных. В частности, побитовые операции (&, |, ^, >>, <<) применимы только к целым числам, но не к числам с плавающей точкой.
2. **Преобразование типов данных важно**. Если нужно использовать переменную в контексте, где ожидается другой тип данных, ее следует явно преобразовать. В нашем случае мы использовали int() для преобразования числа с плавающей запятой в целое число.
3. **Ошибки могут возникать незаметно**. Хотя ошибка возникла на строке 44, реальная причина была в предыдущей операции, которая привела к изменению типа переменной x. Важно внимательно следить за тем, какие типы данных используются в разных частях программы.
4. **Проверяйте промежуточные результаты**. Если возникают ошибки, полезно проверять промежуточные значения переменных, чтобы понять, где именно происходит проблема. В данном случае, если бы вы проверили значение x после операции x /= 4, то увидели бы, что оно стало числом с плавающей точкой, что и привело к проблеме.