Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития Кафедра инфокоммуникаций

# «Работа с функциями в языке Python»

**ОТЧЕТ**

# по лабораторной работе №11 дисциплины

**«Основы программной инженерии»**

|  |
| --- |
| Выполнил:  Сотников Андрей Александрович  2 курс, группа ПИЖ-б-о-21-1,  09.03.04 «Программная инженерия», направленность (профиль) «Разработка и сопровождение программного обеспечения», очная форма обучения  (подпись) |
| Проверил:  (подпись) |

Отчет защищен с оценкой Дата защиты

Ставрополь, 2022 г.

# Проработка примера из лабораторной работы:

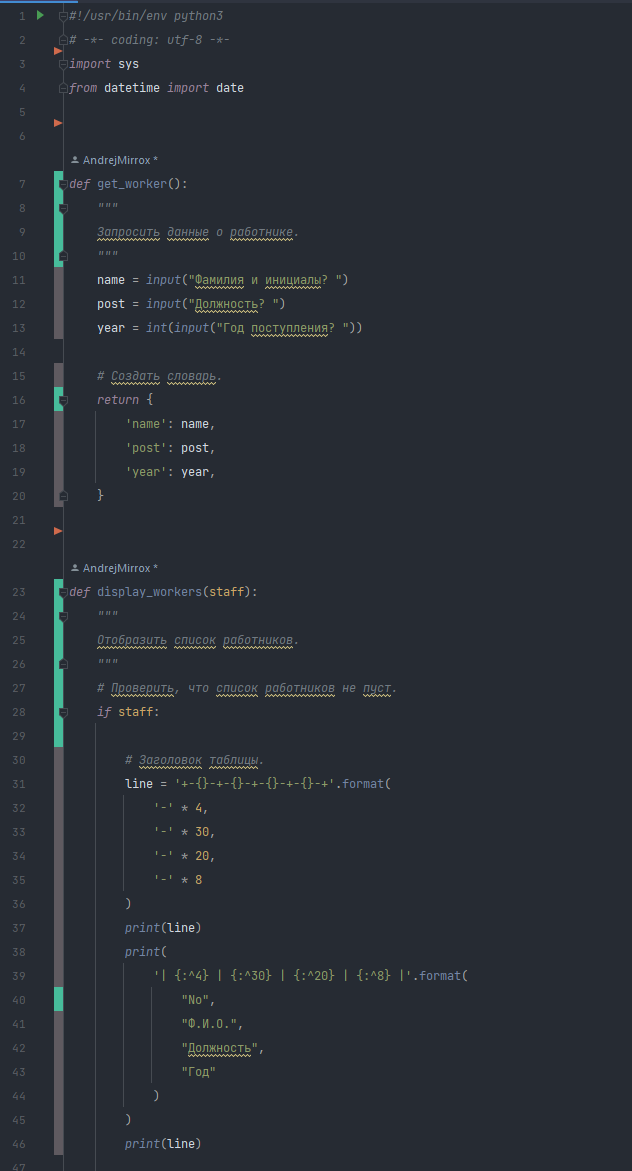
****

Рисунок 1 – Код примера

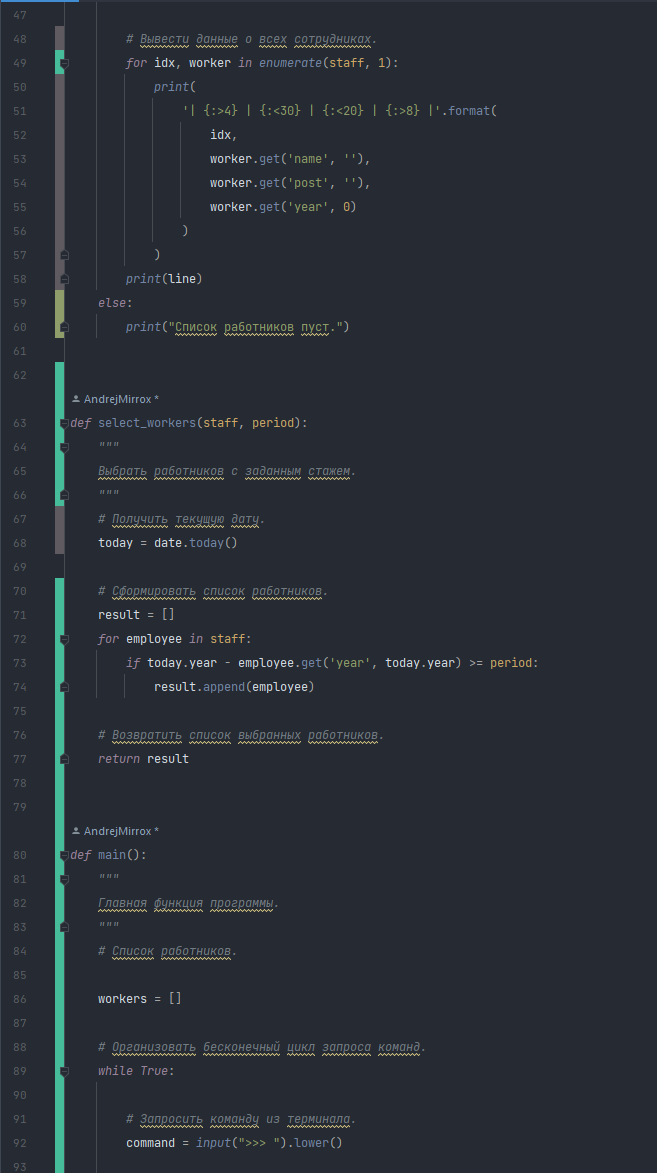


Рисунок 2 – Код примера



Рисунок 3 – Код примера

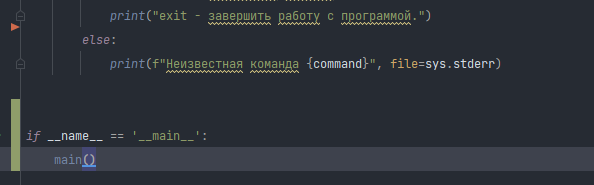


Рисунок 4 – Код примера

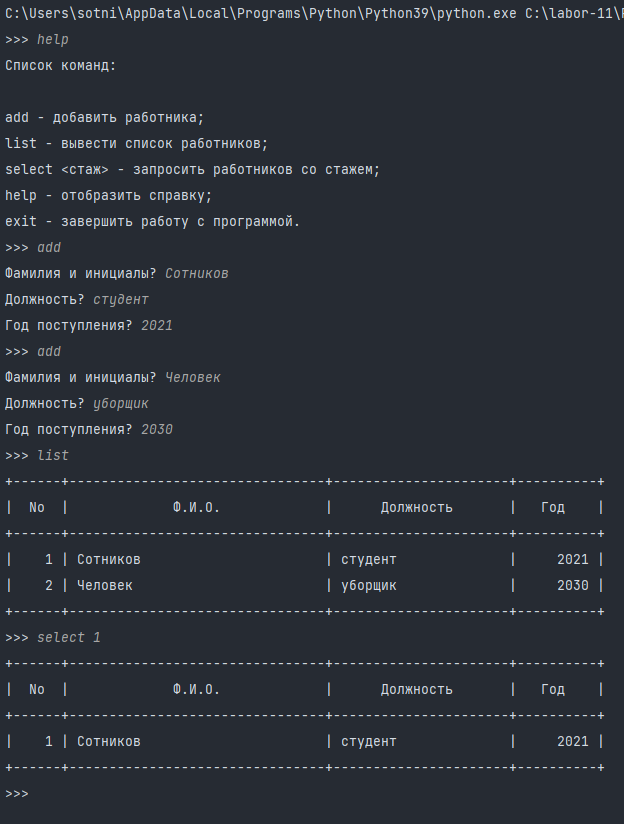


Рисунок 5 – Результат работы примера

**Задание №1**: решить следующую задачу: основная ветка программы, не считая заголовков функций, состоит из двух строки кода. Это вызов функции test() и инструкции if name == ' main ' . В ней запрашивается на ввод целое число. Если оно положительное, то вызывается функция positive(), тело которой содержит команду вывода на экран слова "Положительное". Если число отрицательное, то вызывается функция negative(), ее тело содержит выражение вывода на экран слова "Отрицательное". Понятно, что вызов test() должен следовать после определения функций. Однако имеет ли значение порядок определения самих функций? То есть должны ли определения positive() и negative() предшествовать test() или могут следовать после него? Проверьте вашу гипотезу, поменяв объявления функций местами. Попробуйте объяснить результат.

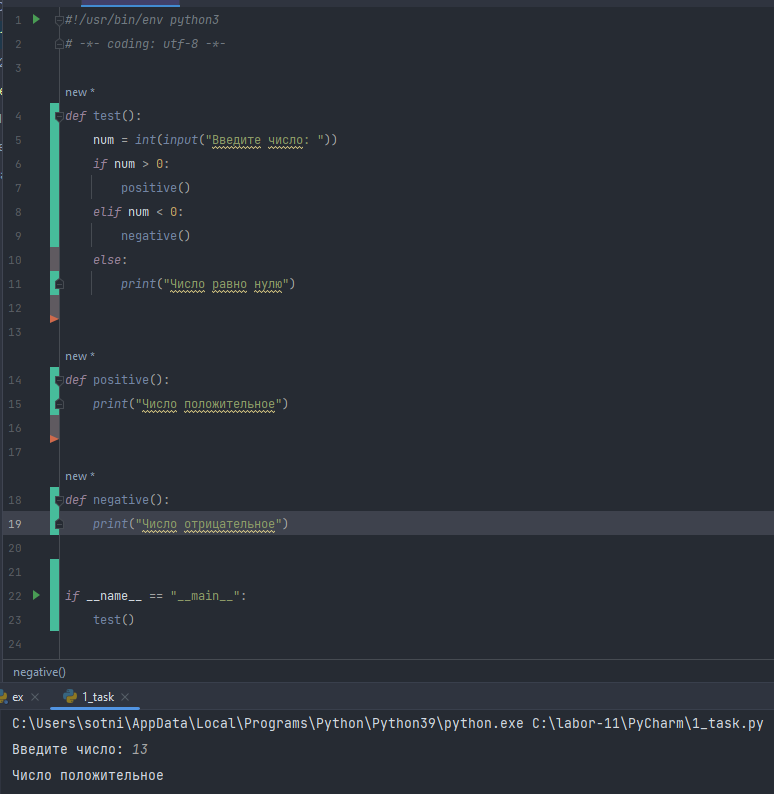


Рисунок 6 – Код и результат работы программы задания №1 с вызовом функций positive() и negative() после test()

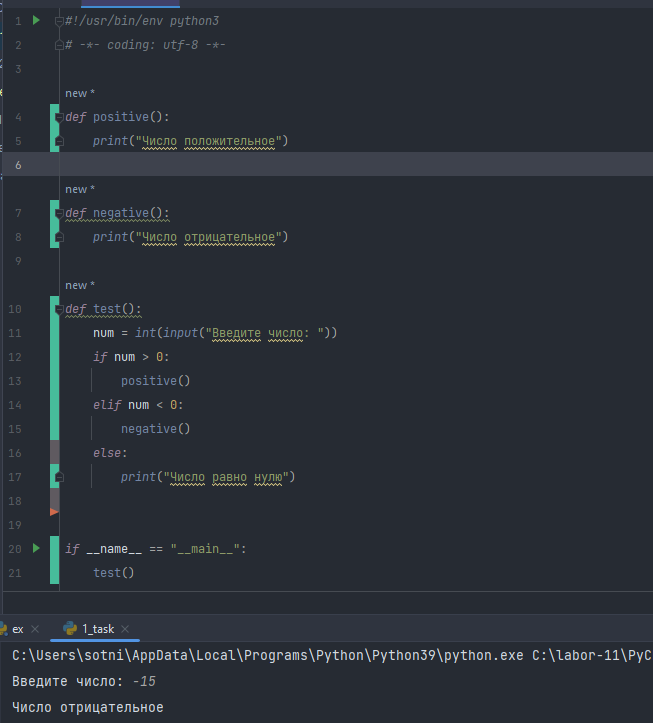


Рисунок 7 – Код и результат работы программы задания №1 с вызовом функций positive() и negative() перед test()

**Задание №2**: Решите следующую задачу: в основной ветке программы вызывается функция cylinder(), которая вычисляет площадь цилиндра. В теле cylinder() определена функция circle(), вычисляющая площадь круга по формуле . В теле cylinder() у пользователя спрашивается, хочет ли он получить только площадь боковой поверхности цилиндра, которая вычисляется по формуле , или полную площадь цилиндра. В последнем случае к площади боковой поверхности цилиндра должен добавляться удвоенный результат вычислений функции circle().

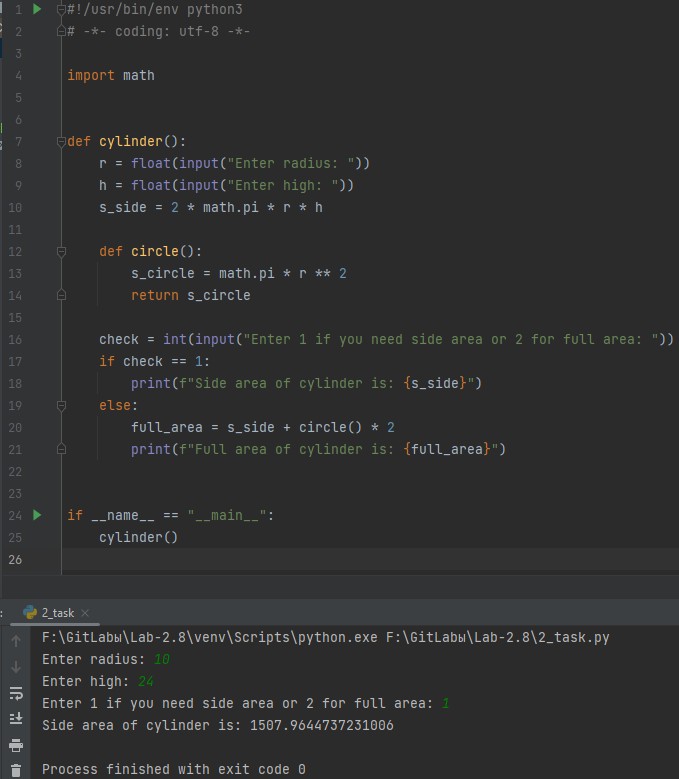


Рисунок 8 – Код и результат работы программы задания №2

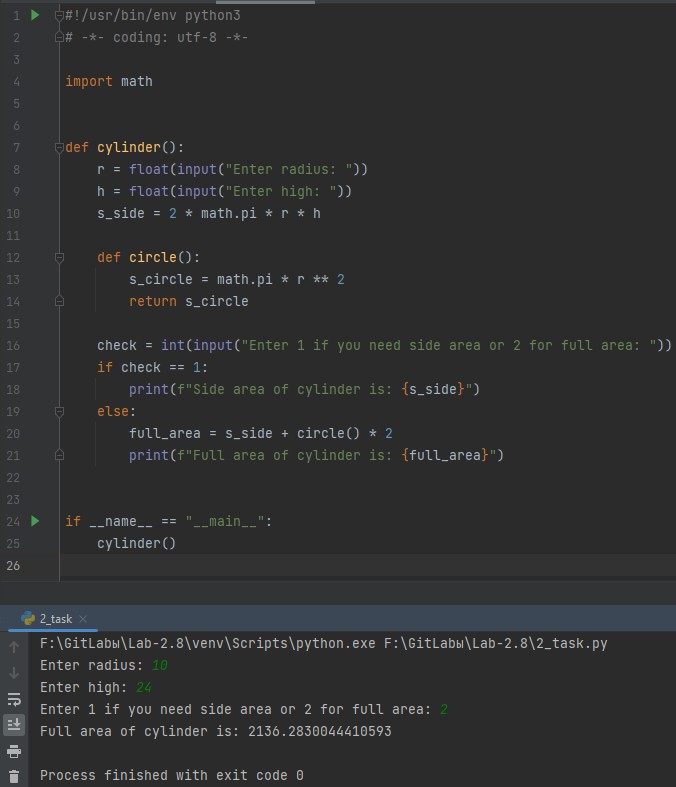


Рисунок 9 – Код и результат работы программы задания №2

**Задание №3**: решите следующую задачу: напишите функцию, которая считывает с клавиатуры числа и перемножает их до тех пор, пока не будет введен 0. Функция должна возвращать полученное произведение. Вызовите функцию и выведите на экран результат ее работы.

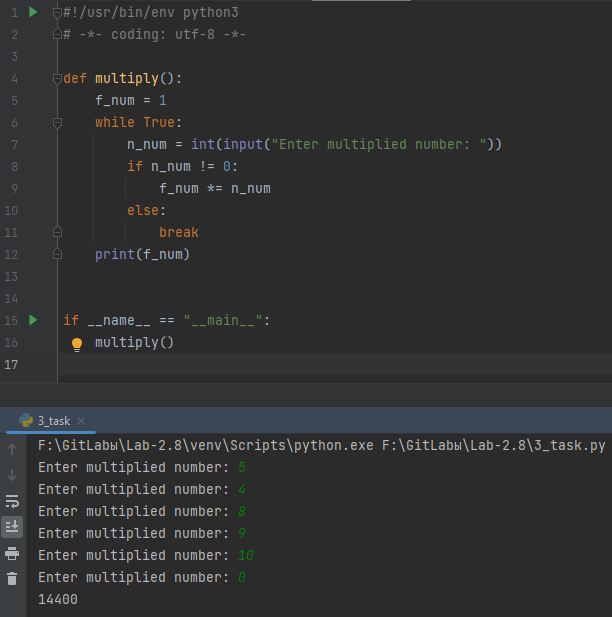


Рисунок 10 – Код и результат работы программы задания №3

**Задание №4**: решите следующую задачу: напишите программу, в которой определены следующие четыре функции:

1. Функция get\_input() не имеет параметров, запрашивает ввод с клавиатуры и возвращает в основную программу полученную строку.
2. Функция test\_input() имеет один параметр. В теле она проверяет, можно ли переданное ей значение преобразовать к целому числу. Если можно, возвращает логическое True. Если нельзя – False.
3. Функция str\_to\_int() имеет один параметр. В теле преобразовывает переданное значение к целочисленному типу. Возвращает полученное число.
4. Функция print\_int() имеет один параметр. Она выводит переданное значение на экран и ничего не возвращает.

В основной ветке программы вызовите первую функцию. То, что она вернула, передайте во вторую функцию. Если вторая функция вернула True, то те же данные (из первой функции) передайте в третью функцию, а возвращенное третьей функцией значение – в четвертую.

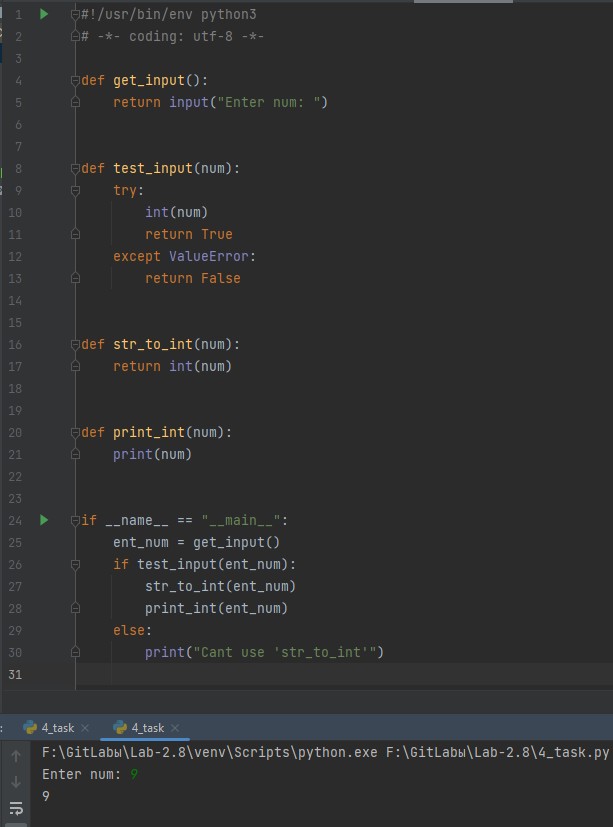


Рисунок 11 – Код и результат работы программы задания №4

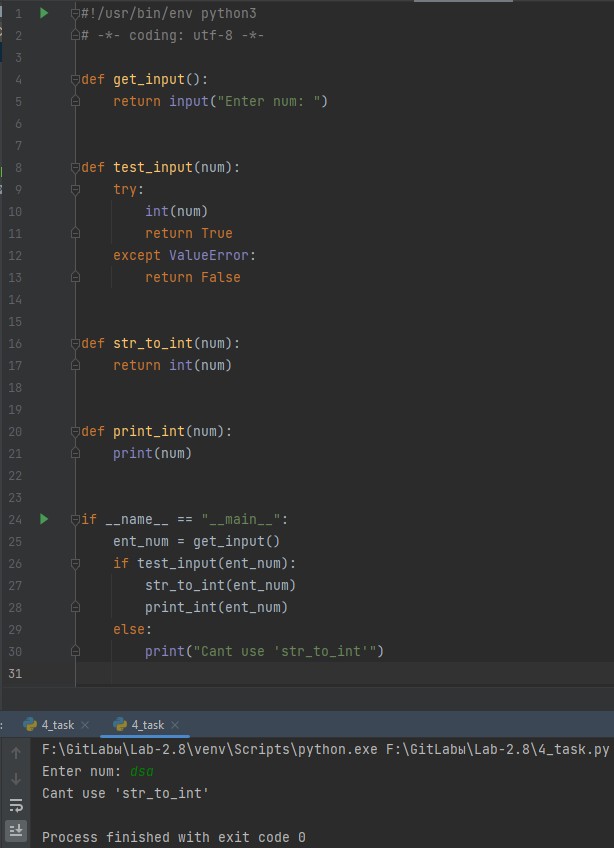


Рисунок 12 – Код и результат работы программы задания №4

**Индивидуальное задание**: решить индивидуальное задание лабораторной работы 2.6, оформив каждую команду в виде отдельной функции.

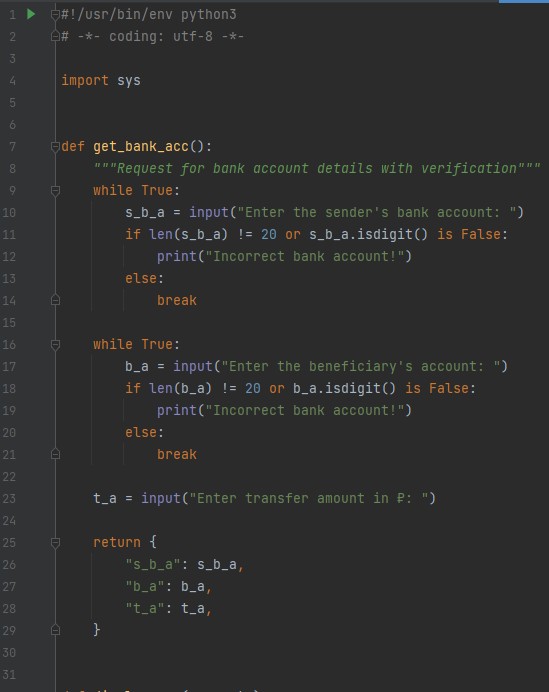


Рисунок 13 – Код программы для индивидуального задания

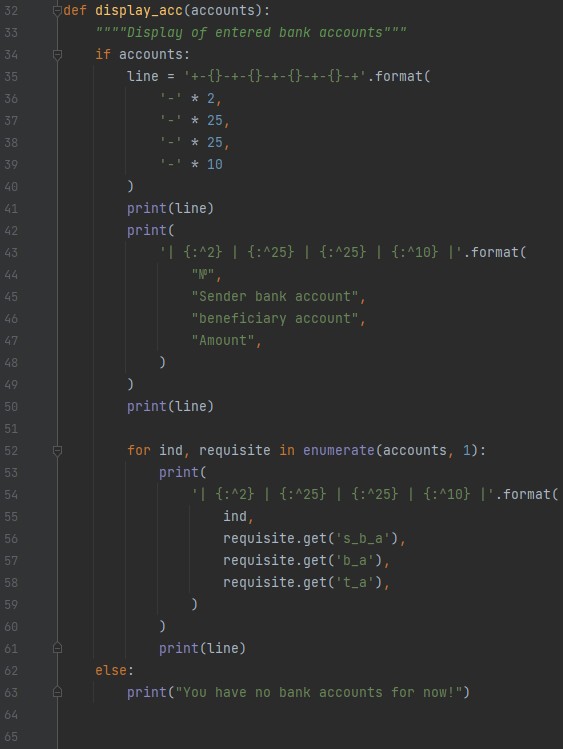


Рисунок 14 – Код программы для индивидуального задания



Рисунок 15 – Код программы для индивидуального задания

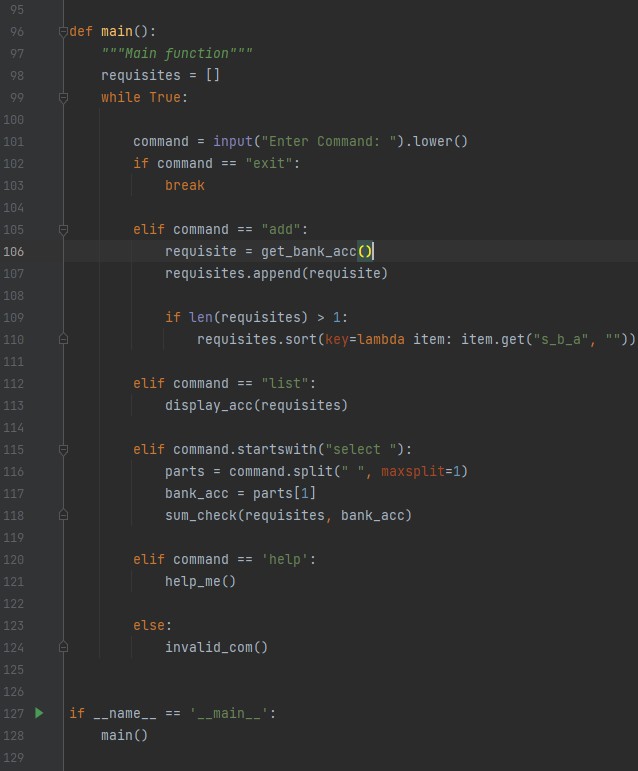


Рисунок 16 – Код программы для индивидуального задания

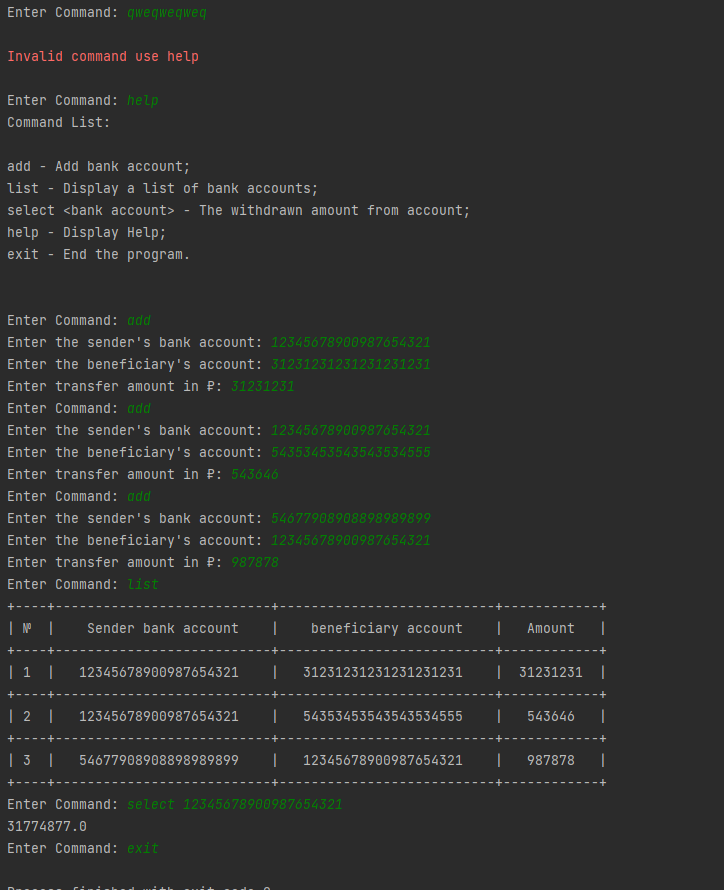


Рисунок 17 – Результат работы программы для индивидуального задания

# Контрольные вопросы

1. Каково назначение функций в языке программирования Python?

Функция в программировании представляет собой обособленный участок кода, который можно вызывать, обратившись к нему по имени, которым он был назван. При вызове происходит выполнение команд тела функции.

1. Каково назначение операторов def и return ?

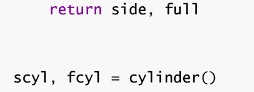
В языке программирования Python функции определяются с помощью оператора def

Функции могут передавать какие-либо данные из своих тел в основную ветку программы. Говорят, что функция возвращает значение. В большинстве языков программирования, в том числе Python, выход из функции и передача данных в то место, откуда она была вызвана, выполняется оператором return.

1. Каково назначение локальных и глобальных переменных при написании функций в Python?

В программировании особое внимание уделяется концепции о локальных и глобальных переменных, а также связанное с ними представление об областях видимости. Соответственно, локальные переменные видны только в локальной области видимости, которой может выступать отдельно взятая функция. Глобальные переменные видны во всей программе. "Видны" – значит, известны, доступны. К ним можно обратиться по имени и получить связанное с ними значение.

1. Как вернуть несколько значений из функции Python?



1. Какие существуют способы передачи значений в функцию?

Через параметры, и через ввод, запрашиваемый самой функцией

1. Как задать значение аргументов функции по умолчанию?



1. Каково назначение lambda-выражений в языке Python?

Python поддерживает интересный синтаксис, позволяющий определять небольшие однострочные функции на лету. Позаимствованные из Lisp, так называемые lambda-функции могут быть использованы везде, где требуется функция

1. Как осуществляется документирование кода согласно PEP257?

Документирование кода в python - достаточно важный аспект, ведь от нее порой зависит читаемость и быстрота понимания вашего кода, как другими людьми, так и вами через полгода. PEP 257 описывает соглашения, связанные со строками документации python, рассказывает о том, как нужно документировать python код. Цель этого PEP - стандартизировать структуру строк документации: что они должны в себя включать, и как это написать (не касаясь вопроса синтаксиса строк документации). Этот PEP описывает соглашения, а не правила или синтаксис.

1. В чем особенность однострочных и многострочных форм строк документации?

Для согласованности, всегда используйте """triple double quotes""" для строк документации. Используйте r"""raw triple double quotes""" , если вы будете использовать обратную косую черту в строке документации. Существует две формы строк документации: однострочная и многострочная.