**Лабораторная работа № 4**

**Тема: “Создание собственных функций, модулей и пакетов “.**

**Задание на выполнение:**

**Задание 1. Создание собственного модуля**

Программист на Python всегда может создать собственный модуль, чтобы использовать его в нескольких своих программах. Требуется создать модуль с функциями для вычисления площадей прямоугольника, треугольника и круга:

from math import pi, pow

def rectangle(a, b):

return round(a \* b, 2)

def triangle(a, h):

return round(0.5 \* a \* h, 2)

def circle(r):

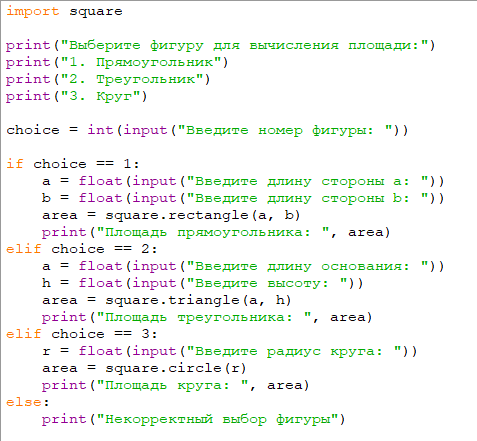
return round(pi \* pow(r, 2), 2)

Поместите данный код в отдельный файл square.py.

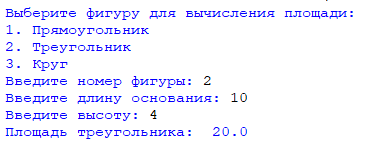
Когда интерпретатор Python встречает команду импорта, то просматривает на наличие файла-модуля определенные каталоги. Если вы сохраните файл-модуль и файл-программу в одном каталоге, то интерпретатор без труда найдет модуль.

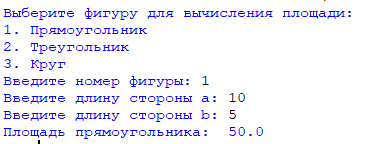
Поместите файл square.py в тот же каталог, где будет исполняемая программа. Ее код должен включать инструкцию импорта модуля square (при импорте расширение файла не указывается) и вызов той функции и с теми параметрами, которые ввел пользователь. Т. е. у пользователя надо спросить, площадь какой фигуры он хочет вычислить. Далее запросить у него аргументы для соответствующей функции. Передать их в функцию из модуля square, а полученный оттуда результат вывести на экран.

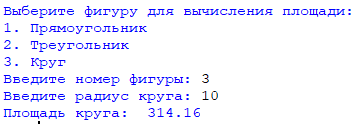
*Код программы:*

**

*Выполнение программы:*

**

**

**

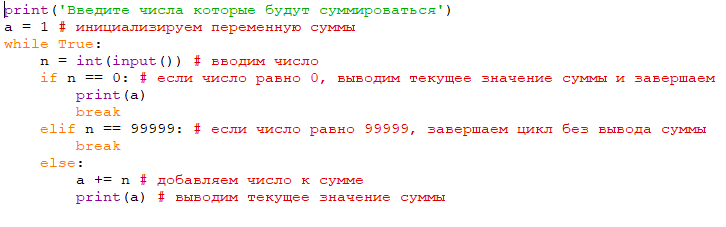
**ЗАДАНИЕ 2. Напишите программы с использованием функций:**

10. Суммировать вводимые числа, среди которых нет нулевых. При

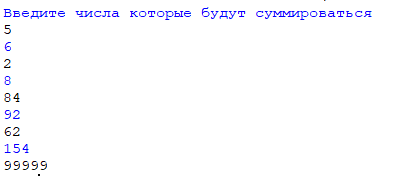
вводе нуля обеспечить вывод текущего значения суммы. При вводе

числа 99999 закончить работу.

Код программы:



Выполнение программы:



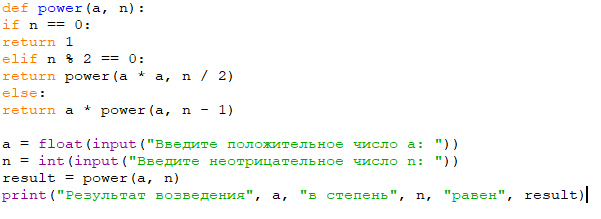
**Задание 3**

Дано действительное положительное число a и целое

неотрицательное число n. Вычислите a n  не используя циклы, возведение в

степень через \*\* и функцию math.pow(), а используя рекуррентное

соотношение a n =a⋅a n-1 . Решение оформите в виде функции power(a, n).





**КОНТРОЛЬНЫЕ ВОПРОСЫ**

1.Что такое функция? Какие преимущества дает использование функций?

Функция в программировании представляет собой обособленный участок кода, который можно

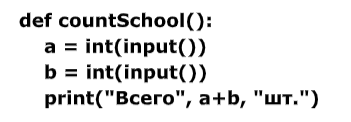
вызывать, обратившись к нему по имени, которым он был назван. При вызове происходит выполнение

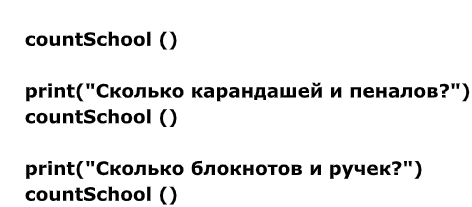
команд тела функции.

Функции можно сравнить с небольшими программками, которые сами по себе,

т. е. автономно, не исполняются, а встраиваются в обычную программу.

2.Приведите пример определения и вызова функции в программе.



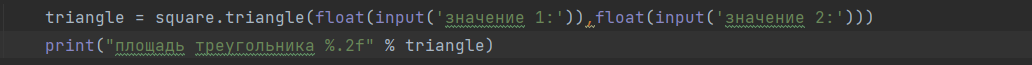


3.В чем разница между локальными и глобальными переменными?

Локальные переменные видны только в локальной области видимости, которой может

выступать отдельно взятая функция. Глобальные переменные видны во всей программе.

4.Как передать результаты работы функции в вызывающую программу?



5.Что означает «передача параметра по ссылке»?

Передача по ссылке позволяет возвратить из функции сразу несколько значений.

6.Сложные структуры данных следует передавать «по ссылке» или «по значению»?

По ссылке

7.Что понимают под аргументом функции?

Конкретные значения, которые передаются в функцию при ее вызове, будем называть

аргументами

8.Приведите примеры встроенных функций

Abs() chr() int() dir()

9.Каково назначение модулей?

Каждый модуль содержит коллекцию функций и классов, предназначенных для решения задач из

определенной области.

10.Приведите пример использования модуля math в программе.



11.Какие способы импортирования модулей вы знаете? Чем они отличаются?

Импортирование всего модуля: import module\_name. Этот метод импортируетвесь модуль и поз

воляет обращаться к его содержимому через точку, например,  module\_name.functionя\_name().

Импортирование конкретной функции из модуля:

from module\_name import function\_name. Этот метод импортирует только указанную функцию из модуля

 и позволяет обращаться к ней напрямую, без указания названиямодуля, например, function\_name().

Импортирование всего модуля с псевдонимом:

import module\_name as alias\_name. Этот метод импортирует весь модуль, но

позволяет обращаться к его содержимому через псевдоним, например,  alias\_name.function\_name().

Импортирование конкретной функции из модуля с псевдонимом: from

module\_name import function\_name as alias\_name. Этот метод импортирует только

указанню функцию из модуля и позволяет обращаться к ней через псевдоним ,  например, alias\_name().

Каждый из этих способов имеет свои преимущества и недостатки, и выбор зависит от конкретн

ой задачи и предпочтений программиста.