**Импорт модулей. Базовые модули (os.path, random, math). Pip и работа с пакетами.**

Функции языка Python позволяют упаковывать фрагменты программного кода, чтобы его можно было многократно использовать во всей программе. Но что делать если требуется использовать одни и те же функции в разных программах? Для решения такой проблемы в языке Python существуют модули.

**Импорт модулей**

Модуль – это средство объединения функций в коллекции, что бы их можно было использовать в разных программах. Любую программу, написанную на языке Python и расположенную в файле с расширением .py, можно считать отдельным модулем. Главное различие между модулем и программой состоит в том, что программа предназначен для того, чтобы её запускали, а модуль предназначен для того, чтобы его импортировали, т.е. подключали к основной программе, и использовали в программе.

Синтаксис подключения модулей к программе имеет несколько вариантов:

* import module\_name – импортируем в программу модуль с именем module\_name;
* import module\_name1, module\_name 2, … , module\_nameN – импортируем в программу N модулей;
* import module\_name as alias - импортируем в программу модуль с именем module\_name и с помощью ключевого слова as даём ему псевдоним alias;

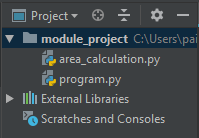
Ключевое слово import указывает на то, что после него следует перечисление имен модулей, которые будут использованы в проекте. Ключевое слово as используется для того, чтобы давать короткие названия для модулей внутри программы. Обычно, подключение модулей помещается в начало программы. Имена модулей содержат только буквы нижнего регистра и цифры, в качестве разделите между словами используется “\_”. Остальные символы не должны присутствовать в названии модуля.

Для обращения к функциям и переменным определенных в модули используется следующая конструкция: имя\_модуля(или псевдоним).название\_функции (переменной).

Язык программирования Python поставляется с большим количеством дополнительных модулей, которые объединены в стандартную библиотеку Python. Модули из этой библиотеки импортируются ранее описанным способом.

Пример:

Для примера, в проекте создадим 2 файла: area\_calculation и program. Структура проекта примет следующий вид:



В файле area\_calculation.py определим функции для расчета площади следующих фигур:

* прямоугольник;
* круг;
* треугольник;

Модуль area\_calculation:

# Импортируем модуль math из стандартной библиотеки языка Python  
import math  
  
  
# Фнкция возвращает площадь прямоугольника  
def rectangle\_area(a, b):  
 return a \* b  
  
  
# Фнкция возвращает площадь круга  
# Обратите внимание, число Pi берется из модуля math, что повышает "читаемость" кода.  
def circle\_area(r):  
 return math.pi \* r \*\* 2  
  
  
# Фнкция возвращает площадь треугольника  
# Для расчета квадратного корня числа, используется метод sqrt из модуля math  
def triangle\_area(a, b, c):  
 p = (a + b + c) / 2  
 return math.sqrt(p \* (p - a) \* (p - b) \* (p - c))

Программа program.py:

import area\_calculation as ac, math  
  
# Выведет на консоль площадь прямоугольника со сторонами 10 и 5  
print(ac.rectangle\_area(10, 5))  
  
# Выведет на консоль площадь круга с радиусом 5.  
# Полученный результат округляется до ближайшего большего числа с помощью функции ceil.  
# Функция определена в модули math  
print(math.ceil(ac.circle\_area(5)))  
  
# Выведет на консоль площадь прямоугольника со сторонами 3,4,5  
print(ac.triangle\_area(3, 4, 5))

**Базовые модули (os.path, random, math)**

Как уже упоминалось ранее, в стандартную библиотеку языка Python входит множество модулей. В этом пособии, мы рассмотрим следующие модули:

* os.path - является вложенным модулем в модуль os, и реализует полезные функции для работы с путями.
* random - предоставляет функции для генерации случайных чисел, букв, случайного выбора элементов последовательности.
* math - этот модуль предоставляет обширный функционал для работы с числами.