

1.6. Функции

Программа – это последовательность операторов. Часто случается, что какая-то часть программы неоднократно повторяется. Например, если в задаче требуется вычислить попарное расстояние между тремя точками на плоскости, то одна и та же формула применяется для вычисления трех расстояний.

Kod:

```
from math import sqrt

# здесь нужно ввести или задать координаты трех точек

r_ab = sqrt((x_a - x_b) ** 2 + (y_a - y_b) ** 2)
r_bc = sqrt((x_b - x_c) ** 2 + (y_b - y_c) ** 2)
r_ac = sqrt((x_c - x_a) ** 2 + (y_c - y_a) ** 2)
```

Чтобы устранить подобного рода избыточность программного кода, используют понятие **функции**. Повторяющийся (да и не только) блок программного кода обозначают некоторым уникальным именем, чтобы потом при необходимости обратиться к нему по этому имени.

Для определения функции используется ключевое слово **def**, после которого указывается ее имя, круглые скобки и двоеточие.

Функции, вычисляющей расстояние между двумя точками, дадим имя **compute_len**, ее объявление имеет вид:

```
def compute_len():
```

Внутри функции можно использовать любые операторы языка, в нашем случае должна вычисляться длина стороны:

```
def compute_len():
    len_line = sqrt((x_b - x_a) ** 2 + (y_b - y_a) ** 2)
```

Но реализованная таким образом функция вычисляет только расстояние между точками **A** и **B**. Для того чтобы вычислить расстояние между двумя произвольными точками с координатами (x_0, y_0) и (x_1, y_1) , необходимо в заголовке функции указать параметры (аргументы), с которыми она будет работать, а внутри функции использовать эти параметры как обычные переменные:

```
def compute_len(x_0, y_0, x_1, y_1):
    len_line = sqrt((x_1 - x_0) ** 2 + (y_1 - y_0) ** 2)
```

Расстояние между точками вычислено. Для того чтобы результат стал известен вне функции, используется оператор **return**, после которого указывается переменная (или выражение), значение которой является результатом выполнения функции (говорят, что это значение «возвращается» из функции):

```
def compute_len(x_0, y_0, x_1, y_1):
    len_line = sqrt((x_1 - x_0) ** 2 + (y_1 - y_0) ** 2)
    return len_line
```

Данная функция реализована и может быть использована для вычисления расстояния между двумя точками на плоскости, заданными своими координатами. Для этого к ней обращаются (вызывают ее) с конкретными значениями координат, а результат заносят в некоторую переменную. Например, чтобы вычислить расстояние между точками $A(x_A, y_A)$ и $B(x_B, y_B)$:

С помощью функции **compute_len** можно вычислить расстояние между тремя точками на плоскости $A(x_A, y_A)$, $B(x_B, y_B)$ и $C(x_C, y_C)$.

Kod:

```
from math import sqrt

def compute_len(x_0, y_0, x_1, y_1):
    len_line = sqrt((x_1 - x_0) ** 2 + (y_1 - y_0) ** 2)
    return len_line
```

```
x_a = float(input("x_a = "))
y_a = float(input("y_a = "))
x_b = float(input("x_b = "))
y_b = float(input("y_b = "))
x_c = float(input("x_c = "))
y_c = float(input("y_c = "))
```

```
r_ab = compute_len(x_a, y_a, x_b, y_b)
r_bc = compute_len(x_b, y_b, x_c, y_c)
r_ac = compute_len(x_a, y_a, x_c, y_c)
```

```
print("r_ab =", round(r_ab, 4))
print("r_bc =", round(r_bc, 4))
print("r_ac =", round(r_ac, 4))
```

Результат:

```
x_a = 4
y_a = 1
x_b = 8.2
y_b = -3
x_c = 4.5
y_c = -1
r_ab = 5.8
r_bc = 4.2059
r_ac = 2.0616
```

Задания для самопроверки

1. Какое значение получит переменная **res** после выполнения следующего фрагмента программы:

```
def compute_resist(r_1, r_2):
    r = r_1 * r_2 / (r_1 + r_2)
    return r
res = compute_resist(15.0, 11.0)
```

2. Описать функцию, которая вычисляет прибыль от вложения суммы **x** руб на **n** месяцев под **k** процентов годовых с капитализацией процентов. Формула для вычисления прибыли:

$$income = x \cdot \left(1 + \frac{k}{12 \cdot 100}\right)^n - x$$

Заголовок функции должен иметь вид:

```
# deposit - сумма вклада, interest_rate -процентная ставка,
# amount_months - количество месяцев
def compute_income(deposit, interest_rate, amount_months):
```

С помощью этой функции посчитать прибыль от вложения для различных входных данных:

Тест 1

```
x = 100000
k = 4
n = 18
Прибыль: 6173.06
```

Тест 2

```
x = 50600
k = 4.5
n = 36
Прибыль: 7298.94
```

Тест 3

```
x = 1000000
k = 5
n = 6
Прибыль: 25261.868
```

1.7. Форматный вывод

Для наглядного представления выходных данных в Python используется форматный вывод.

Форматный вывод осуществляется с помощью оператора **%**, который используется в операторе **print()**:

```
print("шаблон вывода" %(список вывода))
```

Список вывода – это перечисленные через запятую переменные, которые выводятся на экран. **Шаблон вывода** – строка текста со вставленными туда элементами форматирования для отображения каждого элемента списка вывода. Допускается форматный

вывод как числовых значений, так и строк. В таблице 10 приведены форматы для вывода наиболее распространённых типов данных.

Таблица 10 – Описание форматов

Тип данных	Формат	Описание
Целое число	%8d	Целое число размещается в 8 позициях, прижато к правому краю
Целое число	%-8d	Целое число размещается в 8 позициях, прижато к левому краю
Вещественное число	%8.4f	Вещественное число размещается в 8 позициях, выводится 4 знака после запятой, прижато к правому краю, незаполненные позиции после запятой заполняются 0
Вещественное число	%-8.4f	Вещественное число размещается в 8 позициях, выводится 4 знака после запятой, прижато к левому краю, незаполненные позиции после запятой заполняются 0
Строка текста	%8s	Если длина строки меньше 8 символов, то она размещается в 8 позициях, прижата к правому краю, в противном случае формат не учитывается
Строка текста	%-8s	Если длина строки меньше 8 символов, то она размещается в 8 позициях, прижата к левому краю, в противном случае формат не учитывается

На рисунке 2 показано соответствие между форматом и переменной. Так, первый встретившийся в шаблоне вывода формат определяет способ вывода первой переменной в списке вывода, второй формат используется для второй переменной.

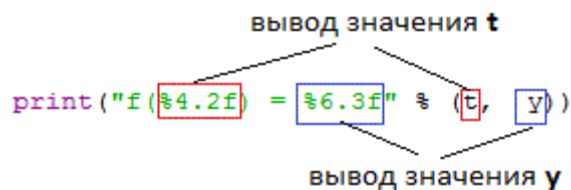


Рисунок 2. Форматный вывод

В строке шаблонов на рисунке 2 заданы строковые константы, которые должны быть выведены на экран ("f(", ") = "), а также форматы для двух переменных:

%4.2f – вещественное число, выводится в 4 позициях, количество знаков после запятой – 2, используется как формат для вывода переменной **t**;

%6.3f – вещественное число выводится в 6 позициях, количество знаков после запятой – 3, используется как формат для вывода значения **y**.

Задание для самопроверки

Дан фрагмент программы:

```
r = 5.9247836
v = 1.9747316
print("r = %__, v = %__" % (r, v))
```

В операторе **print** укажите форматы для каждой переменной так, чтобы в консоли был выведен следующий результат:

```
r = 5.925, v = 1.9747
```

1.8. Задача, вычисления на плоскости

Задача. Дан треугольник на плоскости, заданный координатами своих вершин. Для этого треугольника вычислить:

- длины сторон;
- периметр;

- площадь;
- величины углов в градусах.

1.8.1. Неформальная постановка задачи

Для наглядности нарисуем треугольник на плоскости (рисунок 3). Каждая его вершина представляет собой точку. Для описания точки на плоскости используются две координаты, например, точка А на рисунке 3 имеет координаты 8 по оси ОХ и 4 по оси ОУ.

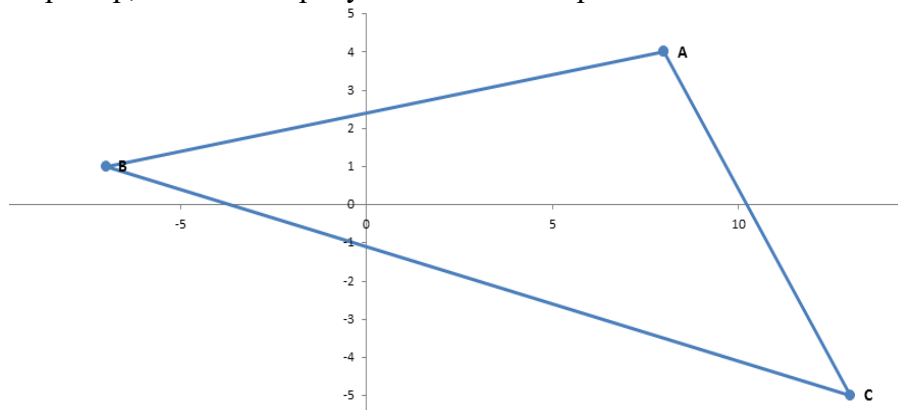


Рисунок 3. Треугольник ABC

Следовательно, входными данными нашей задачи будут 3 пары переменных, каждая пара – это координаты **x** и **y** вершины треугольника. Примерами координат являются числа 3 или -4.98. Ограничений на ввод координат треугольника нет.

Но есть ограничения на расположение точек. Так треугольник построить нельзя, если:

- или все его вершины совпадают,
- или две любые вершины совпадают,
- или все вершины лежат на одной прямой.

В этом случае никакие характеристики треугольника посчитать нельзя, поскольку это либо точка, либо отрезок.

Все эти случаи можно учесть с помощью признака существования треугольника: «треугольник существует, если сумма длин двух любых его сторон больше третьей».

В таблице 11 приведены входные данные задачи. Также (для сокращения записи, чтобы не повторять таблицу в разделе проектирования) в этой таблице укажем имена переменных и их тип. Например, координату **x** точки А назовем **x_a**, координату точки А – **y_a**. Все переменные имеют тип **float**.

Таблица 11 – Входные данные

Описание	Имя	Тип	Пример	Недопустимые значения
Координата x точки А	x_a	float	5, -4.5	Нет
Координата y точки А	y_a	float	-15.1, 6.5	Нет
Координата x точки В	x_b	float	0, 1	Нет
Координата y точки В	y_b	float	-1.1, 9.45	Нет
Координата x точки С	x_c	float	2.55, 5.1	Нет
Координата y точки С	y_c	float	-2, -6.7	Нет
Условие существования треугольника: сумма двух любых его сторон больше третьей				

Результатом выполнения нашей программы (выходными данными) должны стать различные характеристики треугольника: длины его сторон, площадь и периметр, величины углов в градусах. Имена переменным выберем, как это принято в геометрии. Длину стороны назовем маленькой буквой, которая соответствует вершине, лежащей напротив стороны. Например, сторона **AB**, ее длина **c**, так как эта сторона лежит напротив вершины **C**. Площадь

и периметр назовем **s** и **p**. Величины углов обозначаются специальным знаком угла « $\angle$ », после которого пишется имя вершины. Использовать символ угла в имени мы не можем, поэтому вместо него напишем **angle**, для соединения с именем вершины воспользуемся подчеркиванием. Типы результатов – вещественные числа. В таблице 12 приведены выходные данные задачи.

Таблица 12 – Выходные данные

Описание	Имя	Тип	Пример
Длина стороны АВ	c	float	2.704, 16.6623
Длина стороны ВС	a	float	13.327, 16.3218
Длина стороны АС	b	float	14.866, 6.2412
Периметр ABC	p	float	30.897, 39.2253
Площадь ABC	s	float	15.6, 50.46
Величина угла А	angle_A	float	9.061, 82.1801
Величина угла В	angle_B	float	50.906, 76.0373
Величина угла С	angle_C	float	120.033, 21.7826

На следующем этапе неформальной постановки задачи необходимо определиться с алгоритмами и формулами, которые нам понадобятся в процессе реализации. Эти формулы можно найти в справочнике по геометрии или на любом информационном ресурсе, посвященном аналитической геометрии.

1. *Длины сторон:*

$$AB = c = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2},$$

$$BC = a = \sqrt{(x_C - x_B)^2 + (y_C - y_B)^2},$$

$$AC = b = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}.$$

2. *Периметр:*

$$p = a + b + c.$$

3. *Площадь (формула Герона):*

$$s = \sqrt{\frac{p}{2} \cdot \left(\frac{p}{2} - a\right) \cdot \left(\frac{p}{2} - b\right) \cdot \left(\frac{p}{2} - c\right)}.$$

4. Для вычисления углов используется теорема косинусов:

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos(\angle C).$$

Выразим величины углы (значения будут получены в радианах):

$$\angle A = \arccos\left(\frac{c^2 + b^2 - a^2}{2 \cdot c \cdot b}\right),$$

$$\angle B = \arccos\left(\frac{c^2 + a^2 - b^2}{2 \cdot c \cdot a}\right),$$

$$\angle C = \arccos\left(\frac{a^2 + b^2 - c^2}{2 \cdot a \cdot b}\right).$$

Последним шагом неформальной постановки задачи является описание **функциональных возможностей** программы. Наша программа должна:

- ввести координаты точки А;
- ввести координаты точки В;
- ввести координаты точки С;
- вычислить длины сторон треугольника;
- проверить условие существования треугольника; если треугольник не существует, вывести сообщение;

- вычислить периметр треугольника;
- вычислить площадь треугольника;
- вычислить величины углов треугольника в радианах;
- перевести значения величин углов в градусы;
- вывести длины сторон, периметр, площадь и углы.

1.8.2. Проектирование и реализация программы

Для реализации программы мы по-прежнему выбираем язык Python, именование входных и выходных данных, выбор типов данных уже выполнено на этапе неформальной постановки задачи. Сразу перейдем к описанию проекта программы:

```
описать функцию вычисления длины стороны по координатам ее вершин
описать функцию вычисления площади по длинам трех сторон
описать функцию вычисления угла по длинам трех сторон
ввести входные данные
вычислить длины сторон треугольника
если входные данные неверные
то
    вывести сообщение об ошибке
иначе
    вычислить площадь
    вычислить периметр
    вычислить углы
    вывести результаты
```

Ниже приведен код программы, последовательность реализации программы описана в виде комментариев.

Код:

```
# Импортировать функцию для перевода радиан в градусы,
# арккосинуса и квадратного корня.
from math import sqrt, acos, degrees

#Реализовать функцию для вычисления длины отрезка
#с координатами концов (x0, y0) и (x1, y1).
def compute_len(x_0, y_0, x_1, y_1):
    len_line = sqrt((x_1 - x_0) ** 2 + (y_1 - y_0) ** 2)
    return len_line

#Реализовать функцию для вычисления площади треугольника
# по трем сторонам a_1, a_2, a_3.
def compute_area(a_1, a_2, a_3):
    p = (a_1 + a_2 + a_3) / 2
    area = sqrt(p * (p - a_1) * (p - a_2) * (p - a_3))
    return area

#Реализовать функцию для вычисления угла треугольника в градусах,
# если известны длины сторон a_1, a_2, a_3.
def compute_angle(a_1, a_2, a_3):
    angle_rad = acos((a_1 ** 2 + a_2 ** 2 - a_3 ** 2) / (2 * a_1 * a_2))
    return degrees(angle_rad)

#Ввести координаты вершин треугольника с подсказкой пользователю.
x_a = float(input("x_a = "))
y_a = float(input("y_a = "))
x_b = float(input("x_b = "))
y_b = float(input("y_b = "))
x_c = float(input("x_c = "))
y_c = float(input("y_c = "))
```

```

# Вычислить длины сторон треугольника.
c = compute_len(x_a, y_a, x_b, y_b)
a = compute_len(x_b, y_b, x_c, y_c)
b = compute_len(x_a, y_a, x_c, y_c)

# Проверить, существует ли треугольник, построенный по заданным точкам,
# если нет - выдать сообщение пользователю.
if a + b <= c or b + c <= a or a + c <= b:
    print("Треугольник не существует")
# В противном случае вычислить площадь, периметр, величины углов.
else:
    s = compute_area(a, b, c)
    p = a + b + c
    angle_A = compute_angle(c, b, a)
    angle_B = compute_angle(c, a, b)
    angle_C = compute_angle(a, b, c)

# Вывести результаты вычисления на экран,
# округлив их до 3 знаков после запятой.
print("Стороны:", round(a, 3), round(b, 3), round(c, 3))
print("Площадь:", round(s, 3))
print("Периметр:", round(p, 3))
print("Углы:", round(angle_A, 3), round(angle_B, 3), round(angle_C, 3))

```

Результат:

тест 1

```

x_a = 5
y_a = -15.1
x_b = 0
y_b = -1.1
x_c = 2.55
y_c = -2
Стороны: 2.704 13.327 14.866
Площадь: 15.6
Периметр: 30.897
Углы : 9.061 50.906 120.033

```

тест 2

```

x_a = 1
y_a = 1
x_b = 2
y_b = 2
x_c = 3
y_c = 3
Треугольник не существует

```

1.8.3. Отладка и тестирование

На этом этапе, после того как найдены и исправлены все ошибки (то есть программа запускается и выдает результат), проверяется работоспособность программы на различных входных данных. В таблице 13 приведены результаты тестирования.

Таблица 13 – Тестирование программы для различных входных данных

Набор входных данных	Правильный результат	Верно?
x _a = 0 y _a = 0 x _b = 9 y _b = 1 x _c = 9 y _c = 1	Треугольник не существует	✓
x _a = 0 y _a = 1 x _b = 0 y _b = 2 x _c = 0 y _c = 5	Треугольник не существует	✓
x _a = 0 y _a = 10 x _b = -5 y _b = 0 x _c = 5 y _c = 0	Стороны: 10.0 11.18 11.18 Площадь: 50.0 Периметр: 32.361 Углы: 53.13 63.435 63.435	✓
x _a = 0 y _a = 1 x _b = 0 y _b = 0 x _c = 1 y _c = 0	Стороны: 1.0 1.414 1.0 Площадь: 0.5 Периметр: 3.414 Углы: 45.0 90.0 45.0	✓
x _a = 1 y _a = -5 x _b = 5 y _b = -1 x _c = 6 y _c = 8	Стороны: 9.055 14.318 8.062 Площадь: 3.5 Периметр: 31.435 Углы: 35.48 113.405 31.115	✓

Задание для самопроверки

Решить следующую задачу:

Дан треугольник ABC на плоскости, заданный координатами своих вершин. Для этого треугольника вычислить:

- радиус вписанной в треугольник окружности;
- радиус описанной вокруг треугольника окружности;
- сумму длин трех медиан треугольника.

Результаты округлить до 4 знаков после запятой.

Проверить программу для следующих входных данных:

Тест 1

x_a = 1
y_a = 1
x_b = 0
y_b = 0
x_c = 1
y_c = 1

Результат: Треугольник не существует

Тест 2

x_a = -12.8
y_a = 3.4
x_b = -7.7
y_b = 8.6
x_c = -14.6
y_c = -3.5

Результат: 0.9113 14.0042 22.8319

Формулы для вычислений

1. Радиус вписанной окружности вычисляется:

$$r = \sqrt{\frac{\left(\frac{p}{2} - a\right) \cdot \left(\frac{p}{2} - b\right) \cdot \left(\frac{p}{2} - c\right)}{\frac{p}{2}}}.$$

2. Радиус описанной вокруг треугольника окружности:

$$R = \frac{a \cdot b \cdot c}{4 \cdot s}.$$

3. Длины медиан вычисляются по формулам:

$$M_a = \frac{1}{2} \cdot \sqrt{2 \cdot (c^2 + b^2) - a^2},$$

$$M_b = \frac{1}{2} \cdot \sqrt{2 \cdot (c^2 + a^2) - b^2},$$

$$M_c = \frac{1}{2} \cdot \sqrt{2 \cdot (a^2 + b^2) - c^2}.$$