# Урок 07.01. Вещественные числа. Функции float и round

Мы с Вами уже многие задачи умеем делать, хотя применяли всегда только целые и строковые переменные. Сегодня мы с Вами познакомимся с дробными или правильно их называть вещественными числами. И для этого рассмотрим задачу.

## Задача «Букмекерская контора»

Допустим, букмекерская контора нам заказала написать программу, которая вычисляет выигрыш игрока по его ставка и по введенному коэффициенту. Мы с Вами легко можем сделать такую программу

bet = int(input('Введите ставку: '))

coefficient = int(input('Введите коэффициент: '))

win = bet \* coefficient

print('Выигрыш составляет: ', win, 'руб.')

Попробуем ввести 1000 руб., по ставке 2, получилось – 2000 руб. Всё верно. А теперь давайте введем чуть более реалистичные данные: 1000руб., по ставке 1.2 (это вещественное число, потому что мы использовали разделитель точка). И вот наша программа не работает, выдает ошибку «ValueError: invalid literal for int() with base 10: '1.2'» – перевод, звучит так: «Ошибка значения: не правильный литерал для int() с базой 10: '1.2'» (под базой следует понимать систему исчисления, в данном случае десятичная система исчисления). Это произошло из-за того, что мы вещественное число пытаемся преобразовать в целое (команда int). Чтобы решить данную проблему, как вы, думаю, уже догадались, нужно просто преобразовать ввод коэффициента ставки не в целое число, а в вещественное. Это делает при помощи команды float , это сокращение от floating point (переводится как «плавающая точка», поэтому такие числа программисты и называют «числа с плавающей точкой»). И теперь наша программа будет выглядеть так

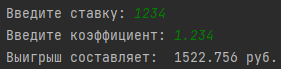
bet = int(input('Введите ставку: '))

coefficient = float(input('Введите коэффициент: '))

win = bet \* coefficient

print('Выигрыш составляет: ', win, 'руб.')

Отлично, наша программа работает. Вот пример:



Всё хорошо, вот только как мы можем отдать выигрыш 75,6 копеек? Никак. Поэтому нам нужно округлить (не отбросить!) результат до двух десятых знаков. И это делается при помощи функции round

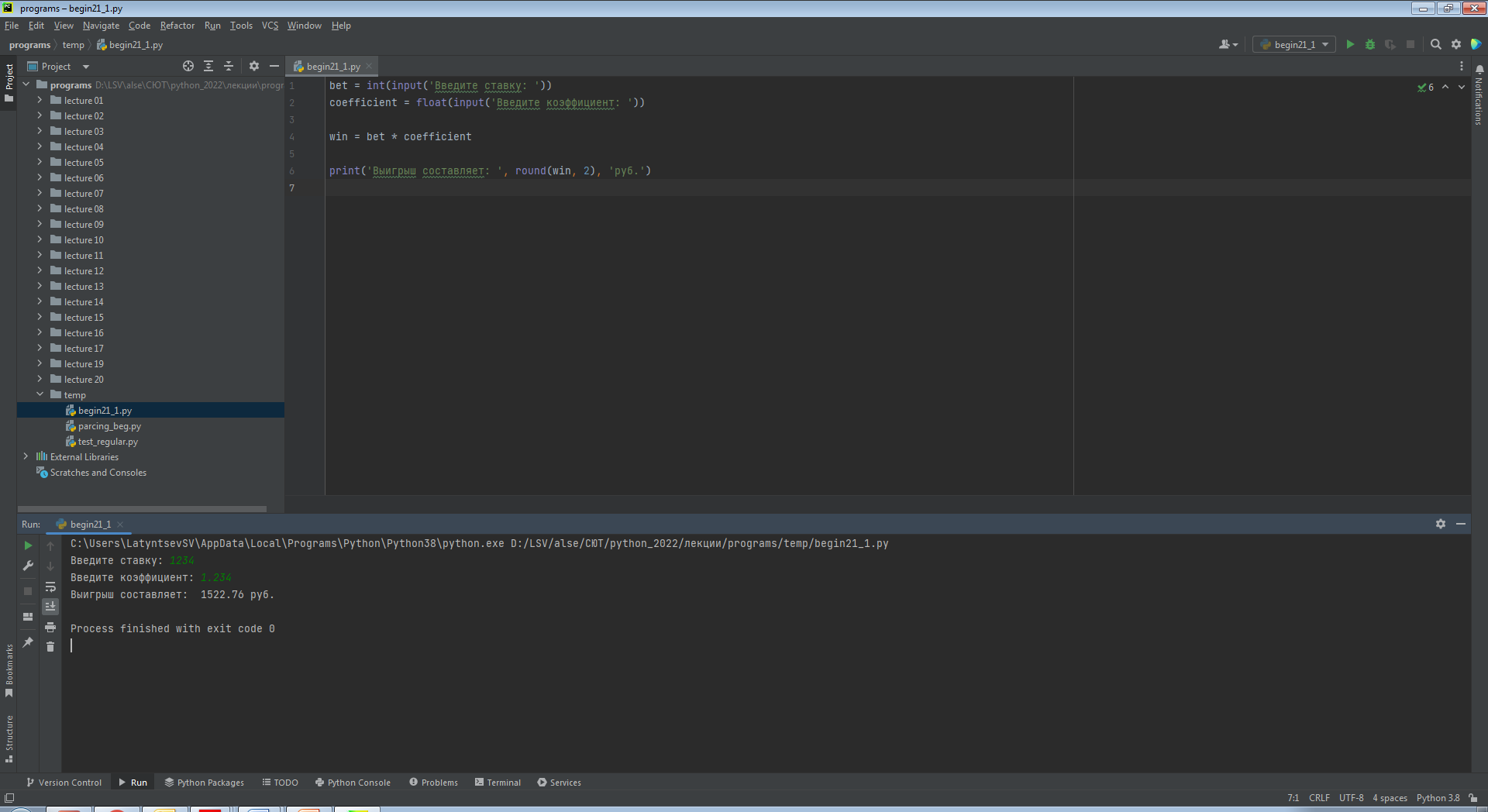
bet = int(input('Введите ставку: '))

coefficient = float(input('Введите коэффициент: '))

win = bet \* coefficient

print('Выигрыш составляет: ', round(win, 2), 'руб.')

У этой функции в скобках указывается число (или переменная), которое нужно округлить, потом запятая, затем число знаков после запятой для округления.



Вот теперь всё в порядке!

Скоро мы рассмотрим ещё больше интересного с вещественными числами.

Для закрепления рассмотрим еще одну задачу.

## Задача «Индекс массы тела»

Алексей работает диетологом в частной клинике, каждый день он принимает пациентов разных возрастов и с разными показателями роста (в метрах) и веса (в кг). Для каждого человека ему нужно считать индекс массы тела - это вес поделить на рост в квадрате. По государственным стандартам индекс округляется до двух знаков после точки. Затем по этому индексу определяется, всё ли в порядке у человека с массой тела: если до 18.5, то недобор; до 25 - всё нормально, до 30 уже идёт избыток, а после 30 - ожирение. Напишите такую программу для Алексея.

user\_weight = float(input("Вес пациента: "))

user\_height = float(input("Рост пациента: "))

result = round(user\_weight / user\_height \*\* 2, 2)

if result < 18.5:

print("Недобор")

elif result < 25:

print("Норма")

elif result < 30:

print("Избыток")

else:

print("Ожирение")

## Математические функции. Работа с модулем math

В математике встречаются такие страшные слова как квадратный корень, синус, косинус и им подобные. На уроках математике с ними часто приходится работать. И сегодня мы с вами попробуем их использовать. И, как всегда, практиковаться будем на задаче.

## Задача «GPS – навигатор»

Нам необходимо написать программу, которая будет считать расстояния от нас до координат, которые будут вводиться пользователем. При этом пользователь стоит в координатах (0, 0).

Для этого нужно проанализировать задачу: нам даны координаты точки и нам нужно определить длину отрезка, соединяющих центр с этой точкой. Из школьного курса геометрии известно, что для этого нужно посчитать гипотенузу (красная линия) прямоугольного треугольника. Чтобы найти гипотенузу нужно взять квадратный корень из суммы квадратов катетов (зеленые линии).

с

b

a

Формула будет такая .

С текущими нашими знаниями мы можем решить данную задачу следующим образом

x = float(input('Введите координату x: '))

y = float(input('Введите координату y: '))

distance = (x \*\* 2 + y \*\* 2) \*\* 0.5

print('Расстояние:', distance)

Однако, оптимальное по времени будет другое решение, которое использует функции, написанные за нас другими людьми. И такие математические действия находятся в модуле math, а функция вычисления квадратного корня называется sqrt. И программу будет выглядеть так

import math

x = float(input('Введите координату x: '))

y = float(input('Введите координату y: '))

distance = math.sqrt(x \*\* 2 + y \*\* 2)

print('Расстояние:', distance)

Обращаю ваше внимание на следующее:

* чтобы подключить новый модуль нужно написать слово import и название модуля;
* чтобы вызвать функцию из подключенного модуля нужно написать имя модуля, поставить точку и написать имя функции.

Список функций модуля можно увидеть, если написать имя модуля и поставить точку. Более того показывается подсказка: что и сколько нужно передать в функцию.

Также обращаю внимание, что модули нужно импортировать в самом начале программы.

Давайте попробуем теперь решить несколько обратную задачу.

Задача «Радар»

У нас есть радар и он сканирует местность вокруг нас, при обнаружении цели мы можем получить расстояние до неё и угол на который повернут радар. Наша задача по этим числам найти координаты цели.

с

(x, y)

α

Опять же из курса школьной геометрии решение такой задачи не является сложной. Представим такое решение:

Тогда такая программу будет выглядеть так

import math

distance = float(input('Введите расстояние (километры): '))

alpha = float(input('Введите угол (градусы): '))

x = distance \* math.sin(math.radians(alpha))

y = distance \* math.cos(math.radians(alpha))

print('Координата x:', x)

print('Координата y:', y)

В программе мы дополнительно выполнили преобразование градусы в радианы (math.radians), потому что функции sin и cos работают с углами в радианах.