

Упражнения по программированию главы 4

```
# coding: utf-8
```

Упражнение по программированию 4.1. Сборщик ошибок

```
# Инициализировать переменные для системных ошибок и
# общего количества собранных ошибок.
bugs = 0
total = 0

# Получить количество системных ошибок, собранных в каждый день
for day in range(5):
    bugs = int(input('Введите количество системных ошибок, собранных сегодня: '))
    total += bugs

# Показать общее количество собранных системных ошибок.
print ('Общее количество собранных системных ошибок: ', total)
```

Упражнение по программированию 4.2. Сожженные калории

```
# Объявить и инициализировать переменную
# для калорий, сжигаемых в минуту.
caloriesPerMinute = 4.2

# Объявить переменные для количества сожженных калорий и
# количества минут.
caloriesBurned = 0.0
minutes = 0

print ('Минуты\t\tСожженные калории')
print ('-----')

# Исполнить цикл for, чтобы показать сожженные калории.
for minutes in range(10, 31, 5):
    caloriesBurned = caloriesPerMinute * minutes
    print (minutes, "\t\t", caloriesBurned)
```

Упражнение по программированию 4.3. Анализ бюджета

```
# Объявить переменные для хранения суммы бюджета,
# потраченной суммы, разницы и общей суммы.
budget = 0.0
difference = 0.0
spent = 1.0      # инициализировать цикл while
total = 0.0

# Получить от пользователя сумму бюджета.
budget = float(input('Введите сумму, выделенную на месяц: '))
```

```

# Получить от пользователя общую потраченную сумму.
while spent != 0:
    spent = float(input('Введите потраченную сумму (0, чтобы выйти): '))
    # прибавить к total
    total += spent

# Определить, превысил или сэкономил пользователь свой бюджет,
# и показать результат.
print ('Выделено: $', format(budget, '.2f'))
print ('Потрачено: $', format(total, '.2f'))

if budget > total:
    difference = budget - total
    print ('Вы потратили на $', format(difference, '.2f'), \
          'меньше бюджета. МОЛОДЦЫ!')
elif budget < total:
    difference = total - budget
    print ('Вы потратили на $', format(difference, '.2f'), \
          'сверх бюджета. В СЛЕДУЮЩИЙ РАЗ ПЛАНИРУЙТЕ ЛУЧШЕ!')
else:
    print ('Расходы соответствуют бюджету. ХОРОШЕЕ ПЛАНИРОВАНИЕ!')

```

Упражнение по программированию 4.4. Пройденное расстояние

```

# Объявить переменные для расстояния, скорости и времени.
distance = 0
speed = 0.0
time = 0

# Получить от пользователя скорость.
speed = float(input('Введите скорость транспортного средства в км/ч: '))

# Получить время в пути и преобразовать целое число в полные часы.
time = int(float(input('Введите количество часов в пути: ')))

# Показать расстояние, пройденное в зависимости от часа.
print ('Час\t Пройденное расстояние')
print ('-----')

for hour in range(1, time + 1):
    distance = hour * speed
    print (hour, '\t', distance)

```

Упражнение по программированию 4.5. Средняя толщина дождевых осадков

```

# Объявить переменные для суммарной толщины дождевых осадков,
# ежемесячной толщины дождевых осадков, среднемесячной толщины дождевых осадков,
# количества лет и общего количества месяцев.
totalRainfall = 0.0

```

```

monthRainfall = 0.0
averageRainfall = 0.0
years = 0
totalMonths = 0

# Получить количество лет
years = int(input('Введите количество лет: '))

# Получить толщину дождевых осадков ежемесячно
for year in range(years):
    print ('Для года ', year + 1, ':')
    for month in range(12):
        monthRainfall = float(input( \
            'Введите толщину дождевых осадков для месяца: '))
        # Прибавить к общему количеству месяцев
        totalMonths += 1
        # Прибавить к общему количеству дождевых осадков
        totalRainfall += monthRainfall

# Вычислить среднюю толщину дождевых осадков
averageRainfall = totalRainfall / totalMonths

print('Для ', totalMonths, 'месяцев')
print('Суммарная толщина дождевых осадков: ', format(totalRainfall, '.2f'), 'см')
print('Среднемесечная толщина дождевых осадков: ', \
    format(averageRainfall, '.2f'), 'см')

```

Упражнение по программированию 4.6. Таблица соответствия между градусами Цельсия и градусами Фаренгейта

```

# Объявить переменную для температуры в градусах
# по шкале Фаренгейта.
fahrenheit = 0.0

# Вычислить и напечатать величину для каждой температуры.
print ('Цельсий\t\tФаренгейт')
print ('-----')

for celsiusDegree in range(21):
    fahrenheit = ((9 * celsiusDegree) / 5) + 32
    print (celsiusDegree, '\t\t', fahrenheit)

# Упражнение по программированию 4.7

# Мелкая монета для зарплаты

# Объявить переменные для количества копеек в день,
# количества дней и общего количества копеек.
dayPennies = 1
numDays = 0
total = 0.0

```

```

# Получить от пользователей количество дней.
numDays = int(input('Введите количество дней: '))

# Показать таблицу с заработной платой за каждый день.
print ('День\tРубли')
print ('-----')

for day in range(1, numDays + 1):
    print(day, '\t\t', float(dayPennies / 100))
    total += dayPennies
    dayPennies *= 2

# Показать общую заработную плату.
print('Общая заработная плата за', numDays, \
      'дней составит: ', float(total/100), 'рублей')

```

Упражнение по программированию 4.8. Сумма чисел

```

# Объявить переменные для количества и итоговой суммы.
number = 1.0    # Инициализировать для цикла while
total = 0.0

# Продолжать добавлять числа до тех пор пока они положительные.
while number > 0:
    number = float(input('Введите положительное число' \
                          ' (отрицательное, чтобы выйти): '))

    # Проверить, что число положительное, чтобы
    # не изменить значение итоговой суммы.
    if number > 0:
        total = total + number

# Показать итоговую сумму.
print ('Всего:', format(total, '.2f'))

```

Упражнение по программированию 4.9. Уровень океана

```

# Константа для подъема уровня в год.
RISE_PER_YEAR = 1.6

# Объявить переменную для хранения величины подъема.
rise = 0.0

# Вычислить и напечатать величину подъема в каждом году.
print ('Год\t\t Подъем (в миллиметрах)')
print ('-----')

for year in range(25):
    rise += RISE_PER_YEAR

```

```
print ((year + 1), '\t\t', format(rise, '.2f'))
```

Упражнение по программированию 4.10. Рост платы за обучение

```
# Константы для роста платы за обучение в год и
# начальной стоимости обучения.
INCREASE_PER_YEAR = 0.03
STARTING_AMOUNT = 145000.0

# Объявить переменную для хранения оплаты за обучение.
tuition = STARTING_AMOUNT

# Вычислить и напечатать рост стоимости обучения в каждом году.
print ('Год\t Плановая стоимость обучения (за курс)')
print ('-----')

for year in range(5):
    tuition += (tuition * INCREASE_PER_YEAR)
    print ((year + 1), '\t', '$', \
            format(tuition, '.2f'))
```

Упражнение по программированию 4.11. Потеря массы

```
# Получить исходную массу тела пользователя.
print('Какой Ваш исходный вес? ', end='')
weight = int(input())

# Показать таблицу потери веса.
for month in range(1, 6):
    weight = weight - 1.5
    print('В конце месяца', month,
          'Вы будете весить', weight, 'кг.')
```

Упражнение по программированию 4.12. Вычисление факториала числа

```
# Инициализировать число нулем.
number = 0

# Получить от пользователя допустимое число.
while number <= 0:
    number = int(input('Введите неотрицательное число: '))

# Инициализировать накопительную переменную.
factorial = 1

# Вычислить факториал числа.
for factor in range(1, number + 1):
    factorial *= factor

# Показать факториал числа.
print('Факториал числа', number, 'равняется', factorial)
```

Упражнение по программированию 4.13. Популяция

```
num = 0      # Исходное количество организмов
avg = 0.0    # Среднедневный рост количества
days = 0    # Количество дней для размножения

# Получить от пользователя допустимое значение для
# исходного количества организмов.
while num <= 0:
    num = int(input('Исходное количество организмов: '))

# Получить от пользователя допустимое значение для
# среднедневного роста количества.
while avg <= 0:
    avg = float(input('Среднедневный рост количества: '))

# Получить от пользователя допустимое значение для
# количества дней для размножения.
while days <= 0:
    days = int(input('Количество дней для размножения: '))

# Определить, был ли введен среднедневный рост,
# как целое число; если да, то поделить на
# 100, что отформатировать это значение как процент.
if avg >= 1.0:
    avg /= 100.0

# Вычислить и напечатать размер увеличения в каждый день.
print ('День\t\t Популяция')
print ('-----')

for day in range(days):
    # Применить рост после первого дня.
    if (day > 0):
        num += (num * avg)
    print ((day + 1), '\t\t\t', num)
```

Упражнение по программированию 4.14. Вложенные циклы для рисования узора (1)

```
character = '*' # Печатаемый символ
size = 7        # Количество строк и столбцов

# Выполнить итеративный обход строк.
for row in range(size):

    # Каждая строка имеет меньше столбцов.
    for col in range(size, row, -1):
        print(character, end='')
```

```
# Перейти к следующей строке.  
print()
```

Упражнение по программированию 4.15. Вложенные циклы для рисования узора (2)

```
character = '#' # Печатаемый символ  
numRows = 7     # Количество строк  
space = ' '     # Пробельный символ  
  
# Выполнить итеративный обход строк.  
for row in range(numRows):  
  
    # Каждая строка содержит 2 столбца и  
    # затем номер строки.  
    for col in range(row + 2):  
  
        # Напечатать символ в первом  
        # и последнем столбце.  
        if col == 0 or col == row + 1:  
            print(character, end='')  
  
        # Добавить пробелы между символами.  
        else:  
            print (space, end='')  
  
    # Перейти к следующей строке.  
    print()
```

Упражнение по программированию 4.16. Черепашня графика: повторение квадратов

```
import turtle  
  
# Именованные константы  
STARTING_X = -4  
STARTING_Y = 4  
STARTING_SIZE = 8  
NUM_SQUARES = 50  
STEP = 4  
NUM_SIDES = 4  
ANGLE = 90  
ANIMATION_SPEED = 0  
  
# Задать скорость анимации и спрятать черепаху.  
turtle.speed(ANIMATION_SPEED)  
turtle.hideturtle()  
  
# Задать исходные координаты x и y, а также исходный размер.  
x = STARTING_X
```

```

y = STARTING_Y
size = STARTING_SIZE

# Начертить узор.
for count in range(NUM_SQUARES):
    # Спозиционировать черепаху.
    turtle.penup()
    turtle.goto(x, y)
    turtle.pendown()

    # Начертить квадрат
    for s in range(NUM_SIDES):
        turtle.forward(size)
        turtle.right(ANGLE)

    # Подготовить следующий квадрат.
    x = x - STEP
    y = y + STEP
    size = size + STEP

```

Упражнение по программированию 4.17. Черепашня графика: звездочный узор

```

import turtle

# Именованные константы
INITIAL_ANGLE = 45
ANGLE_STEP = 135
NUM_LINES = 8
LENGTH = 200
ANIMATION_SPEED = 0

# Инициализировать черепаху.
turtle.hideturtle()
turtle.speed(ANIMATION_SPEED)
turtle.left(INITIAL_ANGLE)

# Начертить линии.
for count in range(NUM_LINES):
    turtle.forward(LENGTH)
    turtle.left(ANGLE_STEP)
turtle.done()

```

Упражнение по программированию 4.18. Черепашня графика: гипнотический узор

```

import turtle

# Именованные константы
ANIMATION_SPEED = 0

```



```

NUM_LINES = 50
STARTING_LENGTH = 1
ENDING_LENGTH = 500
STEP = 10
ANGLE = 90

# Инициализировать черепаху.
turtle.hideturtle()
turtle.speed(ANIMATION_SPEED)

# Начертить линии.
for x in range(STARTING_LENGTH, ENDING_LENGTH, STEP):
    turtle.forward(x)
    turtle.left(ANGLE)
turtle.done()

```

Упражнение по программированию 4.19. Черепашня графика: знак STOP

```

import math
import turtle

# Именованные константы
WINDOW_WIDTH = 400
WINDOW_HEIGHT = 400
ANIMATION_SPEED = 0
NUM_SIDES = 8
LENGTH = 100
ANGLE = 45
TEXT_X = -5
TEXT_Y = -10

# Размер окна.
turtle.setup(WINDOW_WIDTH, WINDOW_HEIGHT)

# Вычислить диаметр шестиугольника, чтобы его
# можно было центрировать в графическом окне.
#
#           s
#
#   -----
#
#   / | \
# s / | \
# / | x \ s
# / | \
# |-----|
# | x |
# | |
#
# Чтобы получить диаметр:
# diameter = s + 2 * x
#
# Мы знаем, что s равняется 100.
# Применить теорему Пифагора, чтобы получить x:
# s^2 = x^2 + x^2

```

```

#   s^2 = 2*x^2
#   x = s / sqrt(2)
s = LENGTH
x = s / math.sqrt(2)
diameter = s + (2 * x)

# Инициализировать черепаху.
turtle.penup()
turtle.hideturtle()
turtle.speed(ANIMATION_SPEED)

# Переместить черепаху в исходную точку.
starting_x = (0 - (diameter / 2)) + ((WINDOW_WIDTH - diameter) / 2)
starting_y = (s / 2) + x
turtle.goto(starting_x, starting_y)
turtle.pendown()

# Начертить шестиугольник.
for x in range(NUM_SIDES):
    turtle.forward(LENGTH)
    turtle.right(ANGLE)

# Показать надпись 'STOP'
turtle.penup()
turtle.goto(TEXT_X, TEXT_Y)
turtle.pendown()
turtle.write('STOP')
turtle.done()

```