

Пример 7.15 (создание и тестирование подпрограмм). Создадим функции для работы с векторами.

<code>random_list()</code>	Создание и заполнение списка случайными целыми числами из интервала <code>[1; max_number]</code> . Параметры подпрограммы: <code>count_elements</code> (количество элементов в списке) и <code>max_number</code> (значение правой границы отрезка). Возвращаемый результат: новый список.
<code>input_list()</code>	Создание и заполнение списка введенными с клавиатуры целыми числами. Параметр подпрограммы: <code>count_elements</code> (количество элементов в списке). Возвращаемый результат: новый список.
<code>print_list()</code>	Вывод списка. Параметр подпрограммы: <code>_list</code> (список).
<code>mult_list()</code>	Умножение числа на вектор. Параметры подпрограммы: <code>_list</code> (список) и <code>number</code> (значение числа). Возвращаемый результат: новый список.

Для подпрограммы несущественны имена переменных, с которыми она работает (они никак не связаны с реальными именами объектов).

Функция `random_list(count_elements, max_number)`

```
import random

# Создание списка (элементы -- случайные целые числа)
def random_list(count_elements, max_number):
    new_list = []
    for i in range(count_elements):
        x = random.randint(1, max_number)
        new_list = new_list + [x]
    return new_list

# строки для проверки работы функции
a = random_list(5, 35) # создание и заполнение списка
print(a)               # [28, 25, 27, 19, 11]
```

Каждую отдельную подпрограмму требуется проверять на соответствие заданию (в частности, количество и тип параметров) и на правильность возвращаемого результата. В дальнейшем строки для проверки функции можно убрать (но лучше просто закомментировать).

Функция input_list(count_elements)

```
# Создание списка (элементы вводятся с клавиатуры)
def input_list(count_elements):
    new_list = []
    for i in range(count_elements):
        elem = int(input('[' + str(i) + '] = '))
        new_list.append(elem)
    return new_list

# строки для проверки работы функции
a = input_list(5) # создание и заполнение списка из 6 элементов
print(a)
```

Результат работы программы

```
[0] = 0
[1] = -3
[2] = 5
[3] = 10
[4] = -12
[0, -3, 5, 10, -12]
```

Функция print_list(_list)

```
# Печать списка
def print_list(_list):
    for elem in _list: print('%5d' %elem, end = ' ')
    print()

# строки для проверки работы функции
a = [4, 55, 78, 100] # задаем произвольный список
print_list(a)       # результат:    4    55    78    100
```

Функция mult_list(_list, number)

```
# Умножение вектора на целое число
def mult_list(_list, number):
    new_list = []
    for i in range(len(_list)):
        new_elem = _list[i] * number
        new_list = new_list + [new_elem]
    return new_list
```

```
# строки для проверки работы функции
a = [4, 55, 78, 10] # задаем произвольный список
new_a = mult_list(a, 2)
print(new_a)          # результат: [8, 110, 156, 20]
```

Пример 7.16 (меню и диалог с пользователем). Для вектора (списка) с целочисленными элементами написать программу, позволяющую выполнить одну из операций, выбранных пользователем:

- 1: Создание и заполнение списка случайными целыми числами;
- 2: Создание и заполнение списка введенными с клавиатуры целыми числами;
- 3: Вывод списка;
- 4: $A = tA$ (t — любое целое число, введенное пользователем);
- 5: $D = kA$ (k — случайное двузначное натуральное число);
- 9: Печать меню;
- 0: Выход из программы.

Решение. В примере 7.15 были описаны все нужные функции. Осталось записать все функции в начале одного файла, добавить функцию печати меню и описать тело программы для формирования диалога с пользователем. В нашей программе есть ограничение: пользователь обязательно должен начать работу с создания списков (запишем это в виде рекомендации при запуске программы).

В полном тексте программы для функции `random_list()` приведен пример документирования функции. Команда `help(идентификатор функции)` позволит вывести информацию о функции, которая была задокументирована.

Диалог пользователя и вызов справки по функции

```
!!!: Начните с заполнения списков (п.1 или п.2)
```

```
-----
1 : Получение элементов случайным образом
2 : Ввод элементов с клавиатуры
3 : Вывод списка
4 : A = tA (t - любое целое число)
5 : D = kA (k - случайное двузн. натур. число)
9 : Вывод меню
0 : Выход из программы
-----
```

```
Выберите команду меню (9: Вывод меню): 0
```

```
>>> help(random_list)
```

Help on function random_list in module __main__:

random_list(count_elements, max_number)

Функция для создания списка из count_elements случайных чисел из отрезка [1; max_number].

Аргументы:

int count_elements (число элементов)

int max_number (число элементов)

Тип возвращаемого значения: list (список)

Меню и диалог с пользователем

```
import random
```

```
def random_list(count_elements, max_number):  
    '''
```

Функция для создания списка из count_elements случайных чисел из отрезка [1; max_number].

Аргументы:

int count_elements (число элементов)

int max_number (число элементов)

Тип возвращаемого значения: list (список)

```
    '''
```

```
    new_list = []
```

```
    for i in range(count_elements):
```

```
        x = random.randint(1, max_number)
```

```
        new_list = new_list + [x]
```

```
    return new_list
```

```
# Создание списка (элементы вводятся с клавиатуры)
```

```
def input_list(count_elements):
```

```
    new_list = []
```

```
    for i in range(count_elements):
```

```
        elem = int(input('[' + str(i) + '] = '))
```

```
        new_list.append(elem)
```

```
    return new_list
```

```
# Печать списка
```



```
def print_list(_list):
    for elem in _list: print('%5d' %elem, end = ' ')
    print()

# Умножение вектора на целое число
def mult_list(_list, number):
    new_list = []
    for i in range(len(_list)):
        new_elem = _list[i] * number
        new_list = new_list + [new_elem]
    return new_list

# функция печати меню программы
def PrintMenu():
    print('-----')
    print('1 : Получение элементов случайным образом')
    print('2 : Ввод элементов с клавиатуры')
    print('3 : Вывод списка')
    print('4 : A = tA (t - любое целое число)')
    print('5 : D = kA (k - случайное двузн. натур. число)')
    print('9 : Вывод меню')
    print('0 : Выход из программы')

# ---- основная программа -----
random.seed()

vv = 9 # сразу выберем п.9 - печать меню
print('!!!: Начните с заполнения списков (п.1 или п.2)')
while vv != 0:
    if vv == 1:
        n = int(input('Количество элементов: '))
        max_number = int(input('Макс. значение: '))
        list_a = random_list(n, max_number)
        print('Получены элементы списка. Печать: п.3')
    elif vv == 2:
        n = int(input('Количество элементов: '))
```

```
    print('Введите элементы: ')
    list_a = input_list(n)
elif vv == 3:
    print_list(list_a)
elif vv == 4:
    t = int(input('t = '))
    list_a = mult_list(list_a, t)
    print('Список изменён a[i] = a[i] * % t. печать: п.3'' % t)
elif vv == 5:
    k = random.randint(10, 99)
    D = mult_list(list_a, k)
    print('Элементы нового списка = a[i] * % k' % k)
    print_list(D)
elif vv == 9:
    PrintMenu()
print('-----')
vv = int(input('Выберите команду меню (9: Вывод меню): '))
```

Пример работы программы и диалога с пользователем см. [далее](#).

Результат работы программы

!!!: Начните с заполнения списков (п.1 или п.2)

1 : Получение элементов случайным образом
2 : Ввод элементов с клавиатуры
3 : Вывод списка
4 : $A = tA$ (t - любое целое число)
5 : $D = kA$ (k - случайное двузн. натур. число)
9 : Вывод меню
0 : Выход из программы

Выберите команду меню (9: Вывод меню): 1

Количество элементов: 9

Макс. значение: 11

Получены элементы списка. Печать: п.3

Выберите команду меню (9: Вывод меню): 3

1 6 6 5 8 10 11 1 8

Выберите команду меню (9: Вывод меню): 4

$t = 2$

Список изменён $a[i] = a[i] * 2$. Печать: п.3

Выберите команду меню (9: Вывод меню): 3

2 12 12 10 16 20 22 2 16

Выберите команду меню (9: Вывод меню): 5

Элементы нового списка = $a[i] * 96$

192 1152 1152 960 1536 1920 2112 192 1536

Выберите команду меню (9: Вывод меню): 0

Пример 7.17 (меню и диалог с пользователем-2). Для двух натуральных чисел a и b написать программу, позволяющую выполнить одну из операций, выбранных пользователем:

- 0 : Печать меню;
- 1 : Получение чисел a и b случайным образом;
- 2 : Ввод чисел a и b с клавиатуры;
- 3 : $c = a \cdot 2^b$ (c — новое число);
- 4 : $b = \frac{a+d}{2}$, где $d = (b+1) \cdot 2^a$;
- 5 : Выход из программы.

Указание. Все операции с числами должны быть оформлены в виде подпрограмм.

```
from random import randint

# функция печати меню программы
def menu():
    zag = '=' * 20 + ' MENU ' + '=' * 20
    print(zag)
    print('| 0) Вывод меню |')
    print('| 1) Получение чисел a и b случайным образом |')
    print('| 2) Ввод чисел a и b с клавиатуры |')
    print('| 3) Получение нового числа... |')
    print('| 4) Среднее арифметическое чисел... |')
    print('| 5) Завершение работы (выход) |')
    print('=' * len(zag))

# ввод одного целого числа с клавиатуры
def inputNumb(s = 'Введите натуральное число: '):
    return int(input(s))

# функция для случайного получения целого числа из интервала [1, k]
def randomNumb(k):
    return randint(1, k)

# произведение вида m * 2^n, m и n -- натуральные числа
def prod(m, n):
    return m << n

# среднее арифметическое двух чисел
```

```
def srAr(x, y):  
    return 0.5 * (x + y)  
  
# основная программа  
menu()  
p = int(input('Введите номер пункта '))  
while p != 5:  
    if p == 0:  
        menu()  
    elif p == 1:  
        m = int(input('Введите границы для random '))  
        a = randomNumb(m)  
        b = randomNumb(m)  
        print('Первое число', a)  
        print('Второе число', b)  
    elif p == 2:  
        a = inputNumb()  
        b = inputNumb()  
    elif p == 3:  
        c = prod(a, b)  
        print('Новое число', c)  
    else:  
        s = srAr(a, prod(b + 1, a))  
        print('Среднее арифметическое', s)  
    p = int(input('Выберите команду меню (0: Вывод меню): '))
```

Обратите внимание: программа работает с двумя числами, но подпрограммы получения и печати чисел написаны для одного числа и просто дважды вызываются.