

## Практическое занятие №

**Тема:** составление программ со списками в IDE PyCharm Community.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ со списками в IDE PyCharm Community.

### Задание №1:

**Постановка задачи:** Разработать программу, выводящую сначала четные, а затем нечетные элементы списка A.

**Текст программы:**

```
'''Вариант 13.
Дан список A размера N. Вывести вначале его элементы с четными номерами
(в порядке возрастания номеров), а затем – элементы с нечетными
номерами (также в порядке возрастания номеров): A2, A4, A6, . . ., A1,
A3, A5, ... . Условный оператор не использовать.'''
from random import randint
def typecheck(x, type_:type):
    while type(x) != type_:
        try:
            return type_(x)
        except ValueError:
            x = input(f'Неправильный ввод {"целого " if type_ == int
else "вещественного " if type_ == float else ""}числа!\nПовторите ввод:
')
a, odd, even = [], [], []
n = typecheck(input("Введите размер списка: "), int)
for i in range(n):
    a.append(randint(-100, 100))
for i in range(0, len(a), 2):
    even.append(a[i])
for i in range(1, len(a), 2):
    odd.append(a[i])
print(f'Полученный список:{a}\nСписок с четными
индексами:{even}\nСписок с нечётными индексами:{odd}')
```

**Протокол работы программы:**

Введите размер списка: 12

Полученный список: [-96, 58, 91, 31, 53, 59, -42, 94, -6, -91, -26, -88]

Список с четными индексами: [-96, 91, 53, -42, -6, -26]  
Список с нечётными индексами: [58, 31, 59, 94, -91, -88]

## Задание №2:

**Постановка задачи:** Разработать программу, которая находит два соседних элемента списка, сумма которых наиболее близка к числу R.

### Текст программы:

```
'''Вариант 13.  
Дано число R и список размера N. Найти два соседних элемента списка,  
сумма которых наиболее близка к числу R, и вывести эти элементы в  
порядке возрастания их индексов (определение наиболее близких чисел –  
то есть такой элемент AK, для которого величина |AK – R| является  
минимальной).'''  
from random import randint  
def typecheck(x, type_:type):  
    while type(x) != type_:  
        try:  
            return type_(x)  
        except ValueError:  
            x = input(f'Неправильный ввод {"целого " if type_ == int  
else "вещественного " if type_ == float else ""}числа!\nПовторите ввод:  
' )  
def find_neighbours(R, N):  
    min_diff = float('inf')  
    min_pair = None  
    for i in range(len(N) - 1):  
        diff = abs(N[i] + N[i + 1] - R)  
        if diff < min_diff:  
            min_diff = diff  
            min_pair = (N[i], N[i + 1])  
    return min_pair  
R = randint(1, 100)  
N = []  
Ns = typecheck(input('Введите размер списка: '), int)  
for i in range(Ns):  
    N.append(randint(1, 100))  
print(f'Число R: {R}\nНайденные соседние элементы:{find_neighbours(R,  
N)}')
```

### Протокол работы программы:

Введите размер списка: 14

Число R: 75

Найденные соседние элементы:(17, 57)

### Задание №3:

**Постановка задачи:** Разработать программу, осуществляющую сдвиг списка влево на K позиций, а в последним K элементам списка присваивала 0.

#### Текст программы:

```
'''Вариант 13.  
Дан список размера N и целое число K ( $1 < K < N$ ). Осуществить сдвиг  
элементов списка влево на K позиций (при этом  $A_N$  перейдет в  $A_{N-K}$ ,  $A_{N-1}$   
– в  $A_{N-K-1}$ , ..  $A_{K+1}$  – в  $A_1$ , а исходное значение K первых элементов будет  
потеряно). Последние K элементов полученного списка положить равными  
0.'''  
from random import randint  
def typecheck(x, type_:type):  
    while type(x) != type_:  
        try:  
            return type_(x)  
        except ValueError:  
            x = input(f'Неправильный ввод {"целого " if type_ == int  
else "вещественного " if type_ == float else ""}числа!\nПовторите ввод:  
' )  
def shift_left(lst, K):  
    # Сдвигаем элементы влево на K позиций  
    new_lst = lst[K:] + [0] * K  
    return new_lst  
N = typecheck(input("Введите размер списка: "), int)  
K = typecheck(input("Введите количество позиций для сдвига: "), int)  
A = []  
while K < N:  
    print('Число K должно быть больше N.')  
    N = typecheck(input("Введите размер списка: "), int)  
    K = typecheck(input("Введите количество позиций для сдвига: "),  
int)  
for i in range(N):  
    A.append(randint(0, 100))  
print("Исходный список:", A)  
print("Сдвинутый список:", shift_left(A, K))
```

**Протокол работы программы:**

Введите размер списка: 12

Введите количество позиций для сдвига: 2

Полученный список: [100, 15, 48, 29, 89, 95, 27, 46, 45, 93, 25, 86]

Сдвинутый список: [48, 29, 89, 95, 27, 46, 45, 93, 25, 86, 0, 0]

**Вывод:** Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ со списками в IDE PyCharm Community.