

## Практическое занятие № 6

**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.

### Постановка задачи.

1. Дан список A размера N и целое число K ( $1 < K < N$ ). Вывести элементы списка с порядковыми номерами, кратными K: AK, A2\*K, A3\*K,... . Условный оператор не использовать.

**Тип алгоритма:** циклический

### Текст программы 1:

```
1  #1. Дан список A размера N и целое число K (1 < K < N). Вывести элементы списка с порядковыми номерами,
2  # кратными K: AK, A2*K, A3*K,... . Условный оператор не использовать.
3
4  n = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
5
6  k = input('Придумай число не больше 10: ')
7  > while type(k) != int: ...
15
16  result = []
17
18  for i in range(0, 10, k): #перебираю список с интервалом равным k
19      result.append(i)
20
21  del result[0] #убираю ноль
22  print(result)
```

### Протокол работы программы:

Придумай число не больше 10: 3

[3, 6, 9]

Process finished with exit code 0

### Постановка задачи.

2. Дан список размера N. Найти количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).

Тип алгоритма: ветвящийся

### Текст программы 2:

```
1  #2. Дан список размера N. Найти количество его промежутков монотонности
2  # (то есть участков, на которых его элементы возрастают или убывают).
3
4  import random
5
6  n = input('Сколько в списке будет символов? ')
7
8  > while type(n) != int: ...
14
15  a = []
16
17  for i in range(n):
18      a.append(random.randint(-100, b: 100))
19
20  if len(a) <= 1: #если в списке 1 или меньше элементов то кол-во промежутков тоже 1 или 0
21      print(len(a))
22
23
24  def k_monotony(a): 1 usage  pinktasha1
25      k = 1 # счётчик кол-ва промежутков монотонности
26      monotony = None #true будет возрастающим промежутком, а false - убывающим
27      for i in range(1, len(a)):
28          if a[i] > a[i - 1]: # проверяю элемент на возрастание
29              if monotony == False: # если промежуток был убывающим, то +1 к счётчику
30                  k += 1
31                  monotony = True
32          elif a[i] < a[i - 1]: # проверяю элемент на убывание
33              if monotony == True:
34                  k += 1
35                  monotony = False
36      return k
```

### Протокол работы программы:

Сколько в списке будет символов? 9

Твой список: [61, -94, 52, -13, -78, 70, -61, 48, 3]

Количество промежутков монотонности в нём: 7

Process finished with exit code 0

### Постановка задачи.

3. Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию (при этом A1 перейдет в A2, A2 — в A3, ..., AN-1 — в AN, а исходное значение последнего элемента будет потеряно). Первый элемент полученного списка положить равным 0.

Тип алгоритма: ветвящийся

### Текст программы 2:

```
1  #3. Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию
2  # (при этом A1 перейдет в A2, A2 — в A3, ..., AN-1 — в AN, а исходное значение последнего элемента будет потеряно)
3  # Первый элемент полученного списка положить равным 0.
4
5  import random
6
7  n = input('Сколько в списке будет символов? ')
8
9  while type(n) != int: #обработка исключений
10     try:
11         n = int(n)
12     except ValueError :
13         print('Неправильно ввели!')
14         n = input('Введите число заново: ') #пусть пользователь введёт число заново
15
16  original_a = []
17
18  for i in range(n):
19      original_a.append(random.randint(-100, b: 100))
20
21
22  def offset(original_a): 1 usage  pinktasha1
23      result_a = [0] + original_a[:-1]
24      return result_a
25
26
27  print('Твой список: ', original_a)
28  print('Твой изменённый список: ', offset(original_a))
```

### Протокол работы программы:

Сколько в списке будет символов? 8

Твой список: [-46, 9, -82, 53, 95, 24, 51, -87]

Твой изменённый список: [0, -46, 9, -82, 53, 95, 24, 51]

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия выработал(а) навыки составления программ с использованием функций в IDE PyCharm Community. выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.