

## 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 целочисленный список размера 10. Вывести вначале все содержащиеся в данном списке четные числа в порядке возрастания их индексов, а затем — все нечетные числа в порядке убывания их индексов.

**Тип алгоритма:** Циклический

**Текст программы:**

```
'''
Вариант 9.
Дан целочисленный список размера 10. Вывести вначале все содержащиеся в
данном
списке четные числа в порядке возрастания их индексов, а затем — все нечетные
числа в порядке убывания их индексов.
'''
numbers = [10, 17, 20, 25, 30, 33, 40, 45, 50, 55]

chetnie_numbers = []
nechetnie_numbers = []

#разделение чисел на четные и нечетные
for number in numbers:
    if number % 2 == 0:
        chetnie_numbers.append(number)
    else:
        nechetnie_numbers.append(number)

print("Четные числа: ", chetnie_numbers)

print("Нечетные числа: ", nechetnie_numbers[::-1])
```

**Протокол работы программы:**

Четные числа: [10, 20, 30, 40, 50]

Нечетные числа: [55, 45, 33, 25, 17]

**Постановка задачи 2:** Дан список размера N. Найти количество участков, на которых его элементы монотонно убывают.

**Тип алгоритма:** Циклический.

**Текст программы:**

```

'''
Вариант 9.
Дан список размера N. Найти количество участков, на которых его элементы
монотонно убывают.
'''
import random

N = 10
numbers = [random.randint(1, 10) for x in range(N)]
print("Сгенерированный список:", numbers)

#подсчет участков, где элементы монотонно убывают
count = 0

for i in range(1, len(numbers)):
    #проверяем, начинается ли новый участок убывания
    if numbers[i] < numbers[i - 1]:
        #если это первый элемент убывающего участка
        if i == 1 or numbers[i - 1] >= numbers[i - 2]:
            count += 1

print("Количество участков, где элементы монотонно убывают:", count)

```

### Протокол работы программы:

Сгенерированный список: [5, 8, 5, 2, 5, 7, 7, 10, 5, 8]

Количество участков, где элементы монотонно убывают: 2

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