

## Практическое занятие №6

**Тема:** составление программ со списками в IDE PyCharmCommunity.

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

**Постановка задачи №1:** Дан список A размера N. Найти максимальный элемент из его элементов с нечетными номерами: A1, A3, A5, ... .

**Тип алгоритма:** циклический

**Текст программы:**

```
from random import randint

n = int(input("Введите количество элементов в списке: ")) # Ввод
данных
a = [randint(1, 100) for i in range(n)] # Задаем список
print("Исходный список: ", a) # Вывод списка
p = len(a)
b = []
for i in range(p):
    if i % 2 != 0:
        b.append(a[i])

print("Список элементов с нечетными номерами: ", b) # Вывод списка
print("Максимальный элемент из списка с нечетными номерами: ", max(b))
# Вывод списка
```

**Протокол работы программы:**

```
Введите количество элементов в списке: 10

Исходный список:  [40, 56, 72, 84, 58, 89, 22, 36, 52, 34]

Список элементов с нечетными номерами:  [56, 84, 89, 36, 34]

Максимальный элемент из списка с нечетными номерами:  89

Process finished with exit code 0
```

**Постановка задачи №2:** Дан целочисленный список A размера N ( $< 15$ ). Переписать в новый целочисленный список B все элементы с порядковыми номерами, кратными трем (3, 6, ...), и вывести размер полученного списка B и его содержимое. Условный оператор не использовать.

**Тип алгоритма:** циклический

**Текст программы:**

```

from random import randint

n = int(input('Введите n (n < 15): ')) # Ввод данных
a = [randint(1, 100) for i in range(n)]
print(a)
b = []
for i in range(0, len(a), 3):
    b.append(a[i]) # Добавление нового значения
print(f'Длина списка b: {len(b)}') # Вывод длины
print(f'Список b: {b}') # Вывод списка

```

### Протокол работы программы:

```

Введите n (n < 15): 5

[52, 39, 73, 42, 92]

Длина списка b: 2

Список b: [52, 42]

Process finished with exit code 0

```

**Постановка задачи №3:** Дано множество A из N точек ( $N > 2$ , точки заданы своими координатами x, y). Найти наименьший периметр треугольника, вершины которого принадлежат различным точкам множества A, и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества A). Расстояние R между точками с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$  вычисляется по формуле:  $R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ . Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат

**Тип алгоритма:** смешанный

### Текст программы:

```

from random import randint
from math import sqrt

# функция для нахождения расстояния между точками
def r(point1, point2):
    return sqrt((point2[0] - point1[0]) ** 2 + (point2[1] - point1[1]) ** 2)

n = int(input('n > 2: '))
my_list = []

# создаем лист с кортежами координат x и y
for i in range(n):
    xy = (randint(-100, 100), randint(-100, 100))
    if xy not in my_list:
        my_list.append(xy)

# переменные для минимального периметра и минимальных точек

```

```

min_p = 99999
min_1 = tuple()
min_2 = tuple()
min_3 = tuple()

# тремя циклами сравниваем периметр треугольника с вершинами в каждом наборе
точек
for i in my_list:
    for j in my_list:
        for k in my_list:

            # вычисляем стороны треугольника
            a = r(i, j)
            b = r(i, k)
            c = r(j, k)

            # вводим проверку, чтобы точки не повторялись
            if a == b or b == c or a == c:
                continue

            p = a + b + c
            # если найденный периметр меньше минимального, то:
            if p < min_p:
                min_p = p
                min_1 = i
                min_2 = j
                min_3 = k

print("Периметр треугольника: ", min_p)
print("Точки :", min_1, min_2, min_3)

```

### Протокол работы программы:

n > 2: 10

Периметр треугольника: 95.7299891551339

Точки : (20, 62) (-7, 57) (-23, 43)

Process finished with exit code 0

**Вывод:** в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ со списками в IDE PyCharm Community. Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.