

## Практическое занятие №5

**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. Найти минимальный элемент из его элементов с четными номерами: A<sub>2</sub>, A<sub>4</sub>, A<sub>6</sub>, ... .
2. Дан целочисленный список A размера N. Переписать в новый целочисленный список B все четные числа из исходного списка (в том же порядке) и вывести размер полученного списка B и его содержимое.
3. Дано множество A из N точек ( $N > 2$ , точки заданы своими координатами x, y). Найти наибольший 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<sub>1</sub>, y<sub>1</sub>) и (x<sub>2</sub>, y<sub>2</sub>) 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 первой программы:** циклический

**Тип алгоритма второй программы:** циклический

**Тип алгоритма третьей программы:** циклический

### Текст первой программы:

```
# 1. Дан список A размера N. Найти минимальный элемент из его элементов с
# четными
# номерами: A2, A4, A6, .... .
import random
A = []
A1 = []
N = input("Введите размер списка: ")
while type(N) != int:
    try:
        N = int(N)
    except ValueError:
        print('Введите целое число!')
        N = input("Введите размер списка: ")
F = N
while F > 0:
    F -= 1
    A.append(random.randint(-100, 100))
```

```

index = 1
while index < N:
    if index % 2 == 0:
        A1.append(A[index])
    index += 1
print('Первоначальный список', A)
A1.sort(key=None, reverse=False)
element = A1[0]
print('Список с четными номерами элементов', A1)
print('Минимальный элемент с четным номером', element)

```

**Протокол:** Введите размер списка: 5

Первоначальный список [-50, -39, 18, 0, -69]  
 Список с четными номерами элементов [-69, 18]  
 Минимальный элемент с четным номером -69

**Текст второй программы:**

```

# Дан целочисленный список A размера N. Переписать в новый целочисленный
# список B все четные числа из исходного списка (в том же порядке) и вывес
ти размер
# полученного список B и его содержимое..
import random
A = []
B = []
N = input("Введите размер списка: ")
while type(N) != int:
    try:
        N = int(N)
    except ValueError:
        print('Введите целое число!')
        N = input("Введите размер списка: ")
F = N
while F > 0:
    F -= 1
    A.append(random.randint(-100, 100))
for element in A:
    if element % 2 == 0:
        B.append(element)
    else: continue
count = 0
for element in B:
    count += 1
print('Первоначальный список', A)
print('Список четных чисел', B)
print('Количество элементов в списке', count)

```

**Протокол:**

Введите размер списка: 12

Первоначальный список [80, 7, -37, 47, 33, 13, -49, 85, -63, 87, -17, -29]  
Список четных чисел [80]  
Количество элементов в списке 1

### Текст третьей программы:

```
# Дано множество A из N точек (N > 2, точки заданы своими координатами x,
y). Найти
# наибольший 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 (x1, y1) и (x2, y2) вычисляется по формуле:
#  $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$ 
# Для хранения данных о каждом наборе точек следует использовать по два списка: первый
# список для хранения абсцисс, второй – для хранения ординат.
import random
from math import sqrt, ceil
def formula(C1,D1,C2,D2):
    R = sqrt(((C2-C1)**2)+((D2-D1)**2))
    R = ceil(R)
    return R
def peri(chislo1,chislo2,chislo3):
    sum = chislo1+chislo2+chislo3
    return sum
A = [] # список для точек
PER = [] # список для периметров
X = [] # список для координат x
Y = [] # список для координат y
N = 4
print('В множестве A количество точек:', N)
F = N
names = 1
while F > 0:
    F -= 1
    A.append(names)
    names += 1
print('Точки в множестве:', A)
count = N
while count > 0:
    count -= 1
    X.append(random.randint(-10, 10))
    Y.append(random.randint(-10, 10))
print('Координаты по X:', X)
print('Координаты по Y:', Y)
count = 0
a = formula(X[0],Y[0],X[1],Y[1])
b = formula(X[1], Y[1], X[2], Y[2])
c = formula(X[2],Y[2],X[0],Y[0])
PER.append(peri(a,b,c))
```

```
a = formula(X[0],Y[0],X[1],Y[1])
b = formula(X[1], Y[1], X[3], Y[3])
c = formula(X[3],Y[3],X[0],Y[0])
PER.append(peri(a,b,c))
a = formula(X[1],Y[1],X[2],Y[2])
b = formula(X[2], Y[2], X[3], Y[3])
c = formula(X[3],Y[3],X[1],Y[1])
PER.append(peri(a,b,c))
a = formula(X[2],Y[2],X[3],Y[3])
b = formula(X[3], Y[3], X[0], Y[0])
c = formula(X[0],Y[0],X[2],Y[2])
PER.append(peri(a,b,c))
PER.sort(key=None, reverse=True)
print('Список периметров:', PER)
print('Наибольший периметр', PER[0])
```

**Протокол:**

В множестве A количество точек: 4  
Точки в множестве: [1, 2, 3, 4]  
Координаты по X: [-4, -5, 3, -3]  
Координаты по Y: [-4, 3, -6, 0]  
Список периметров: [29, 26, 22, 17]  
Наибольший периметр 29

**Вывод:**

Я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции while, for, def

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовый код выложен на GitHub