

## 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 размера  $N$  и целые числа  $K$  и  $L$  ( $1 < K < L < N$ ). Найти сумму всех элементов списка, кроме элементов с номерами от  $K$  до  $L$  включительно.

**Текст программы:**

*# Вариант 21*

*# Дан список размера  $N$  и целые числа  $K$  и  $L$  ( $1 < K < L < N$ ).*

*# Найти сумму всех элементов списка, кроме элементов с номерами от  $K$  до  $L$  включительно.*

```
import random
```

```
list1 = [ ]
```

```
N = int(input("Введите количество элементов списка: "))
```

```
for i in range(N):                                # Находим случайный список размера N
```

```
    list1.append(random.randint(1, N))
```

```
    list1.sort()
```

```
print("Получившийся случайный список: ", list1)
```

```
K = int(input("Введите номер элемента K: "))
```

```
L = int(input("Введите номер элемента L: "))
```

```
sum1 = 0
```

```
for i in range(len(list1)):                        # Находим сумму необходимых элементов
```

```
    if K - 1 > i or i > L - 1:
```

```
        sum1 += list1[i]
```

```
    else:
```

```
        continue
```

```
print("Сумма элементов списка, кроме элементов с номерами от K до L включительно: ",  
sum1)
```

## Протокол работы программы:

Введите количество элементов списка: 10

Получившийся случайный список: [2, 2, 3, 4, 7, 8, 9, 9, 9, 9]

Введите номер элемента K: 1

Введите номер элемента L: 8

Сумма элементов списка, кроме элементов с номерами от K до L включительно: 18

Process finished with exit code 0

## Постановка задачи:

2) Дан целочисленный список размера N. Если он содержит все числа от 1 до N, то вывести 0; в противном случае вывести номер первого недопустимого элемента.

## Текст программы:

*# Дан целочисленный список размера N.*

*# Если он содержит все числа от 1 до N, то вывести 0;*

*# в противном случае вывести номер первого недопустимого элемента.*

```
import random
```

```
list1 = []
```

```
N = int(input("Введите количество элементов списка: "))
```

```
for i in range(N):
```

*# Находим случайный список размера N*

```
    list1.append(random.randint(1, N))
```

```
    list1.sort()
```

```
print("Получившийся случайный список: ", list1)
```

```
def check_arr(arr):
```

*# Функция по нахождению одинаковых элементов в массиве*

*массиве*

```
    n = len(arr)
```

```
    res = []
```

```
    arr2 = set(arr)
```

```
    for u in range(n - 1):
```

```
        for j in range(u + 1, n):
```

```
            if arr[u] == arr[j]:
```

```
                res.append(j+1)
```

```
                print("Номер первого недопустимого элемента списка: ", end="")
```

```
                return res
```

```
    if len(arr) == len(arr2):
```

```
        print("Список содержит все числа от 1 до N: ", end="")
```

```
        return [0]
```

```
print(check_arr(list1))
```

## Протокол работы программы:

Введите количество элементов списка: 10

Получившийся случайный список: [1, 2, 3, 4, 5, 5, 5, 7, 8, 10]

Номер первого недопустимого элемента списка: [6]

Process finished with exit code 0

## Постановка задачи:

Дано множество  $A$  из  $N$  точек на плоскости и точка  $B$  (точки заданы своими координатами  $x, y$ ). Найти точку из множества  $A$ , наиболее близкую к точке  $B$ . 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.

## Текст программы:

```
# Дано множество A из N точек на плоскости и точка B (точки заданы своими  
# координатами x, y). Найти точку из множества A, наиболее близкую к точке B.  
# Расстояние R между точками с координатами (x1, y1) и (x2, y2) 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.
```

```
import random  
import math
```

```
abc = []  
ordinate = []  
N = int(input("Введите количество точек в множестве A: "))  
for i in range(N): # Находим случайный список размера N  
    abc.append(random.randint(1, N))  
    ordinate.append(random.randint(1, N))  
    abc.sort()  
    ordinate.sort()  
print("Список абсцисс: ", abc)  
print("Список ординат: ", ordinate)  
  
Bx = input("Введите координаты Bx: ")  
By = input("Введите координаты By: ")
```

```

print("Точка В имеет координаты: ", "(" , Bx, ";" , By, ")")

i = 0
check = math.sqrt((int(abc[0]) - int(Bx)) ** 2 + ((int(ordinate[0]) - int(By)) ** 2))
Ax = abc[0]
Ay = ordinate[0]
while i < N:                                # Находим ближайшую точку к точке В
    R = math.sqrt((int(abc[i]) - int(Bx)) ** 2 + ((int(ordinate[i]) - int(By)) ** 2))
    if R <= check:
        check = R
        Ax = abc[i]
        Ay = ordinate[i]
    i += 1
print("Ближайшая к точке В точка А имеет координаты: ", "(" , Ax, ";" , Ay, ")")
print("Расстояние между точками: ", check)

```

### **Протокол работы программы:**

Введите количество точек в множестве A: 10

Список абцисс: [3, 4, 5, 5, 6, 6, 8, 8, 9, 9]

Список ординат: [1, 1, 2, 4, 5, 6, 6, 7, 8, 10]

Введите координаты Bx: 6

Введите координаты By: 8

Точка В имеет координаты: ( 6 ; 8 )

Ближайшая к точке В точка А имеет координаты: ( 6 ; 6 )

Расстояние между точками: 2.0

Process finished with exit code 0

**Вывод:** в ходе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, выработал навыки составления программ со списками в IDE PyCharm Community. Выполнены разработка, отладка, тестирование, оптимизация программного кода. Код выложен на GitHub.

