

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 задач.

- 1) Дан список размером N и целые числа K и L ($1 < K \leq L \leq N$). Найти среднее арифметическое всех элементов списка, кроме элементов с номерами от K до L включительно.
- 2) Даны списки A и B одинакового размера N . Поменять местами их содержимое и вывести вначале элементы преобразованного списка A , а затем – элементы преобразованного списка B .
- 3) Дано множество A из N точек (точки заданы своими координатами x, y). Среди всех точек этого множества, лежащих во второй четверти, найти точку, наиболее удаленную от начала координат. Если таких точек нет, то вывести точку с нулевыми координатами.

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.

Текста программ.

1)

```
# Дан список размера N и целые числа K и L (1 < K <= L <= N) .
# Найти среднее арифметическое всех элементов списка
# кроме элементов с номерами от K до L.
import random

try: # Обработчик исключений
    N = int(input('Введите размер списка: ')) # Ввод размера списка
    print('Введите два целых числа K и L, которые (1 < K <= L <= N)')
    K = int(input('Значение K: '))
    L = int(input('Значение L: '))
    if (K > 1) and (L >= K) and (N >= L):
        a = [] # Вводим пустой список для записи чисел
        b = N # Счетчик для цикла
        while b != 0:
            a.append(random.randint(-10, 10)) # Заполняем список
            значениями в кол-ве N
            b -= 1
        print(f'Список: {a}')
        x = a[:K] # Делаем срез списка до элемента с номером K
        z = a[L + 1:] # Срез списка после элемента с номером L
        x.extend(z) # Увеличиваем список x значениями из списка z
        print(sum(x) / len(x)) # Находим среднее арифметическое
    else:
        print('Введенные числа не подходят')
```

```
except ValueError:
    print('Неверно введенные данные')
```

2)

```
# Даны списки A и B одинакового размера N.
# Поменять местами их содержимое и вывести вначале элементы
# преобразованного списка A
# а затем элементы преобразованного списка B
import random

try: # Обработчик исключений
    N = int(input('Введите размер списков A и B (целое число): ')) #
    Ввод размера списков
    A = [] # Ввод пустого списка A
    B = [] # Ввод пустого списка B
    c = N # Счетчик для цикла
    while c != 0:
        A.append(random.randint(-10, 10)) # Ввод значений для списка A
        B.append(random.randint(-10, 10)) # Ввод значений для списка B
        c -= 1
    A1 = A # Сохранение изначального списка A
    B1 = B # Сохранение изначального списка B
    A = B1 # Меняем содержимое списка A на элементы списка B
    B = A1 # Меняем содержимое списка B на элементы списка A
    print(f'Изначальный список A: {A1}\nИзначальный список B:
{B1}\nРезультат: ')
    print(*A, end=' ')
    print(*B)
except TypeError:
    print('Введите целое число')
```

3)

```
# Дано множество A из точек N (точки заданы своими координатами x, y).
# Среди всех точек этого множества, лежащих во второй четверти, найти
# точку
# наиболее удаленную от начала координат.
# Если таких точек нет, то вывести точку с нулевыми координатами.
from math import *
import random
```

```
a = [] # Пустой список для значений расстояния
maxi_X = 0 # Стартовая координата самой удаленной точки (x)
maxi_Y = 0 # Стартовая координата самой удаленной точки (y)
```

```
def cor(x1, y1, x2, y2):
    R = sqrt(((x2 - x1) ** 2) + ((y2 - y1) ** 2))
    a.append(R)
```

```
x2 = 0 # Абцисса начала координат
y2 = 0 # Ордината начала координат
maxi = 0 # Стартовое максимальное значение расстояния
try:
    N = int(input('Введите количество точек: '))
    if N >= 0:
        X = [] # Начальный список абцисс
        Y = [] # Начальный список ординат
        X1 = [] # Список подходящих абцисс точек
```

```

Y1 = [] # Список подходящих ординат точек
b = N # Счетчик для цикла
while b != 0:
    X.append(random.randint(-10, 10))
    Y.append(random.randint(-10, 10))
    b -= 1
for i in range(0, len(X)):
    if (X[i] <= 0) and (Y[i] >= 0): # Проверка условия, что
        точка находится во 2 четверти
        X1.append(X[i])
        Y1.append(Y[i])
print(f'Абциссы точек: {X}\nОрдинаты: {Y}')
print(f'Подходящие абциссы: {X1}\nОрдинаты: {Y1}')
if len(X1) != 0: # Проверяем есть ли подходящие точки
    for j in range(0, len(X1)):
        cor(X1[j], Y1[j], x2, y2)
    for l in range(0, len(a)):
        if maxi < a[l]: # Находим максимальное значение
            расстояния
            maxi = a[l]
            maxi_X = X1[l] # Записываем координаты самых
            удаленных точек
            maxi_Y = Y1[l]
    else:
        continue
    print(f'Наиболее удалённые от начала координат: ({maxi_X},
{maxi_Y})')
else:
    print(f'Наиболее удалённые от начала координат: ({maxi_X},
{maxi_Y})')
else:
    print('Введите целое число')
except ValueError:
    print('Введите целое число')

```

Протокол работ программ.

1)

Введите размер списка: 7

Введите два целых числа K и L, которые ($1 < K \leq L \leq N$)

Значение K: 2

Значение L: 4

Список: [2, -1, -8, -3, 5, 0, 6]

1.75

Process finished with exit code 0

2)

Введите размер списков A и B (целое число): 7

Изначальный список A: [-7, -7, -6, 5, 9, 0, 8]

Изначальный список В: [-6, -10, -3, 9, 4, -3, 5]

Результат:

-6 -10 -3 9 4 -3 5 -7 -7 -6 5 9 0 8

Process finished with exit code 0

3)

Введите количество точек: 7

Абциссы точек: [-6, 4, -9, -6, 6, -2, -7]

Ординаты: [1, -1, 9, 10, 4, -1, 9]

Подходящие абциссы: [-6, -9, -6, -7]

Ординаты: [1, 9, 10, 9]

Наиболее удалённые от начала координат: (-9, 9)

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ со списками в IDE PyCharm Community.

Выполнена разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHubю