

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 A размера N. Найти минимальный элемент из его элементов с четными номерами: A₂, A₄, A₆, ...

Тип алгоритма: Циклический

Текст программы:

```
1  # Дан список A размера N. Найти минимальный элемент из его элементов с четными
2  # номерами: A2, A4, A6, ...
3  import random
4
5  n = input('Введите длину списка: ')
6  while type(n) != int:
7      try:
8          n = int(n)
9      except ValueError:
10         print('Вы ввели некорректное число!')
11         n = input('Введите длину списка: ')
12  a = [random.randint(1, 99) for _ in range(n)]
13
14  print("Сгенерированный список A:", a)
15
16  even_indexed_elements = a[1::2]
17  if even_indexed_elements:
18      min_element = min(even_indexed_elements)
19      print("Минимальный элемент с четными индексами:", min_element)
20  else:
21      print("Нет элементов с четными индексами.")
22
```

Протокол программы:

Введите длину списка: 4

Сгенерированный список A: [46, 94, 37, 48]

Минимальный элемент с четными индексами: 48

Постановка задачи:

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.

Тип алгоритма: Циклический

Текст программы:

```
1 # Дан целочисленный список A размера N. Переписать в новый целочисленный
2 # список B все четные числа из исходного списка (в том же порядке) и вывести размер
3 # полученного списка B и его содержимое.
4 import random
5
6 n = input('Введите длину списка: ')
7 while type(n) != int:
8     try:
9         n = int(n)
10    except ValueError:
11        print('Вы ввели некорректное число!')
12        n = input('Введите длину списка: ')
13 a = [random.randint(-100, 100) for _ in range(n)]
14
15 b = [x for x in a if x % 2 == 0]
16
17 print("Размер исходного списка A:", a)
18 print("Размер списка B:", len(b))
19 print("Содержимое списка B:", b)
20
```

Протокол программы:

Введите длину списка: 7

Размер исходного списка A: [56, 50, -33, -75, -36, 62, -43]

Размер списка B: 4

Содержимое списка B: [56, 50, -36, 62]

Постановка программы:

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_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: Циклический

Текст программы:

```
4 # Расстояние R между точками с координатами (x1, y1) и (x2, y2) вычисляется по формуле:  $R = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$ .
5 # Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй – для хранения ординат.
6 import math
7 from itertools import combinations
8
9 def distance(x1, y1, x2, y2):
10     return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
11
12 N = int(input("Введите количество точек (N > 2): "))
13 if N <= 2:
14     print("Количество точек должно быть больше 2!")
15     exit()
16
17 x = []
18 y = []
19
20 for i in range(N):
21     x_coord = float(input(f"Введите x для точки {i + 1}: "))
22     y_coord = float(input(f"Введите y для точки {i + 1}: "))
23     x.append(x_coord)
24     y.append(y_coord)
25
26 max_perimeter = 0
27 best_triangle = ()
28
29 for (i, j, k) in combinations(range(N), 3):
30
31     side1 = distance(x[i], y[i], x[j], y[j])
32     side2 = distance(x[j], y[j], x[k], y[k])
33     side3 = distance(x[k], y[k], x[i], y[i])
34
35     perimeter = side1 + side2 + side3
36
37
38     if perimeter > max_perimeter:
39         max_perimeter = perimeter
40         best_triangle = (i, j, k)
41
42
43 if best_triangle:
44     print(f"Наибольший периметр треугольника: {max_perimeter:.2f}")
45     print("Точки, образующие треугольник:")
46     print(f"1: ({x[best_triangle[0]]}, {y[best_triangle[0]]})")
47     print(f"2: ({x[best_triangle[1]]}, {y[best_triangle[1]]})")
48     print(f"3: ({x[best_triangle[2]]}, {y[best_triangle[2]]})")
49 else:
50     print("Нет треугольников, которые можно сформировать из предоставленных точек.")
51
```

Протокол программы:

Введите количество точек (N > 2): 3

Введите x для точки 1: 3

Введите y для точки 1: 5

Введите x для точки 2: 7

Введите y для точки 2: 8

Введите x для точки 3: 4

Введите y для точки 3: 2

Наибольший периметр треугольника: 14.87

Точки, образующие треугольник:

1: (3.0, 5.0)

2: (7.0, 8.0)

3: (4.0, 2.0)

Вывод: В процессе работы я закрепил полученные ранее навыки, приобрел новые навыки в использование циклов и функций научился создавать программы с использованием циклов и функций, приобрел навыки составления программ со списками в IDE PyCharm Community.