

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:

```
C:\Users\I\Downloads> python 12_6.py ...
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 список: первый список для хранения абсцисс, второй — для хранения ординат.

Тип алгоритма: Циклический

Текст программы:

```
1 # Дано множество A из N точек (N > 2, точки заданы своими координатами x, y).
2 # Найти наибольший периметр треугольника, вершины которого принадлежат различным точкам множества A.
3 # и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества A).
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
8 def distance(x1, y1, x2, y2):
9     return math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
10
11 def perimeter(x1, y1, x2, y2, x3, y3):
12     return distance(x1, y1, x2, y2) + distance(x2, y2, x3, y3) + distance(x3, y3, x1, y1)
13
14 def largest_perimeter(points):
15     n = len(points)
16     max_perim = 0
17     best_triangle = None
18
19     for i in range(n):
20         for j in range(i + 1, n):
21             for k in range(j + 1, n):
22                 x1, y1 = points[i]
23                 x2, y2 = points[j]
24                 x3, y3 = points[k]
25                 perim = perimeter(x1, y1, x2, y2, x3, y3)
26
27                 if perim > max_perim:
28                     max_perim = perim
29                     best_triangle = (points[i], points[j], points[k])
30
31     return max_perim, best_triangle
32
33 if __name__ == "__main__":
34     x_coords = [8, 4, 6, 9]
35     y_coords = [0, 1, 2, 3]
36
37     points = list(zip(x_coords, y_coords))
38     max_perimeter, triangle_points = largest_perimeter(points)
39
40     print("Наибольший периметр:", max_perimeter)
41     print("Точки треугольника с наибольшим периметром:", triangle_points)
```

Протокол программы:

Наибольший периметр: 12.670548092920544

Точки треугольника с наибольшим периметром: ((8, 0), (4, 1), (9, 3))

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.