

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) Разработать программу, выводящую на экран минимальный элемент из элементов с четными номерами: A₂, A₄, A₆, ... со списка A размером N.

Текст программы:

```
# Вариант 27. Дан список A размера N. Найти минимальный элемент из его элементов с четными
# номерами: A2, A4, A6, ...

1 usage new *
def min_index_element(arr):
    min_element = arr[1]
    for i in range(3, len(arr), 2):
        if arr[i] < min_element:
            min_element = arr[i]
    return min_element

A = [5, 3, 8, 2, 9, 6, 4,]
print(A)
result = min_index_element(A)
print("Минимальный элемент с чётным индексом:", result)
```

Протокол программы:

[5, 3, 8, 2, 9, 6, 4]

Минимальный элемент с чётным индексом: 2

Process finished with exit code 0

Тип алгоритма: циклический, ветвление

- 2) Разработать программу, которая переписывает в новый целочисленный список В все четные числа из исходного списка А с размером N (в том же порядке) и выводит размер полученного списка В и его содержимое.

Текст программы:

```

# Дан целочисленный список А размера N. Переписать в новый целочисленный
# список В все четные числа из исходного списка (в том же порядке) и вывести размер
# полученного списка В и его содержимое.

1 usage new *
2
3 def extract_numbers(arr):
4     B = [num for num in arr if num % 2 == 0]
5     print("Размер полученного списка В:", len(B))
6     print("Содержимое списка В:", B)
7
8 A = [3, 8, 1, 7, 2, 9, 4, 6]
9 print(A)
10
11 extract_numbers(A)
```

Протокол программы:

[3, 8, 1, 7, 2, 9, 4, 6]

Размер полученного списка В: 4

Содержимое списка В: [8, 2, 4, 6]

Process finished with exit code 0

Тип алгоритма: циклический, ветвление

- 3) Разработать программу, которая найдёт наибольший периметр треугольника, вершины которого принадлежат различным точкам множества А, и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества А). 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 список: первый список для хранения абсцисс, второй — для хранения ординат.

Текст программы:

```
import itertools
3 usages new *
def distance(x1, y1, x2, y2):
    return ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5 # Расстояние между точками по формуле
1 usage new *
def find_perimeter(A):
    max_perimeter = 0
    max_triangle = []

    for i, j, k in itertools.combinations(range(len(A)), r: 3):
        a = distance(A[i][0], A[i][1], A[j][0], A[j][1]) # Расстояние между точками i и j
        b = distance(A[i][0], A[i][1], A[k][0], A[k][1]) # Расстояние между точками i и k
        c = distance(A[j][0], A[j][1], A[k][0], A[k][1]) # Расстояние между точками j и k

        # Вычисляем периметр треугольника
        perimeter = a + b + c

        if perimeter > max_perimeter:
            max_perimeter = perimeter
            max_triangle = [A[i], A[j], A[k]]

    return max_perimeter, max_triangle

A = [(0, 1), (1, 3), (4, 1), (2, 3)]
print(A)

result_perimeter, result_triangle = find_perimeter(A)

print("Наибольший периметр треугольника:", result_perimeter)
print("Вершины треугольника:", result_triangle)
meter()
```

Протокол программы:

[(0, 1), (1, 3), (4, 1), (2, 3)]

Наибольший периметр треугольника: 9.84161925296378

Вершины треугольника: [(0, 1), (1, 3), (4, 1)]

Process finished with exit code 0

Тип алгоритма: циклический, ветвление

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