

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 ненулевых целых чисел размера N. Проверить, чередуются ли в нем положительные и отрицательные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность.

Текст программы:

```
"""Дан список ненулевых целых чисел размера N.
Проверить, чередуются ли в нем положительные и отрицательные числа.
Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого
элемента, нарушающего закономерность."""

def check(lst):
    for i in range(len(lst) - 1):
        if lst[i] * lst[i + 1] > 0: # проверяем, одного ли знака соседние числа
            return i + 1 # возвращаем порядковый номер (индекс)
    return 0 # числа чередуются

A = [1, -2, 3, -4, 5, -6]
print("Список A:", A)
print(check(A))
```

Протокол работы программы:

Список A: [1, -2, 3, -4, 5, -6]

0

Process finished with exit code 0

Постановка задачи № 2:

Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести размер полученного списка B и его содержимое. Условный оператор не использовать.

Текст программы:

```

"""Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный
список B все элементы с нечетными порядковыми номерами (1,3,...) и вывести
размер полученного списка B и его содержимое. Условный оператор не
использовать."""
import random
N = random.randint(1, 14) # Создание случайной длины списка в диапазоне
A = [random.randint(-100, 100) for i in range(N)] # Создание случайных
элементов списка в диапазоне
print("Список A:", A)
B = A[::2] # Занесение нечетных элементов в новый список
print("Размер списка B:", len(B))
print("Список B:", B)

```

Протокол работы программы:

Список A: [4, 1, 80, -80, 69, -90, -70, -63, 74, 16, -65]

Размер списка B: 6

Список B: [4, 80, 69, -70, 74, -65]

Process finished with exit code 0

Постановка задачи № 3:

Дано множество A из N точек ($N > 2$, точки заданы своими координатами x, y). Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна, и саму эту сумму. 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 ( $N > 2$ , точки заданы своими координатами  $x, y$ ).
Найти такую точку из данного множества, сумма расстояний от которой до остальных
его точек минимальна, и саму эту сумму.
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.
"""
import math

x_cords = [5, 8, 16]
y_cords = [3, 4, -9]
distance_sum = []
if len(x_cords) > 2 and len(y_cords) > 2: # Проверка, что есть минимум 3 точки
    for x, y in zip(x_cords, y_cords):
        distances = [] # Список для хранения расстояний от текущей точки до
        # всех остальных
        c_point = (x, y)
        for x1, y1 in zip(x_cords, y_cords):
            point = (x1, y1)

```

```

        if point != c_point: # Проверка, что это не текущая точка
            distance = math.sqrt((x1-x)**2+(y1-y)**2)
            distances.append(distance)
        distances_sum = sum(distances)
        distance_sum.append(distances_sum)
    min_sum = min(distance_sum)
    min_sum_point_index = distance_sum.index(min_sum) # Индекс точки с
минимальной суммой расстояний
    min_sum_point = (x_cords[min_sum_point_index], y_cords[min_sum_point_index])
    print("Точка минимальной суммы расстояний:", min_sum_point)
    print("Минимальная сумма расстояний:", min_sum)
else:
    print("Вы ввели меньше трёх точек")

```

Протокол работы программы:

Точка минимальной суммы расстояний: (8, 4)

Минимальная сумма расстояний: 18.426615182642127

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции цикл for, if, else.

Выполнены разработка кода, отладка, тестирование, оптимизация, программного кода. Готовые программные коды выложены на GitHub.