

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.

Тип алгоритма: Циклический.

Текст программы:

```
'''
Вар 26
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.
'''

ListAppend = []

def introduction_int(Content): # Добавление числа с проверкой
    x = input(Content)
    while type(x) != int:
        try:
            x = int(x)
            return x
        except ValueError:
            print('Вы ввели число не правильно')
            x = input(Content)

def alternation(List): # Узнаём идёт чередование или нет
    for i in range(1, element):
        if (List[i] > 0 and List[i - 1] > 0) or (List[i] < 0 and List[i
- 1] < 0):
            return i + 1
    return 0
```

```
element = introduction_int('Введите сколько будет элемнтов списка: ')

for i in range(element):
    ListAppend.append(introduction_int(f'Введите {i + 1} элемент списка
'))

print(alternation(ListAppend))
```

Протокол работы программы:

Введите сколько будет элемнтов списка: 4

Введите 1 элемент списка -1

Введите 2 элемент списка 3

Введите 3 элемент списка -78

Введите 4 элемент списка 5

0

Постановка задачи 2:

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

Тип алгоритма: Циклический.

Текст программы:

```
'''
Вар 26
Дан целочисленный список А размера N (< 15). Переписать в новый
целочисленный
список В все элементы с нечетными порядковыми номерами (1,3,...) и
вывести
размер полученного списка В и его содержимое. Условный оператор не
использовать.

'''
ListAppend = []

def introduction_int(Content): # Добавление числа с проверкой
```

```
x = input(Content)
while type(x) != int:
    try:
        x = int(x)
        return x
    except ValueError:
        print('Вы ввели число не правильно')
        x = input(Content)

while True:
    element = introduction_int('Введите сколько будет элемнтов
списка(<15): ')
    if element < 15:
        break
    else:
        print('Слишком много элемнтов списка')

for i in range(element):
    ListAppend.append(introduction_int(f'Введите {i + 1} элемент списка
'))

b = []

for i in range(0, element, 2):
    b.append(ListAppend[i])

print(b, len(b))
```

Протокол работы программы:

Введите сколько будет элемнтов списка(<15): 5

Введите 1 элемент списка 4

Введите 2 элемент списка 6

Введите 3 элемент списка 7

Введите 4 элемент списка 4

Введите 5 элемент списка 3

[4, 7, 3] 3

Постановка задачи 3:

Дано множество А 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.

Тип алгоритма: Циклический.

Текст программы:

```
...
Вар 26
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:
 $R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 
 $R = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ 

Для хранения данных о каждом наборе точек следует использовать по два
списка: первый
список для хранения абсцисс, второй — для хранения ординат.

...
import math

def introduction_int(Content): # Добавление числа с проверкой
    x = input(Content)
    while type(x) != int:
        try:
            x = int(x)
            return x
        except ValueError:
            print('Вы ввели число не правильно')
            x = input(Content)

def read_points():
```

```
N = introduction_int("Введите количество точек (N > 2): ")
if N <= 2:
    raise ValueError("Количество точек должно быть больше 2.")

x_coords = []
y_coords = []

for i in range(N):
    x, y = map(float, input(f"Введите координаты точки {i + 1} (x
y): ").split())
    x_coords.append(x)
    y_coords.append(y)

return x_coords, y_coords

def calculate_min_distance_point(x_coords, y_coords):
    N = len(x_coords)
    min_distance_sum = float('inf')
    best_point_index = -1

    for i in range(N):
        distance_sum = 0
        for j in range(N):
            if i != j: # Не учитываем расстояние до самой себя
                distance_sum += math.sqrt((x_coords[j] - x_coords[i]) **
2 + (y_coords[j] - y_coords[i]) ** 2)

        if distance_sum < min_distance_sum:
            min_distance_sum = distance_sum
            best_point_index = i

    return best_point_index, min_distance_sum

x_coords, y_coords = read_points()
best_point_index, min_distance_sum =
calculate_min_distance_point(x_coords, y_coords)

print(f"Точка с индексом {best_point_index + 1} (координаты:
({x_coords[best_point_index]}, {y_coords[best_point_index]})) имеет
минимальную сумму расстояний: {min_distance_sum}")
```

Протокол работы программы:

Введите количество точек (N > 2): 4

Введите координаты точки 1 (x y): 1 4

Введите координаты точки 2 (x y): 3 5

Введите координаты точки 3 (x y): 5 7

Введите координаты точки 4 (x y): 4.7 5.6

Точка с индексом 2 (координаты: (3.0, 5.0)) имеет минимальную сумму расстояний: 6.867270739977974

Вывод: Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ со списками в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода

Готовые программные коды выложены на GitHub.