Практическое занятие №6

Тема: составление программ со списками в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ со списками в IDE PyCharm Community.

Задача 1

Дан список A размера N. Найти максимальный элемент из его элементов с нечетными номерами: A1, A3, A5, ...

Текст программы:

```
# Дан список A размера N. Найти максимальный элемент из его элементов с
# нечетными номерами: A1, A3, A5, ... .

import random

N = int(input("Введите размер списка: "))
while type(N) != int:    # обработка исключений
    try:
        N = int(N)
        if N < 0:
            print("Ошибка: введите положительное число")
            N = input("Введите размер списка: ")
    except ValueError:
        print("Неправильно ввели!")
        N = input("Введите размер списка: ")

ListAppend = []

# Заполнение списка случайными числами от -100 до 100
for t in range(N):
    ListAppend.append(random.randint(-100, 100))

# Начальная инициализация максимального значения
max_value = -101  # Значение меньше минимального возможного

# Поиск максимального элемента с нечетными индексами
for i in range(1, len(ListAppend), 2):  # Начинаем с 1, шаг 2 (нечетные
индексы)
    if ListAppend[i] > max_value:
        max_value = ListAppend[i]  # Обновляем максимальное значение

# Вывод массива и максимального элемента
print("Сгенерированный массив:", ListAppend)
if max_value != -101:  # Проверяем, были ли найдены элементы с нечетными
индексами
    print("Максимальный элемент с нечетными индексами:", max_value)
else:
    print("Нет элементов с нечетными индексами.")
```

Протокол программы:

Введите размер списка: 5

Сгенерированный массив: [74, -12, 72, -3, 23]

Максимальный элемент с нечетными индексами: -3

Задача 2

Дан целочисленный список A размера N (< 15). Переписать в новый целочисленный список B все элементы с порядковыми номерами, кратными трем (3, 6, ...), 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 порядковыми номерами, кратными трем (3, 6, ...), и
вывести
 # размер полученного списка B и его содержимое. Условный оператор не
 # использовать.

# Вводим размер списка
N = input("Введите размер списка: ")
while type(N) != int:   # обработка исключений
    try:
        N = int(N)
        if N > 15:
            print("Ошибка: размер списка должен быть не более 15")
            N = input("Введите размер списка: ")
    except ValueError:
        print("Неправильно ввели!")
        N = input("Введите размер списка: ")

# Вводим элементы списка A
A = []
for i in range(N):
    element = input(f'Введите элемент {i + 1}: ')
    A.append(element)

# Создаем список B, выбирая элементы с индексами, кратными 3
B = A[2::3]  # Начинаем с индекса 2 (т. е. 3-й элемент) и берём каждый 3-й
элемент

# Выводим размер и содержимое списка B
print('Размер списка B:', len(B))
print('Содержимое списка B:', B)
```

Протокол программы:

Введите размер списка: 6

Введите элемент 1: 1

Введите элемент 2: 3

Введите элемент 3: 4

Введите элемент 4: 5

Введите элемент 5: 6

Введите элемент 6: 7

Размер списка B: 2

Содержимое списка B: ['4', '7']

Задача 2

Дано множество A из N точек (N > 2, точки заданы своими координатами x, y). Найти наименьший 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 (x1, y1) и (x2, y2) вычисляется по формуле: R = √(x2 – x1)^2 + (y2 – y1)^2. Для хранения данных о каждом наборе точек следует использовать по два списка: первый список для хранения абсцисс, второй — для хранения ординат.

Текст программы:

```
import math

def distance(x1, y1, x2, y2):
    # Вычисляем расстояние
    distance = math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
    return distance

N = input("Введите количество точек (больше 2):  ")
while type(N) != int:   # обработка исключений
    try:
        N = int(N)
        if N < 3:
            print("Ошибка: количество точек должно быть больше 2.")
            N = input("Введите количество точек (больше 2):  ")
    except ValueError:
        print("Неправильно ввели!")
        N = input("Введите количество точек (больше 2):  ")

# Ввод координат точек
x_coords = []
y_coords = []
for i in range(N):
    x_element = input(f'Введите координату x для точки {i + 1}: ')
    while type(x_element) != int:  # обработка исключений
        try:
            x_element = int(x_element)
        except ValueError:
            print("Неправильно ввели!")
            x_element = input(f'Введите координату x для точки {i + 1}: ')
    y_element = input(f'Введите координату y для точки {i + 1}: ')
    while type(y_element) != int:  # обработка исключений
        try:
            y_element = int(y_element)
        except ValueError:
            print("Неправильно ввели!")
            y_element = input(f'Введите координату y для точки {i + 1}: ')
    x_coords.append(x_element)
    y_coords.append(y_element)


perimeters = []

# Перебор комбинаций точек
for i in range(N):
    for j in range(i + 1, N):  # j должен быть больше, чем i
        for k in range(j + 1, N):  # k должен быть больше, чем j
            # Точки треугольника
            point1 = (x_coords[i], y_coords[i])
            point2 = (x_coords[j], y_coords[j])
            point3 = (x_coords[k], y_coords[k])

            side_a = distance(x_coords[i], y_coords[i], x_coords[j],
```

```
y_coords[j])
            side_b = distance(x_coords[j], y_coords[j], x_coords[k],
y_coords[k])
            side_c = distance(x_coords[k], y_coords[k], x_coords[i],
y_coords[i])

            # Здесь вы можете делать что-то с полученными точками
            print(f"Точки треугольника: {point1}, {point2}, {point3}")

            # Вычисление периметра
            perimeter = side_a + side_b + side_c
            print("Периметр: ", perimeter)
            perimeters.append(perimeter)

# Нахождение минимального периметра
min_perimeter = min(perimeters)
print(f"Минимальный периметр: {min_perimeter}")
```

Протокол программы:

Введите количество точек (больше 2): 4

Введите координату x для точки 1: 2

Введите координату y для точки 1: 3

Введите координату x для точки 2: 4

Введите координату y для точки 2: 1

Введите координату x для точки 3: 5

Введите координату y для точки 3: 3

Введите координату x для точки 4: 6

Введите координату y для точки 4: 2

Точки треугольника: (2, 3), (4, 1), (5, 3)

Периметр: 8.06449510224598

Точки треугольника: (2, 3), (4, 1), (6, 2)

Периметр: 9.18760072786364

Точки треугольника: (2, 3), (5, 3), (6, 2)

Периметр: 8.537319187990756

Точки треугольника: (4, 1), (5, 3), (6, 2)

Периметр: 5.8863495173726745

Минимальный периметр: 5.8863495173726745

Вывод: я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составление программ со списками в IDE PyCharm Community