Студентка группы ИС-25 Скобелина В.В.

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..

Постановка задачи:

1. Дан список A размера N. Найти максимальный элемент из его элементов с нечетными номерами: A1, A3, A5 ...
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.
3. 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:

   $R = \sqrt{(x2 - x1)2 + (y2 - y1)2}$.

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

Тип алгоритма:

1. Ветвление с функцией, в составе которой цикл
2. Ветвление с функцией, в составе которой цикл
3. Ветвление с циклом функцией, в составе которой цикл

Текст программы:

1.
```
#Дан список A размера N. Найти максимальный элемент
#из его элементов с нечетными номерами: A1, A3, A5, ... .
1 usage  victoriaskobelina
def find_max_odd_elements(A):
    max_element = float('-inf')  # Инициализируем максимальный элемент как отрицательную бесконечность
    for i in range(len(A)):
        try:
            if i % 2 != 0 and A[i] > max_element:  # Проверяем нечетный индекс и обновляем максимальный элемент
                max_element = A[i]
        except IndexError:
            print("Индекс выходит за границы списка.")
            return None
    return max_element
A = [1, 5, 2, 3, 8, 4, 7, 6, 4]
result = find_max_odd_elements(A)
if result is not None:
    print("Максимальный элемент с нечетными индексами:", result)
```

2.

```python
#и вывести размер полученного списка B и его содержимое. Условный оператор не использовать.
1 usage  victoriaskobelina
def create_list_B(A):
    B = []  # Создаем пустой список B
    try:
        for i in range(2, len(A), 3):
            B.append(A[i])  # Добавляем элементы с порядковыми номерами, кратными трем, в список B
    except IndexError:
        print("Индекс выходит за границы списка.")
        return None
    return B
try:
    A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
    B = create_list_B(A)

    if B is not None:
        print("Размер списка B:", len(B))
        print("Содержимое списка B:", B)
except ValueError:
    print("Введите целочисленные значения для списка A.")
```

3.

```python
#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.
import math
1 usage  victoriaskobelina
def find_smallest_perimeter(A):
    min_perimeter = float('inf')  # Инициализируем минимальный периметр как положительную бесконечность
    smallest_triangle = []  # Список для хранения точек треугольника с наименьшим периметром
    for i in range(len(A)):
        for j in range(i+1, len(A)):
            for k in range(j+1, len(A)):
                # Вычисляем расстояния между точками
                distance1 = math.sqrt((A[j][0] - A[i][0])**2 + (A[j][1] - A[i][1])**2)
                distance2 = math.sqrt((A[k][0] - A[j][0])**2 + (A[k][1] - A[j][1])**2)
                distance3 = math.sqrt((A[i][0] - A[k][0])**2 + (A[i][1] - A[k][1])**2)
                # Вычисляем периметр треугольника
                perimeter = distance1 + distance2 + distance3
                # Если текущий периметр меньше минимального, обновляем минимальный периметр и точки треугольника
                if perimeter < min_perimeter:
                    min_perimeter = perimeter
                    smallest_triangle = [A[i], A[j], A[k]]
    return min_perimeter, smallest_triangle
N = int(input("Введите количество точек в множестве A: "))
if N > 2:
    A = []
    for _ in range(N):
        x = int(input("Введите абсциссу точки: "))
        y = int(input("Введите ординату точки: "))
        A.append((x, y))
    perimeter, triangle = find_smallest_perimeter(A)
    print("Наименьший периметр треугольника:", perimeter)
    print("Точки треугольника:", triangle)
else:
    print("Количество точек должно быть больше 2")
```

Протокол работы программы:

1. Максимальный элемент с нечетными индексами: 6

   Process finished with exit code 0

   Программа успешно завершена!

2. Размер списка B: 4

   Содержимое списка B: [3, 6, 9, 12]

   Process finished with exit code 0

Студентка группы ИС-25 Скобелина В.В.

Программа успешно завершена!

3. Введите количество точек в множестве A: 5
Введите абсциссу точки: 5
Введите ординату точки: 5
Введите абсциссу точки: 5
Введите ординату точки: 5
Введите абсциссу точки: 5
Введите ординату точки: 5
Введите абсциссу точки: 5
Введите ординату точки: 5
Введите абсциссу точки: 5
Введите ординату точки: 5

Наименьший периметр треугольника: 0.0
Точки треугольника: [(5, 5), (5, 5), (5, 5)]
Process finished with exit code 0
Программа успешно завершена!

Вывод: в процессе выполнения практического занятия были выработаны навыки составления программ со списками в IDE PyCharm Community и закреплены усвоенные знания, понятия, алгоритмы, основные принципы составления программ. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.