Студент группы ИС-21 Дружинин А. А.

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.

1) Постановка задачи.
Дан список A размера N. Найти минимальный элемент из его элементов с четными номерами: A2, A4, A6 …

Тип алгоритма: циклический.

Текст программы:

```
# Дан список A размера N. Найти минимальный элемент из его элементов с четными
номерами: A2, A4, A6, …
from random import randint  # Импортирование модуля randint

n = input('Введите N: ')  # Ввод данных
lst = []
k = 0

while type(n) != int:  # Обработка исключений
    try:
        n = int(n)
    except ValueError:
        print('Неправильный ввод')
        n = input('Введите N: ')

a = [randint(0, 9) for _ in range(n)]  # Генерация списка
print('Исходный список:', a)

for i in a:  # Добавление элементов с четными индексами
    if k % 2 == 0:
        lst.append(i)
    k += 1

print(min(lst))
```

Протокол работы программы:

Введите N: 10
Исходный список: [3, 7, 2, 9, 6, 3, 8, 8, 8, 5]
2

Process finished with exit code 0

2) Постановка задачи.
Дан целочисленный список A размера N. Переписать в новый целочисленный список B все четные числа из исходного списка (в том же порядке) и вывести размер полученного список B и его содержимое.

Тип алгоритма: циклический.

Текст программы:

```
# Дан целочисленный список A размера N. Переписать в новый целочисленный список
B все четные числа из исходного
# списка (в том же порядке) и вывести размер полученного список B и его
содержимое.
from random import randint  # Импортирование модуля randint

n = input('Введите N: ')  # Ввод данных

while type(n) != int:  # Обработка исключений
    try:
        n = int(n)
    except ValueError:
        print('Неправильный ввод')
        n = input('Введите N: ')

A = [randint(0, 9) for _ in range(n)]  # Генерация списка
print('Исходный список:', A)
B = []

for i in A:  # Добавление четных чисел в список B
    if i % 2 == 0:
        B.append(i)

print('Размер списка B:', len(B))  # Вывод результата
print('Список B -', B)
```

Протокол работы программы:

Введите N: 10
Исходный список: [9, 0, 2, 2, 8, 5, 3, 8, 4, 1]
Размер списка B: 6
Список B - [0, 2, 2, 8, 8, 4]

Process finished with exit code 0

3) Постановка задачи.
Дано множество A из N точек (N > 2, точки заданы своими координатами x, y). Найти наибольший 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 список: первый список для хранения абсцисс, второй - для хранения ординат.

Тип алгоритма: циклический.

Текст программы:

```
# Дано множество A из N точек (N > 2, точки заданы своими координатами x, y).
Найти наибольший периметр треугольника,
# вершины которого принадлежат различным точкам множества A, и сами эти точки
(точки выводятся в порядке их
# перечисления 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.
from random import randint  # Импортирование модуля randint

n = input('Введите N: ')  # Ввод данных

while type(n) != int:  # Обработка исключений
    try:
        n = int(n)
    except ValueError:
        print('Неправильный ввод')
        n = input('Введите N: ')

current_per = 0
max_per = 0
a, b, c = 0, 0, 0

A = [(randint(0, 9), randint(0, 9)) for _ in range(n)]  # Генерация множества с
координатами
print('Множество:', A)


def length(x, y):  # Функция для вычисления расстояния R
    return ((x[1] - x[0]) ** 2 + (y[1] - y[0]) ** 2) ** 0.5


def per(x, y, z):  # Функция для вычисления периметра
    return length(x, y) + length(y, z) + length(x, z)


for i in range(n - 2):  # Циклы для перебора множества с координатами
    for j in range(i + 1, n - 1):
        for k in range(j + 1, n):
            current_per = per(A[i], A[j], A[k])  # Текущий периметр
            if current_per > max_per:
                max_per = current_per  # Переприсваивание максимального
периметра на текущий
                a, b, c = A[i], A[j], A[k]  # Сохранение координат максимального
периметра

print('Наибольший периметр:', max_per)  # Вывод результатов
print('Координаты:', a, b, c)
```

Протокол работы программы:

Введите N: 5
Множество: [(8, 1), (4, 5), (8, 7), (5, 0), (7, 0)]
Наибольший периметр: 27.10414547069692
Координаты: (8, 1) (5, 0) (7, 0)

Process finished with exit code 0

Студент группы ИС-21 Дружинин А. А.

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