

Практическое занятие № 6

Тема: Составление программ в IDE PyCharm Community.

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

Постановка задачи №1.

Дан целочисленный список размера 10. Вывести вначале все содержащиеся в данном списке чётные числа в порядке возрастания их индексов, а затем - все нечётные числа в порядке убывания их индексов

Текст программы:

```
import random # Вызываем для "random" для дальнейшей генерации 10
случайных чисел

a = [] # Создаём пустой список для случайных 10 значений
b = [] # Создаём пустой список для занесения и вывода в нужной нам
последовательности
for i in range(10): # Делаем цикл в 10 кругов, для 10 чисел
    a.append(random.randint(1, 99)) # Каждый круг добавляем случайное
значение в список(1 до 99, для удобства)
print('Начальный случайный список значений:', a)
for i in a: # Перебираем список a
    if i % 2 == 0: # Если число чётное, то
        b.append(i) # Заносим в список
a = a[::-1] # Переворачиваем наш список, для того чтобы дальнейшие
числа записывались в обратной последовательности
for i in a: # Перебираем список a
    if i % 2 != 0: # Если число нечётное, то
        b.append(i) # Заносим в список
print('Список в нужной последовательности:', b) # Выводим готовый
список в нужной нам последовательности
```

Протокол работы программы:

Начальный случайный список значений: [62, 57, 52, 44, 17, 9, 48, 92, 85, 62]

Список в нужной последовательности: [62, 52, 44, 48, 92, 62, 85, 9, 17, 57]

Process finished with exit code 0

Постановка задачи №2.

Дан список размера N. Найти количество участков, на которых его элементы монотонно убывают.

Текст программы:

```
spisok = input().split(' ') # Вводим список, через пробел
a = 0 # Создаём переменную-счётчик
for i in range(len(spisok) - 2): # Создаём цикл по длине
    if spisok[i] > spisok[i + 1] > spisok[i + 2]: # Если элемент справа
        > и последующий элемент справа >, то
        a += 1 # Записываем +1 к участкам монотонности
print(a) # Выводим итоговое количество участков
```

Протокол работы программы:

6 5 4 6 5 4

2

Process finished with exit code

Постановка задачи №3.

Дано множество A из N точек на плоскости и точка B (точки заданы своими координатами x, y). Найти точку из множества A, наиболее близкую к точке B.

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: список для хранения абсцисс, второй - для хранения ординат.

Текст программы:

```
import math # Вызов математической библиотеки для дальнейших расчётов
from random import choices # Вызов рандома для генерации координат

print('Введите количество точек в множестве A') # Информировать, что
# нужно ввести пользователю
N = int(input()) # Ввод количества координат в множестве A
r = range(100) # Максимальное значение координаты
A = [] # Создаём пустой список для множества координат
B = [] # Создаём пустой список для одной координаты
for i in range(N): # Делаем цикл для создания случайных значений в
    # количестве N
    x, y = choices(r, k=2) # Создание случайной координаты, каждый раз
    A.append((x, y)) # Каждую случайную координату заводим в список в
    # виде кортежа
print('Множество точек A:', A) # Выводим случайное множество точек A
x, y = choices(r, k=2) # Создаём случайную координату
B.append(x) # Добавляем x координату в лист B
B.append(y) # Добавляем y координату в лист B
print('Точка B:', B) # Выводим случайную точку B
# Рассчитываем разницу координат и выводим координату с минимальной
# разницей
print('Самая близкая точка из A к точке B', (min(A, key=lambda point:
    math.hypot(B[1] - point[1], B[0] - point[0]))))
```

Протокол работы программы:

Введите количество точек в множестве A

4

Множество точек A: [(98, 25), (24, 80), (14, 60), (7, 66)]

Точка B: [88, 33]

Самая близкая точка из A к точке B (98, 25)

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.