

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 целочисленный список размера N. Проверить, чередуются ли в нем четные и нечетные числа. Если чередуются, то вывести 0, если нет, то вывести порядковый номер первого элемента, нарушающего закономерность.

Текст программы 1:

```
import random

def check(aboba_num): 1 usage
    for i in range(1, len(aboba_num)):
        # проверка четности текущего и предыдущего чисел
        if (aboba_num[i] % 2) == (aboba_num[i - 1] % 2):
            # вывод индекса при одинаковой четности чисел
            return i + 1
    return 0 # если чередование соблюдено

# список из 5 случайных чисел от 1 до 100
numbers = [random.randint(a: 1, b: 100) for i in range(5)]

# список наших чисел
print("РАНДОМ:", numbers)

# проверка чередования чисел в списке
result = check(numbers)
print(result)
```

Протокол работы программы:

РАНДОМ: [8, 74, 97, 45, 93]
2

Process finished with exit code 0

Постановка задачи 2 Даны два списка А и В одинакового размера N. Сформировать новый список С того же размера, каждый элемент которого равен максимальному из элементов списков А и В.

Текст программы 2:

```
import random

# списки a и b, которые генерируют 5 рандомных чисел от 1 до 100
A = [random.randint(a: 1, b: 100) for i in range(5)]
B = [random.randint(a: 1, b: 100) for il in range(5)]

# список для наших чисел
Yuumi = []

# для нахождения чисел с макс значением
for max_num in range(len(A)):
    max_value = max(A[max_num], B[max_num])
    Yuumi.append(max_value)

# результат!!
print(Yuumi)
```

Протокол работы программы:

[98, 18, 38, 89, 48]

Process finished with exit code 0

Постановка задачи 3 Даны множества А и В, состоящие соответственно из N1 и N2 точек (точки заданы своими координатами x, y). Найти минимальное расстояние между точками этих множеств и сами точки, расположенные на этом расстоянии (вначале выводится точка из множества А, затем точка из множества В). Расстояние R между точками с координатами (x1, y1) и (x2, y2) 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.

Текст программы 3:

```
import random

# функция будет вычислять расстояние между двумя точками
def calculate_distance(x1, y1, x2, y2): 1 usage
    return ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 0.5

# координаты ху для множества А
A_x = [random.randint(a: 1, b: 100) for i in range(3)] # Абсциссы точек А
A_y = [random.randint(a: 1, b: 100) for il in range(3)] # Ординаты точек А

# координаты ху для множества В
B_x = [random.randint(a: 1, b: 100) for ilu in range(3)] # Абсциссы точек В
B_y = [random.randint(a: 1, b: 100) for lumi in range(3)] # Ординаты точек В

# переменные для данных
min_distance = float('inf')
closest_point_A = None
closest_point_B = None

for k in range(len(A_x)):
    for j in range(len(B_x)):
        distance = calculate_distance(A_x[k], A_y[k], B_x[j], B_y[j]) # расстояние между точками А и В

        # если расстояние меньше минимального, минимальное значение и точки обновляются
        if distance < min_distance:
            min_distance = distance
            closest_point_A = (A_x[k], A_y[k]) # Сохраняем точку из А
            closest_point_B = (B_x[j], B_y[j]) # Сохраняем точку из В

# вывод результатов вычислений
print(f"Минимальное расстояние: {min_distance}")
print(f"Точка из А: {closest_point_A}")
print(f"Точка из В: {closest_point_B}")
```

Протокол работы программы:

Минимальное расстояние: 10.770329614269007

Точка из A: (31, 33)

Точка из B: (27, 23)

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции `int`, `def`, `return`, `input`, `print`, `if`, `else`, `elif`, `списки`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.