

Практическая работа 7

Наименование практического занятия: составление программ с функциями в IDE PyCharm Community

Цели практического занятия: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с функциями в IDE PyCharm Community

Постановка задачи:

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:

```
def check_alternating(nums):
    for i in range(1, len(nums)):
        if nums[i] % 2 == nums[i - 1] % 2:
            return i
    return 0

# Пример использования
numbers = [2, 5, 4, 7, 6, 9]
result = check_alternating(numbers)

if result == 0:
    print("Четные и нечетные числа чередуются.")
else:
    print(f"Закономерность нарушается на позиции {result}.")
```

протокол работы программы:

Четные и нечетные числа чередуются.

Закономерность нарушается на позиции 2

Постановка задачи:

Даны два списка A и B одинакового размера N. Сформировать новый список C того же размера, каждый элемент которого равен максимальному из элементов списков A и B.

Текст программы:

```
def max_of_lists(A, B):
    return [max(A[i], B[i]) for i in range(len(A))]

# Пример использования
list_A = [3, 7, 1, 9, 4]
list_B = [5, 2, 8, 6, 10]
```

```
result_list = max_of_lists(list_A, list_B)
print("Новый список C:", result_list)
```

протокол работы программы:

Новый список C: [5, 7, 8, 9, 10]

Постановка задачи:

Дано множество A из N точек (точки заданы своими координатами x, y). Найти пару различных точек этого множества с максимальным расстоянием между ними и само это расстояние (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

Текст программы:

```
import math

def find_min_distance(A, B):
    min_distance = float('inf')
    min_points = None

    for point_a in A:
        for point_b in B:
            distance = math.sqrt((point_b[0] - point_a[0])**2 + (point_b[1] -
point_a[1])**2)
            if distance < min_distance:
                min_distance = distance
                min_points = (point_a, point_b)

    return min_distance, min_points

set_A = [(1, 4), (2, 5), (3, 6)]
set_B = [(7, 10), (8, 11), (9, 12)]

min_distance, min_points = find_min_distance(set_A, set_B)

print(f"Минимальное расстояние: {min_distance}")
print(f"Точки на минимальном расстоянии: {min_points}")
```

протокол работы программы:

Минимальное расстояние: 5.656854249492381

Точки на минимальном расстоянии: ((3, 6), (7, 10))

Вывод: в процессе выполнения практического занятия выработал навыки составления программ
Линейной структуры в IDE PyCharm Community. Готовые программные коды выложены на GitHub