

Практическое занятие № 6

Тема: составление программ со списками в IDE PyCharm Community.

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

Постановка задачи:

1. Дан список A размера N и целое число K ($1 < K < N$). Вывести элементы списка с порядковыми номерами, кратными K: AK, A2*K, A3*K,... . Условный оператор не использовать.
2. Дан список размера N. Найти количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).
3. Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию (при этом A1 перейдет в A2, A2 — в A3, ..., AN-1 — в AN, а исходное значение последнего элемента будет потеряно). Первый элемент полученного списка положить равным 0.

Тип алгоритма: циклическая

Текст программы (6.1):

```
# Дан список A размера N и целое число K (1 < K < N). Вывести элементы списка
с
# порядковыми номерами, кратными K: AK, A2*K, A3*K,... . Условный оператор не
# использовать.
import random
def generate_random_list(size, min_value, max_value):
    return [random.randint(min_value, max_value) for _ in range(size)]
def extract_elements_by_step(lst, step):
    return lst[step-1::step]
if __name__ == "__main__":
    N = 10 # Размер списка
    K = 3  # Шаг для выбора элементов
    min_value = -100
    max_value = 100
    A = generate_random_list(N, min_value, max_value)
    print("Исходный список:", A)
    result = extract_elements_by_step(A, K)
    print("Элементы с индексами, кратными K:", result)
```

Протокол работы программы:

Исходный список: [-54, -7, -49, -26, -48, 89, 73, -35, 59, -48]

Элементы с индексами, кратными K: [-49, 89, 59]

Текст программы (6.2):

```
# Дан список размера N. Найти количество его промежутков монотонности (то
# есть
# участков, на которых его элементы возрастают или убывают).
import random
def generate_random_list(n, min_value=-10, max_value=10):
    return [random.randint(min_value, max_value) for _ in range(n)]
def count_monotonic_segments(lst):
    try:
        if not lst:
            return 0
        current_length = 1
        direction = None
        monotonic_count = 0
        for i in range(len(lst) - 1):
            if lst[i + 1] > lst[i]:
                new_direction = 'increasing'
            elif lst[i + 1] < lst[i]:
                new_direction = 'decreasing'
            else:
                continue

            if direction is None or direction == new_direction:
                current_length += 1
            else:
                monotonic_count += 1
                current_length = 2
                direction = new_direction
        monotonic_count += 1
        return monotonic_count
    except TypeError as e:
        print(f"Произошла ошибка: {e}. Переданный аргумент должен быть
        списком чисел.")
        return None
def main():
    N = 10
    lst = generate_random_list(N)
    print("Список:", lst)
    monotonic_count = count_monotonic_segments(lst)
    if monotonic_count is not None:
        print("Количество участков монотонности:", monotonic_count)
    else:
        print("Не удалось посчитать количество участков монотонности.")
if __name__ == "__main__":
    main()
```

Протокол работы программы:

Список: [-6, 1, 1, -10, 5, -1, -4, -9, -10, -4]

Количество участков монотонности: 1

Текст программы (6.3):

```
# Дан список размера N. Осуществить сдвиг элементов списка вправо на одну
# позицию (при этом A1 перейдет в A2, A2 — в A3, ..., AN-1 — в AN, а исходное
# значение последнего элемента будет потеряно). Первый элемент полученного списка
# положить равным 0.
import random
def generate_random_list(size, min_value, max_value):
    return [random.randint(min_value, max_value) for _ in range(size)]
def shift_right_and_set_first_to_zero(lst):
    shifted_lst = lst[1:] + [0] # Копируем все элементы кроме первого и
    # добавляем ноль в конец
    shifted_lst[0] = 0          # Устанавливаем первый элемент равным нулю
    return shifted_lst
if __name__ == "__main__":
    N = 10 # Размер списка
    min_value = -100
    max_value = 100
    original_list = generate_random_list(N, min_value, max_value)
    print("Исходный список:", original_list)
    shifted_list = shift_right_and_set_first_to_zero(original_list)
    print("Результат сдвига:", shifted_list)
```

Протокол работы программы:

Исходный список : [86, 66, 22, 21, -73, -54, -7, 73, -1, -32]

Результат сдвига : [0, 22, 21, -73, -54, -7, 73, -1, -32, 0]

Вывод: Я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с функциями в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.