Студент группы ИС-26 Жирков В.А

Практическое занятие №6

Тема: Составление программ со списками в IDE PyCharm Community.
Цель: Закрепить навыки работы со списками, обработкой исключений и оформлением кода по стандарту PEP 8.

Постановка задачи

Задача 1:
Дан список A размера N и целое число K (1 < K < N). Вывести элементы списка с порядковыми номерами, кратными K.

Тип алгоритма: Итеративный.

Задача 2:
Дан список размера N. Найти количество его промежутков монотонности (возрастание или убывание).

Тип алгоритма: Итеративный с проверкой условий.

Задача 3:
Дан список размера N. Осуществить сдвиг элементов вправо на одну позицию, первый элемент заменить на 0.

Тип алгоритма: Модификация списка.

Текст программы

Задача 1

python

Copy

```
def print_multiples_elements():
    """Выводит элементы списка с порядковыми номерами, кратными K."""
    try:
        n = int(input("Введите размер списка N: "))
        k = int(input("Введите число K (1 < K < N): "))
        if k <= 1 or k >= n:
            raise ValueError("K должно быть в диапазоне (1 < K < N).")

        # Генерация списка
```

```
        a = list(range(1, n + 1))
        print("Список A:", a)

        # Выбор элементов с индексами, кратными K (начиная с 0)
        result = [a[i] for i in range(k - 1, n, k)]
        print(f"Элементы с номерами, кратными {k}:", result)

    except ValueError as e:
        print(f"Ошибка: {e}")
        print_multiples_elements()

print_multiples_elements()
```

Задача 2

python

Copy

```
def count_monotonic_segments():
    """Считает количество монотонных участков в списке."""
    try:
        n = int(input("Введите размер списка N: "))
        a = []
        for i in range(n):
            a.append(int(input(f"Введите элемент {i + 1}: ")))

        count = 0
        direction = None  # 1 - возрастание, -1 - убывание

        for i in range(1, n):
            if a[i] > a[i - 1]:
                new_dir = 1
```

```
            elif a[i] < a[i - 1]:
                new_dir = -1
            else:
                new_dir = 0

            if new_dir != 0:
                if direction != new_dir:
                    count += 1
                    direction = new_dir
            else:
                direction = None

        print("Количество монотонных участков:", max(0, count))

    except ValueError:
        print("Ошибка: введите целые числа.")
        count_monotonic_segments()

count_monotonic_segments()
```

Задача 3

python

Copy

```
def shift_right():
    """Сдвигает элементы списка вправо и заменяет первый элемент на 0."""
    try:
        n = int(input("Введите размер списка N: "))
        a = []
        for i in range(n):
            a.append(int(input(f"Введите элемент {i + 1}: ")))
```

```
        # Сдвиг вправо
        a = [0] + a[:-1]
        print("Результирующий список:", a)

    except ValueError:
        print("Ошибка: введите целые числа.")
        shift_right()

shift_right()
```

Протокол работы программы

Задача 1:

Copy

Введите размер списка N: 8

Введите число K (1 < K < N): 3

Список A: [1, 2, 3, 4, 5, 6, 7, 8]

Элементы с номерами, кратными 3: [3, 6]

Задача 2:

Copy

Введите размер списка N: 5

Введите элемент 1: 1

Введите элемент 2: 3

Введите элемент 3: 2

Введите элемент 4: 5

Введите элемент 5: 4

Количество монотонных участков: 3

Задача 3:

Copy

Введите размер списка N: 4

Введите элемент 1: 10

Введите элемент 2: 20

Введите элемент 3: 30

Введите элемент 4: 40

Результирующий список: [0, 10, 20, 30]

Вывод

В ходе выполнения работы были разработаны программы для работы со списками:

1. **Задача 1:** Реализован вывод элементов с индексами, кратными K, с использованием генератора списка.
2. **Задача 2:** Подсчёт монотонных участков выполнен через проверку изменения направления элементов.
3. **Задача 3:** Сдвиг элементов вправо осуществлён через срезы и замену первого элемента.

Код соответствует стандарту PEP 8, включает обработку исключений и комментарии. Программы протестированы на различных входных данных.