

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 список размера N и целые числа K и L ($1 < K < L < N$). Найти среднее арифметическое элементов списка с номерами от K до L включительно.

Текст программы:

```
def calculate_average(lst, K, L):
    # Проверяем, что K и L находятся в допустимых границах
    if not (1 < K < L < len(lst)):
        raise ValueError("Некорректные значения K и L! Должно быть 1 < K < L < N.")

    # Выбираем нужные элементы списка (K-1 и L включительно)
    selected_elements = lst[K-1:L] # Индексы K-1 и L-1 из-за нулевой индексации в Python

    # Находим среднее арифметическое
    average = sum(selected_elements) / len(selected_elements)

    return average, selected_elements

# Главный блок программы
while True:
    try:
        # Ввод данных
        N = int(input("Введите размер списка: ")) # Запрашиваем размер списка у пользователя
        my_list = list(map(int, input(f"Введите {N} целых чисел через пробел: ").split())) # Вводим элементы списка

        # Проверка, что введено ровно N чисел
        if len(my_list) != N:
            raise ValueError(f"Вы должны ввести ровно {N} чисел.")

        # Ввод индексов K и L
        K = int(input("Введите число K: "))
        L = int(input("Введите число L: "))

        # Вычисляем среднее арифметическое
        average, selected_elements = calculate_average(my_list, K, L)

        # Выводим результат
        print(f"Среднее арифметическое элементов списка с индексами от {K} до {L} равно: {average}")

        break # Выход из цикла, если все прошло успешно

    except ValueError as e:
        print(f"Ошибка: {e}. Пожалуйста, введите корректные значения.")
```

Протокол работы программы:

Введите размер списка: 4

Введите 4 целых чисел через пробел: 1 2 3 4

Введите число K: 2

Введите число L: 3

Среднее арифметическое элементов списка с индексами от 2 до 3 равно: 2.5

Постановка задачи №2

Дан целочисленный список размера N. Найти максимальное количество его одинаковых элементов.

Текст программы:

```
def count_max_occurrences(): 1 usage
    while True:
        try:
            # Ввод размера списка
            N = int(input("Введите размер списка: "))
            # Проверка, что размер списка положительный
            if N <= 0:
                raise ValueError("Размер списка должен быть положительным числом.")

            # Ввод элементов списка через пробел и преобразование их в целые числа
            my_list = list(map(int, input(f"Введите {N} чисел через пробел: ").split()))

            # Проверка, что введено ровно N чисел
            if len(my_list) != N:
                raise ValueError(f"Вы должны ввести ровно {N} чисел.")

            # Подсчет количества каждого элемента в списке
            element_count = {}
            for num in my_list:
                # Если элемент уже в словаре, увеличиваем его счетчик
                if num in element_count:
                    element_count[num] += 1
                else:
                    # Если элемента нет, добавляем его со счетчиком 1
                    element_count[num] = 1

            # Поиск максимального количества одинаковых элементов
            max_count = max(element_count.values())
            print(f"Максимальное количество одинаковых элементов: {max_count}")
            break # Выход из цикла при успешном выполнении

        except ValueError as e:
            # Обработка ошибок ввода данных
            print(f"Ошибка: {e}. Пожалуйста, попробуйте еще раз.")

# Вызов функции
count_max_occurrences()
```

Протокол работы программы:

Введите размер списка: 4

Введите 4 чисел через пробел: 2 2 3 4

Максимальное количество одинаковых элементов: 2

Постановка задачи №3

Дан список размера N, все элементы которого, кроме одного, упорядочены по убыванию. Сделать список упорядоченным, переместив элемент, нарушающий упорядоченность, на новую позицию.

Текст программы:

```
def fix_ordered_list(lst): 1 usage
    for i in range(len(lst) - 1):
        # Если текущий элемент меньше следующего, значит порядок нарушен
        if lst[i] < lst[i + 1]:
            out_of_order_index = i + 1 # Сохраняем индекс элемента, который нарушает порядок
            break
    else:
        return lst # Если порядок не нарушен, возвращаем оригинальный список

    out_of_order_element = lst[out_of_order_index] # Извлекаем элемент, который нарушает порядок

    lst.pop(out_of_order_index) # Удаляем его из списка

    # Находим правильную позицию для этого элемента в отсортированном списке
    insert_index = 0
    while insert_index < len(lst) and lst[insert_index] > out_of_order_element:
        insert_index += 1 # Увеличиваем индекс, пока не найдем место для вставки

    lst.insert(insert_index, out_of_order_element) # Вставляем элемент на правильную позицию

    return lst # Возвращаем исправленный список

while True:
    try:
        N = int(input("Введите размер списка: "))
        # Проверка, что размер списка положительный
        if N <= 0:
            raise ValueError("Размер списка должен быть положительным числом.")

        # Ввод элементов списка через пробел и преобразование их в целые числа
        lst = list(map(int, input(f"Введите {N} чисел через пробел: ").split()))

        # Проверка, что введено ровно N чисел
        if len(lst) != N:
            raise ValueError(f"Вы должны ввести ровно {N} чисел.")

        # Вызываем функцию для упорядочивания списка
        sorted_list = fix_ordered_list(lst)
        print("Упорядоченный список:", sorted_list) # Выводим упорядоченный список
        break # Выход из цикла, если все прошло успешно
    except ValueError as e:
        print(f"Ошибка: {e}. Пожалуйста, попробуйте еще раз.") # Обработка ошибок ввода
```

Протокол работы программы:

Введите размер списка: 4

Введите 4 чисел через пробел: 4 2 3 1

Упорядоченный список: [4, 3, 2, 1]

Вывод: в процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.