

Практическое занятие № 6

Тема: Составление программ циклической структуры в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ ветвящейся структуры в IDE PyCharm Community.

Постановка задачи.

Разработать программу, выводящую на экран значение выражения приближенного значения функции \cos в точке X .

Текст программы 6_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.
def average_of_sublist(lst, K, L):
    # Проверяем, что K и L находятся в допустимых пределах
    if K < 1 or L >= len(lst) or K > L:
        raise ValueError("Индексы K и L должны быть в диапазоне от 1 до N и K < L.")

    # Суммируем элементы от K до L включительно
    total_sum = 0
    count = 0

    for i in range(K - 1, L): # Индексы в Python начинаются с 0, поэтому K-1
        total_sum += lst[i]
        count += 1

    # Вычисляем среднее арифметическое
    average = total_sum / count
    return average

# Пример использования функции
N = 10
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] # Пример списка
K = 3
L = 7

result = average_of_sublist(my_list, K, L)
print(f"Среднее арифметическое элементов с индексами от {K} до {L}: {result}")
```

Протокол работы программы 6_1:

C:\Users\night\AppData\Local\Programs\Python\Python312\python.exe

I:\Алгоритмизация\Tovmasyan_py_prof\PZ\PZ_6\PZ_6_1.py

Введите значение 2

Введите значение 3

Среднее арифметическое элементов с индексами от 2 до 3: 2.5

Process finished with exit code 0

Текст программы 6_2:

```
# Дан целочисленный список размера N. Найти максимальное количество его
# одинаковых элементов
def max_equal_elements_count(int_list):
    if not int_list: # Проверка на пустой список
        return 0

    # Словарь для подсчета количества каждого элемента
```

```

count_dict = {}

# Проходим по каждому элементу списка
for element in int_list:
    # Если элемент уже есть в словаре, увеличиваем его счетчик
    if element in count_dict:
        count_dict[element] += 1
    else:
        # Если элемент не найден, добавляем его в словарь с начальным значением 1
        count_dict[element] = 1

# Находим максимальное значение счетчиков
max_count = 0
for count in count_dict.values():
    if count > max_count:
        max_count = count

return max_count

# Пример использования функции
N = int(input("Введите размер списка: "))
int_list = []

print("Введите элементы списка:")
for _ in range(N): # Здесь мы используем цикл for для получения элементов, не используя range
    element = int(input())
    int_list.append(element)

result = max_equal_elements_count(int_list)
print(f"Максимальное количество одинаковых элементов: {result}")

```

Протокол работы программы 6_2:

C:\Users\night\AppData\Local\Programs\Python\Python312\python.exe

I:\Алгоритмизация\Tovmasyan_py_prof\PZ\PZ_6\PZ_6_2.py

Введите размер списка: 4

Введите элементы списка:

3

3

2

1

Максимальное количество одинаковых элементов: 2

Process finished with exit code 0Текст программы 6_3:

```

# Дан список размера N, все элементы которого, кроме одного, упорядочены по
# убыванию. Сделать список упорядоченным, переместив элемент, нарушающий
# упорядоченность, на новую позицию.
def find_and_correct_disorder(int_list):
    N = len(int_list)

    # Найдем индекс элемента, нарушающего порядок
    disorder_index = -1
    for i in range(1, N):
        if int_list[i] > int_list[i - 1]:
            disorder_index = i
            break

    # Если порядок не нарушен, возвращаем оригинальный список
    if disorder_index == -1:
        return int_list

```

```

# Элемент, нарушающий порядок
disorder_element = int_list[disorder_index]

# Удаляем элемент из списка
int_list.pop(disorder_index)

# Найдем правильное место для disorder_element
insert_index = 0
while insert_index < len(int_list) and int_list[insert_index] >= disorder_element:
    insert_index += 1

# Вставляем элемент на правильную позицию
int_list.insert(insert_index, disorder_element)

return int_list

# Пример использования функции
N = int(input("Введите размер списка: "))
int_list = []

print("Введите элементы списка:")
for _ in range(N): # Используем цикл for для ввода элементов
    element = int(input())
    int_list.append(element)

sorted_list = find_and_correct_disorder(int_list)
print("Упорядоченный список:", sorted_list)

```

Протокол работы программы 6_3:

C:\Users\night\AppData\Local\Programs\Python\Python312\python.exe

I:\Алгоритмизация\Tovmasyan_py_prof\PZ\PZ_6\PZ_6_3.py

Введите размер списка: 2

Введите элементы списка:

1

2

Упорядоченный список: [2, 1]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ в IDE PyCharm Community. Были использованы языковые конструкции while, for, in, import, def, range, if else, return, break. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.