

## Практическое занятие №6

**Тема:** составление программ с функциями в IDE PyCharm Community.

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

### Постановка задачи:

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

**Тип алгоритма:** циклический

### Текст программы:

```
#Дан список размера N и целые числа K и L (1 < K < L < N). Найти сумму элементов
списка с номерами от K до L включительно.
import random

while True:
    try:
        N = int(input('Введите размер списка: '))
        break
    except ValueError:
        print("Введите целое число!")

lst = [] #Инициализируем список

# генерация N случайных целых чисел от 1 до 100
for i in range(N):
    lst.append(random.randint(1, 100))

print(f'Сгенерированный список: {lst}')

#Ввод K и L
while True:
    try:
        K = int(input('Введите K: '))
        L = int(input('Введите L: '))
        if 1 <= K <= N and 1 <= L <= N and K <= L:
            break
        else:
            print(f"K и L должны быть в пределах от 1 до {N} и K должно быть меньше
или равно L.")
    except ValueError:
        print("Введите целое число!")

#Вычисление суммы элементов с номерами от K до L
result_sum = sum(lst[K-1:L]) # Суммируем элементы от K до L (включительно)

print(f'Сумма элементов с номерами от {K} до {L}: {result_sum}')
```

## Протокол программ:

Введите размер списка: 10

Сгенерированный список: [35, 99, 17, 99, 9, 18, 4, 85, 81, 65]

Введите K: 3

Введите L: 9

Сумма элементов с номерами от 3 до 9: 313

Process finished with exit code 0

## Постановка задачи:

Дан целочисленный список размера N. Найти количество различных элементов в данном списке.

**Тип алгоритма: Циклический**

## Текст программы:

```
#Дан целочисленный список размера N. Найти количество различных элементов в данном
списке.
import random

while True:
    try:
        N = int(input('Введите размер списка: '))
        if N > 0:
            break
        else:
            print("Размер списка должен быть положительным целым числом.")
    except ValueError:
        print("Введите целое число!")

# Генерация списка случайных целых чисел от 1 до 100
lst = [random.randint(1, 100) for _ in range(N)]

print(f'Сгенерированный список: {lst}')

#Находим количество различных элементов
elements = set(lst) # Используем множество для нахождения уникальных элементов
count = len(elements)

print(f'Количество различных элементов в списке: {count}')
```

## Протокол программ:

Введите размер списка: 10

Сгенерированный список: [98, 58, 90, 68, 15, 76, 79, 88, 88, 9]

Количество различных элементов в списке: 9

Process finished with exit code 0

## Постановка задачи:

Дан список размера N, все элементы которого, кроме последнего, упорядочены по возрастанию. Сделать список упорядоченным, переместив последний элемент на новую позицию.

**Тип алгоритма: Циклический**

## Текст программы:

```
#Дан список размера N, все элементы которого, кроме последнего, упорядочены
#по возрастанию. Сделать список упорядоченным, переместив последний элемент на новую
#позицию.

import random

def insert_last_element(sorted_list):
    #Извлекаем последний элемент
    last_element = sorted_list[-1]

    #Перемещаем его влево, пока не найдем правильное место
    i = len(sorted_list) - 2 # Начинаем с предпоследнего элемента
    while i >= 0 and sorted_list[i] > last_element:
        i -= 1

    #Вставляем элемент на правильное место
    sorted_list[i + 1:] = [last_element] + sorted_list[i + 1:-1]

#Генерируем случайный список размера N
while True:
    try:
        N = int(input('Введите размер списка: '))
        break
    except ValueError:
        print("Введите целое число!")
sorted_list = sorted(random.sample(range(1, 100), N - 1)) #Генерируем N-1 уникальных
#случайных чисел и сортируем
last_element = random.randint(1, 100) # генерируем последний элемент
sorted_list.append(last_element) #Добавляем последний элемент

print("Исходный список:", sorted_list)
insert_last_element(sorted_list)
print("Упорядоченный список:", sorted_list)
```

## **Протокол программ:**

Введите размер списка: 12

Исходный список: [8, 23, 37, 44, 49, 53, 69, 74, 84, 87, 92, 70]

Упорядоченный список: [8, 23, 37, 44, 49, 53, 69, 70, 74, 84, 87, 92]

Process finished with exit code 0

## **Вывод:**

В процессе работы я закрепила полученные ранее навыки, приобрела новые навыки в использование списков и работы с ними, научилась создавать программы с использованием библиотеки random в IDE PyCharm Community.



