

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:

Составить программу, находящую среднее арифметическое элементов списка N с номерами от K до L включительно

Тип алгоритма: циклический

Текст программы:

```
import random
list = []
N = 10
for i in range(N):
    list.append(random.randint(0, 100))
print(f"Список создан:", list)

while True:
    try:
        K = int(input("Введите значение K (1 <= K < L): "))
        if K >= 1 and K < N:
            break
    except ValueError:
        print("Ошибка: Ввод должен быть числом.")

while True:
    try:
        L = int(input("Введите значение L (K < L <= N): "))
        if L > K and L <= N:
            break
    except ValueError:
        print("Ошибка: Ввод должен быть числом.")

def average_of_list(list, K, L):
    elements_of_list = list[K-1:L]
    total_sum = sum(elements_of_list)
    count = len(elements_of_list)
    average = total_sum / count

    return average

result = average_of_list(list, K, L)
print(f"Среднее арифметическое элементов с номерами от {K} до {L} включительно: {result}")
```

Протокол работы программы:

Список создан: [78, 21, 97, 51, 30, 61, 57, 95, 20, 47]

Введите значение K ($1 \leq K < L$): 2

Введите значение L ($K < L < N$): 6

Среднее арифметическое элементов с номерами от 2 до 6 включительно: 52.0

Process finished with exit code 0

Постановка задачи 2:

Составить программу, находящую максимальное количество одинаковых элементов списка

Тип алгоритма: с функцией

Текст программы:

```
import random
random_list = []
N = 10
for i in range(N):
    random_list.append(random.randint(0, 100))
print(f"Список создан:", random_list)

def max_similar_elements(random_list):
    random_list.sort()
    max_count = 1
    current_count = 1

    for i in range(1, len(random_list)):
        if random_list[i] == random_list[i - 1]:
            current_count += 1
        else:
            max_count = max(max_count, current_count)
            current_count = 1

    max_count = max(max_count, current_count)

    return max_count

result = max_similar_elements(random_list)
print("Максимальное количество одинаковых элементов:", result)
```

Студентка группы ИС-23 Погорелова Мария

Протокол работы программы:

Список создан: [75, 50, 93, 28, 94, 68, 68, 69, 45, 78]

Максимальное количество одинаковых элементов: 2

Process finished with exit code 0

Постановка задачи 3:

Составить программу, восстанавливающую упорядоченность элементов списка, в которых все, кроме одного отсортированы по убыванию

Тип алгоритма: циклический

Текст программы:

```
def fix_decreasing_list(random_list):
    if len(random_list) < 2:
        return random_list

    index_out_of_order = -1
    for i in range(1, len(random_list)):
        if random_list[i] > random_list[i - 1]:
            index_out_of_order = i
            break

    if index_out_of_order == -1:
        return random_list

    el_out_of_order = random_list[index_out_of_order]
    random_list.pop(index_out_of_order)

    for i in range(len(random_list)):
        if el_out_of_order > random_list[i]:
            random_list.insert(i, el_out_of_order)
            return random_list

    random_list.append(el_out_of_order)
    return random_list

random_list = [5, 4, 3, 2, 6, 1]
print("Исходный список:", random_list)
sorted_list = fix_decreasing_list(random_list)
print("Упорядоченный список:", sorted_list)
```

Протокол работы программы:

Студентка группы ИС-23 Погорелова Мария

Исходный список: [5, 4, 3, 2, 6, 1]

Упорядоченный список: [6, 5, 4, 3, 2, 1]

Process finished with exit code 0

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