

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.

Задача 1:

Постановка задачи.

Сформировать и вывести целочисленный список размера 10, содержащий 10 первых положительных нечетных чисел: 1,3,5,

Текст программы:

```
odd_numbers = [i for i in range(1, 20, 2)] #создаем список от 1 до 20, с шагом 2
print(odd_numbers)
```

Протокол работы программы:

[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

Process finished with exit code 0

Задача 2:

Постановка задачи.

Разработать программу, где дан список размера N и найти номера тех элементов списка, которые больше своего правого соседа, и количество таких элементов. Найденные номера выводить в порядке их возрастания.

Текст программы:

```
#Дан список размера N. Найти номера тех элементов списка, которые больше своего
#правого соседа, и количество таких элементов. Найденные номера выводить в
#порядке их возрастания.
def find_elements(lst):
    indexes = []
    for i in range(len(lst)-1):
        if lst[i] > lst[i+1]:
            indexes.append(i)
    return indexes, len(indexes)

my_list = [10, 5, 7, 0, 3, 2, 1, 11, 12]
result_indexes, result_count = find_elements(my_list)
print('Список:', my_list)
print("Номера элементов:", result_indexes)
print("Количество таких элементов:", result_count)
```

Протокол работы программы:

Список: [10, 5, 7, 0, 3, 2, 1, 11, 12]

Номера элементов: [0, 2, 4, 5]

Количество таких элементов: 4
Process finished with exit code 0

Задача 3:

Постановка задачи.

Разработать программу, где дан список A размера N и целые числа K и L ($1 < K < L < N$). Переставить в обратном порядке элементы списка, расположенные между элементами AK и AL, не включая эти элементы

Текст программы:

```
#Дан список A размера N и целые числа K и L ( $1 < K < L < N$ ). Переставить в
обратном
#порядке элементы списка, расположенные между элементами AK и AL, не включая
#эти элементы.

def reverse_between(arr, k, l):
    # Извлечение подсписка между K и L
    subarray = arr[k + 1:l] # Элементы не включая AK и AL
    subarray.reverse() # Переворот подсписка

    # Замена в оригинальном списке
    arr[k + 1:l] = subarray
    return arr

# Пример списка и ввода K и L
A = [10, 20, 30, 40, 50, 60, 70, 80, 90] # Пример списка
N = len(A) # Размер списка

# Ввод значений K и L
K = input(f"Введите K ( $1 < K < \{N - 1\}$ ): ")
L = input(f"Введите L ( $K < L < \{N\}$ ): ")

while type(K) != int: # обработка исключений
    try:
        K = int(K)
    except ValueError:
        print("Неправильно ввели!")
        K = input(f"Введите K ( $1 < K < \{N - 1\}$ ): ")
while type(L) != int: # обработка исключений
    try:
        L = int(L)
    except ValueError:
        print("Неправильно ввели!")
        L = input(f"Введите L ( $K < L < \{N\}$ ): ")

# Печать исходного списка
print("Исходный список:", A)

# Перестановка и вывод результата
result = reverse_between(A, K, L)
print("Список после перестановки:", result)
```

Протокол работы программы:

Введите K ($1 < K < 8$): 5

Введите L ($K < L < 9$): 8

Исходный список: [10, 20, 30, 40, 50, 60, 70, 80, 90]

Список после перестановки: [10, 20, 30, 40, 50, 60, 80, 70, 90]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community. Была использована функция def и for.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.