

Практическая работа №6

Тема: составление программ со списками в IDE PyCharm Community.

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

Первая работа

Постановка задачи:

Дан список A размера N и целое число K ($1 < K < N$). Необходимо преобразовать список, увеличив каждый его элемент на исходное значение элемента AK .

Тип алгоритма: линейный.

Текст программы:

```
#Дан список A размера N и целое число K (1 < K < N).
#Преобразовать список, увеличив каждый его элемент на исходное значение элемента AK.
def common_list(A, K):
    if not (1 < K < len(A)):
        raise ValueError("K должно быть больше 1 и меньше размера списка N.")
    increment_value = A[K]
    return [i + increment_value for i in A]

try:
    A = [1, 2, 3, 4, 5, 6]
    K = 2
    modified_list = common_list(A, K)
    print("Исходный список:", A)
    print("Результирующий список:", modified_list)
except ValueError as e:
    print(f"Увы, произошла ошибка: {e}")
```

Протокол работы программы:

Исходный список: [1, 2, 3, 4, 5, 6]

Результирующий список: [4, 5, 6, 7, 8, 9]

Process finished with exit code 0

Вторая работа

Постановка задачи:

Дан целочисленный список A размера N. Необходимо переписать в новый целочисленный список B того же размера вначале все элементы исходного списка с четными номерами, а затем — с нечетными: A2, A4, A6, ..., A1, A3, A5, Условный оператор не использовать.

Тип алгоритма: линейный.

Текст программы:

#Дан целочисленный список A размера N. Переписать в новый целочисленный список B

#того же размера вначале все элементы исходного списка с четными номерами,

#а затем — с нечетными: A2, A4, A6, ..., A1, A3, A5,

#Условный оператор не использовать.

```
def integer_list(A):
    if type(A) is not list:
        raise TypeError("Входные данные должны быть списком.")
    B = A[::2] + A[1::2]
    return B

try:
    A = [2, 4, 6, 8, 10, 12]
    modified_list = integer_list(A)
    print("Исходный список:", A)
    print("Результирующий список:", modified_list)
except TypeError as e:
    print(f"Увы, произошла ошибка: {e}")
```

Протокол работы программы:

Исходный список: [2, 4, 6, 8, 10, 12]

Результирующий список: [2, 6, 10, 4, 8, 12]

Process finished with exit code 0

Третья работа

Постановка задачи:

Дан список размера N и целое число K ($1 < K < N$). Необходимо осуществить сдвиг элементов списка вправо на K позиций (при этом A_1 перейдет в A_{K+1} , A_2 — в A_{K+2} , .. A_{N-K} — в A_N , а исходное значение K последних элементов будет потеряно). Первые K элементов полученного списка положить равными 0.

Тип алгоритма: линейный.

Текст программы:

```
#Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию
#(при этом A1 перейдет в A2, A2 — в A3, ..., AN-1 — в AN,
#a исходное значение последнего элемента будет потеряно).
#Первый элемент полученного списка положить равным 0.
def shift_elements(A):
    if type(A) is not list:
        raise TypeError("Входные данные должны быть списком.")
    return [0] + A[:-1]
try:
    A = [1, 2, 3, 4, 5, 6]
    modified_list = shift_elements(A)
    print("Исходный список:", A)
    print("Результирующий список:", modified_list)
except TypeError as e:
    print(f"Увы, произошла ошибка: {e}")
```

Протокол работы программы:

Исходный список: [1, 2, 3, 4, 5, 6]

Результирующий список: [0, 1, 2, 3, 4, 5]

Process finished with exit code 0

Общий вывод: в процессе выполнения практического занятия выработал навыки составления программы со списками в IDE PyCharm Community. Были использованы языковые конструкции: if, def, try, except, [списки], for.

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