

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.

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

Текст программы:

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

def increase_by_k_element(a, k):
    """
    Увеличивает каждый элемент списка a на значение элемента a[k-1].
    """
    try:
        ak = a[k - 1] # Индексация начинается с 0, поэтому k-1
        return [x + ak for x in a]
    except IndexError:
        print("Ошибка: индекс K выходит за границы списка.")
        return a

# Пример вызова
a = [1, 2, 3, 4, 5]
k = 3
result = increase_by_k_element(a, k)
print(f"Исходный список: {a}")
print(f"Преобразованный список: {result}")
```

Протокол работы программы:

Исходный список: [1, 2, 3, 4, 5]

Преобразованный список: [4, 5, 6, 7, 8]

Постановка задачи.

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.

Текст программы:

```
# 2. Дан целочисленный список A размера N.
# Переписать в новый список B: сначала все элементы с четными номерами, затем с
# нечетными.

def split_even_odd_indices(a):
    """
    Формирует список B, состоящий из элементов списка A:
    сначала с четными номерами, затем с нечетными.
    """
    even = a[1::2] # Четные номера: индексы 1, 3, 5, ...
    odd = a[0::2]  # Нечетные номера: индексы 0, 2, 4, ...
    return even + odd

# Пример вызова
a = [10, 20, 30, 40, 50, 60, 70]
result = split_even_odd_indices(a)
print(f"Исходный список: {a}")
print(f"Новый список: {result}")
```

Протокол работы программы:

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

Новый список: [20, 40, 60, 10, 30, 50, 70]

Process finished with exit code 0

Постановка задачи.

Дан список размера N. Осуществить сдвиг элементов списка вправо на одну позицию (при этом A₁ перейдет в A₂, A₂ — в A₃, ..., A_{N-1} — в A_N, а исходное значение последнего элемента будет потеряно). Первый элемент полученного списка положить равным 0.

Текст программы:

```
# 3. Дан список размера N.
# Осуществить сдвиг элементов списка вправо на одну позицию.
# Первый элемент положить равным 0.

def shift_right(a):
    """
    Осуществляет сдвиг элементов списка вправо на одну позицию.
    Первый элемент нового списка становится равным 0.
    """
    n = len(a)
    if n == 0:
        return [0]
    shifted = [0] + a[:-1] # Добавляем 0 в начало, отбрасывая последний элемент
    return shifted

# Пример вызова
a = [1, 2, 3, 4, 5]
```

```
result = shift_right(a)
print(f"Исходный список: {a}")
print(f"Список после сдвига: {result}")
```

Протокол работы программы:

Исходный список: [1, 2, 3, 4, 5]

Список после сдвига: [0, 1, 2, 3, 4]

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с списками в IDE PyCharm Community. Были использованы языковые конструкции `int`, `input`, `print`, `while`, `def`.

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