

## Практическое задание №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. Дан целочисленный список размера N. Увеличить все четные числа, содержащиеся в списке, на исходное значение первого четного числа. Если четные числа в списке отсутствуют, то оставить список без изменений.
2. Дан список A размера N. Сформировать новый список B того же размера по следующему правилу: элемент BK равен сумме элементов списка A с номерами от 1 до K.
3. Дан список размера N и целое число K ( $1 < K < N$ ). Осуществить сдвиг элементов списка вправо на K позиций (при этом A1 перейдет в AK+1, A2 — в AK+2, ..AN-K — в AN, а исходное значение K последних элементов будет потеряно). Первые K элементов полученного списка положить равными 0.

### 1. Тип алгоритма: Циклический

#### Текст программы:

```
# Дан целочисленный список размера N. Увеличит все четные числа содержащиеся в списке,
на исходное значение
# первого четного числа. Если четные числа в списке отсутствуют, то оставить список без
изменений
def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {_ + 1}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних символов: ")
    return list_gen

def first_even_element(default_list: list) -> int: # Функция находящая первый четный элемент
    for _ in default_list:
        if _ % 2 == 0:
            first = _
            return first
```

```

def count_addiction(default_list: list, first_even_el: int or None) -> list:
    # Функция которая увеличивает четные числа на первое четное число из списка
    if first_even_el is None:
        return default_list
    for i in range(len(default_list)):
        if default_list[i] % 2 == 0:
            default_list[i] += first_even
    return default_list

N, lst = input("Введите длину списка: "), []

while type(N) != int: # Обработчик исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите целое число: ")

lst = list_append(N, lst) # Создание списка
first_even = first_even_element(lst) # Присваивание функции
print(f"Финальный вид списка: {count_addiction(lst, first_even)}") # Вывод результата

```

## Протокол работы программы:

Введите длину списка: 5  
 Введите 1-е число: 1  
 Введите 2-е число: 2  
 Введите 3-е число: 3  
 Введите 4-е число: 4  
 Введите 5-е число: 5  
 Финальный вид списка: [1, 4, 3, 6, 5]

Process finished with exit code 0

## 2. Тип алгоритма: Циклический

### Текст программы:

```

# Дан список A размера N. Сформировать новый список B того же размера по следующему
# правилу:
# элемент Bk равен сумме элементов списка A с номерами от 1 до K.
def newlistfunc(first_list: list, second_list: list) -> list: # Функция формирования списка по
условию
    for K in range(len(first_list)):
        if K >= 1: # Элемент списка с 0 индексом не будет включаться в сумму элементов
            second_list.append(sum(first_list[1:K + 1]))
        else:

```

```

        second_list.append(first_list[K])
    return second_list # Возвращает уже наполненный список и сформированный список

def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {_ + 1}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних символов: ")
    return list_gen

N, A, B = input("Введите длину списка: "), [], []

while type(N) != int: # Обработчик исключений
    try:
        N = int(N)
    except ValueError:
        N = input("Введите целое число: ")

A = list_append(N, A)
print(f"Новый список: {newlistfunc(A, B)}")

```

### Протокол работы программы:

Введите длину списка: 7  
 Введите 1-е число: 4  
 Введите 2-е число: 32  
 Введите 3-е число: 23  
 Введите 4-е число: 1  
 Введите 5-е число: 2  
 Введите 6-е число: 3  
 Введите 7-е число: 4  
 Новый список: [4, 32, 55, 56, 58, 61, 65]

Process finished with exit code 0

### 3. Тип алгоритма: Циклический

#### Текст программы:

# Дан список размера N и целое число размера K ( $1 < K < N$ ) 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.

```
def list_append(n: int, list_gen: list) -> list: # Функция генерации списка
    for _ in range(n):
        list_gen.append(input(f"Введите {_ + 1}-е число: "))
        while type(list_gen[_]) != int: # Обработчик исключений элементов списка
            try:
                list_gen[_] = int(list_gen[_])
            except ValueError:
                list_gen[_] = input(f"Введите целое число {_ + 1} без лишних символов: ")
    return list_gen
```

```
def offset_elements(n: int, k: int, lst: list) -> list: # Функция смещающая элементы вправо на K позиций
    for i in range(n - 1, k - 1, -1):
        lst[i] = lst[i - k]
    for i in range(0, k): # Первые K элементов будет потеряны и на их место встанет '0'
        lst[i] = 0
    return lst
```

```
N, K = input("Введите размер списка: "), input(
    "Введите число K, которое меньше длины списка, но больше 1: ")
```

```
while type(N) != int or type(K) != int: # Обработчик исключений
    try:
        N, K = int(N), int(K)
        if K >= N or 1 >= K:
            raise AttributeError # Если число K не входит в диапазон, то будет вызвано исключение
        break
    except ValueError:
        if type(N) != int:
            N = input("Введите целое число длины списка без лишних символов: ")
        if type(K) != int:
            K = input("Введите число K без лишних символов, которое меньше длины списка, но больше 1: ")
    except AttributeError:
        K = input("Введите целое число K без лишних символов в диапазоне от 1 до Длины списка (1 < K < N): ")
```

```
gen_list = list_append(N, []) # Создание листа с нужным размером
```

```
print(f"Результат: {offset_elements(N, K, gen_list)}") # Вывод списка
```

## Протокол работы программы:

Введите размер списка: 8

Введите число K, которое меньше длины списка, но больше 1: 5

Введите 1-е число: 1

Введите 2-е число: 2  
Введите 3-е число: 3  
Введите 4-е число: 4  
Введите 5-е число: 5  
Введите 6-е число: 6  
Введите 7-е число: 7  
Введите 8-е число: 8  
Результат: [0, 0, 0, 0, 0, 1, 2, 3]

Process finished with exit code 0

**Вывод:** закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ со списками в IDE PyCharm Community.