

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 A_K .

Тип алгоритма: циклический.

Текст программы:

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

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:
    N = int(input("Введите количество элементов в списке A: "))
    if N <= 0:
        raise ValueError("Количество элементов должно быть больше 0.")
    A = [random.randint(1, 100) for _ in range(N)]
    K = int(input("Введите значение K: "))
    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:

Введите количество элементов в списке A: 6

Введите значение K: 2

Исходный список: [86, 89, 78, 38, 73, 57]

Результирующий список: [164, 167, 156, 116, 151, 135]

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: циклический.

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,
#a затем — с нечетными: A2, A4, A6, ..., A1, A3, A5, ... .
#Условный оператор не использовать.
import random
def integer_list(A):
    B = A[::2] + A[1::2]
    return B
try:
    N = int(input("Введите количество элементов в списке A: "))
    if N <= 0:
        raise ValueError("Количество элементов должно быть больше 0.")
    A = [random.randint(1, 100) for _ in range(N)]
    modified_list = integer_list(A)
    print("Исходный список:", A)
    print("Результирующий список:", modified_list)
except ValueError as e:
    print(f"Увы, произошла ошибка: {e}")
```

Протокол работы программы:

Введите количество элементов в списке A: 6

Исходный список: [86, 69, 39, 33, 64, 12]

Результирующий список: [86, 39, 64, 69, 33, 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: циклический.

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,
```

```
 #а исходное значение последнего элемента будет потеряно).
```

```
 #Первый элемент полученного списка положить равным 0.
```

```
import random
```

```
def integer_list(A):
```

```
    return [0] + A[:-1]
```

```
try:
```

```
    N = int(input("Введите количество элементов в списке A: "))
```

```
    if N <= 0:
```

```
        raise ValueError("Количество элементов должно быть больше 0.")
```

```
    A = [random.randint(1, 100) for _ in range(N)]
```

```
    modified_list = integer_list(A)
```

```
    print("Исходный список:", A)
```

```
    print("Результирующий список:", modified_list)
```

```
except ValueError as e:
```

```
    print(f"Увы, произошла ошибка: {e}")
```

Протокол работы программы:

Введите количество элементов в списке A: 6

Исходный список: [63, 99, 91, 73, 42, 12]

Результирующий список: [0, 63, 99, 91, 73, 42]

Process finished with exit code 0

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

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