

Практическое занятие № 6

Тема: составление программ со списками в IDE PyCharm Community.

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

№ 6.1

Постановка задачи: Дан список A размера N (N — нечетное число). Вывести его элементы с нечетными номерами в порядке убывания номеров: AN, AN-2, AN-4, ..., A1. Условный оператор не использовать.

Тип алгоритма: циклический со списками.

Текст программы:

```
# вариант 12
# Дан список A размера N (N — нечетное число). Вывести его элементы с нечетными
# номерами в порядке убывания номеров: AN, AN-2, AN-4, ..., A1. Условный оператор
# не
# использовать.
import random

try:
    a = [random.randint(1, 100) for i in range(int(input('Введите длину списка
(нечетное положительное число): ')))]

    while len(a) % 2 == 0:
        print('Нужно ввести целое нечетное положительное значение длины списка!')
        a = [random.randint(1, 100) for i in range(int(input('Введите длину списка
(нечетное положительное число): ')))]
    print('Ваш список: \n', a)
    print('Нечетные номера Вашего списка в порядке убывания:\n', a[-1::-2])
except Exception as ex:
    print('произошла ошибка:\n', ex)
```

Протокол работы программы:

Введите длину списка (нечетное положительное число): 12

Нужно ввести целое нечетное положительное значение длины списка!

Введите длину списка (нечетное положительное число): 11

Ваш список:

[95, 59, 85, 23, 38, 65, 37, 52, 86, 96, 1]

Нечетные номера Вашего списка в порядке убывания:

[1, 86, 37, 38, 85, 95]

Process finished with exit code 0

№ 6.2

Постановка задачи. Дан список размера N. Найти два соседних элемента, сумма которых максимальна, и вывести эти элементы в порядке возрастания их индексов.

Тип алгоритма: циклический со списками.

Текст программы:

```
# вариант 12
# Дан список размера N. Найти два соседних элемента, сумма которых максимальна,
# и вывести эти элементы в порядке возрастания их индексов.
import random

try:
    listA = [random.randint(1, 100) for i in range(int(input('Введите длину списка:
')))]
    print(f'Ваш список:\n{listA}')
    if len(listA) == 2:
        print(*listA)
    elif len(listA) < 2:
        print('Не получится найти сумму значений < 2')
    else:
        SumMax = listA[1] + listA[2]
        k = 2
        for i in range(1, len(listA)):
            if (listA[i - 1] + listA[i]) > SumMax:
                k = i
                SumMax = listA[i - 1] + listA[i]
        print(f'Соседние числа с наибольшей суммой: {listA[k - 1]}, {listA[k]}')
except Exception as ex:
    print('Произошла ошибка: ', ex)
```

Протокол программы:

Введите длину списка: 12

Ваш список:

[10, 95, 8, 49, 73, 20, 65, 26, 27, 57, 32, 59]

Соседние числа с наибольшей суммой: 49 и 73

Process finished with exit code 0

№ 6.3

Постановка задачи. Дан список размера N и целое число 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.

Тип алгоритма: циклический со списками.

Текст программы:

```
# вариант 12
# 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.
try:
    lst = [i+1 for i in range(int(input('Введите длину списка (n): ')))]
    print(f'Ваш список: {lst}')
    k = int(input('Введите (k) длину сдвига вправо (1<k<n): '))
    if 1 < k < len(lst):
        for j in range(k):
            lst = [0]+lst[:-1]
        print(f'Список, сдвинутый {k} значений, \nпервые {k} элементов которого
приравнены к 0: {lst}')
    else:
        print('Соблюдайте все условия при вводе')
except Exception as ex:
    print(ex)
```

Протокол программы:

Введите длину списка (n): 8

Ваш список: [1, 2, 3, 4, 5, 6, 7, 8]

Введите (k) длину сдвига вправо ($1 < k < n$): 3

Список, сдвинутый 3 значений,

первые 3 элементов которого приравнены к 0: [0, 0, 0, 1, 2, 3, 4, 5]

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ циклической структуры с использованием списков и библиотек в IDE PyCharm Community. Были использованы языковые конструкции [while](#), [for](#). Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на [GitHub](#).

