Студентки группы ИС-24 Кухарец И.А.

**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. Вывести вначале его элементы с четными номерами (по возрастанию), затем с нечетными номерами. Условный оператор не использовать.

Дано число R и список размера N. Найти два соседних элемента списка, сумма которых наиболее близка к числу R, и вывести эти элементы в порядке возрастания их индексов.

Дан список размера N и целое число K (1 < K < N). Осуществить сдвиг влево на K позиций. Последние K элементов полученного списка обнулить.

**Тип алгоритма:** Линейный, ветвящийся, циклический

**Текст программы №1:**

```
# Дан список A размера N. Вывести сначала элементы с четными номерами,
# а затем вывести элементы с нечетными номерами.
# Без условного оператора
try:
    lst = list(map(str,input("Введите значения списка -> ").split()))
    print(f'Нечетные элементы -> {lst[1::2]}')
    print(f'Четные элементы -> {lst[2::2]}')
except:
    print('Error')
```

**Текст программы №2:**

```
# Дано число R и список размера N. Найти два соседних элемента списка,
#  сумма которых наиболее близка к числу R.
# Вывести эти элементы в порядке возрастания их индексов
from random import randint
try:
    n = int(input("Количество элементов в списке -> "))
    r = int(input("Введите число ->"))
    lst = []
    c = []
    dop = []
    for i in range(n):  # наполняем список
        a = randint(1, 20)
        lst.append(a)
    print(lst)
    for i, x in enumerate(lst):
```

```
        dop.append(lst[i:i+2])
        c.append(sum(lst[i:i+2]))
    c.pop(-1)
    print(f'Суммы соседних чисел -> {c}')
    def near_v(a, r):  # ближайшее число к сумме соседних чисел
        found = c[0]
        for i in c:
            if abs(i - r) < abs(found - r):
                found = i
                for j in range(len(c)):
                    if c[j] == found:
                        print(f'Соседние числа -> {dop[j]}')
        return found
    print(f'Ближайшая сумма -> {near_v(a, r)}')
except:
    print("Error")
```

**Текст программы №3:**

```
# Дан список размера N и целое число K (1 < K < N).
# Осуществлять сдвиг элементов списа влево на K позиций.
# Последние K элементов полученного списка положить равными 0
from random import randint
try:
    n = int(input("Количество элементов в списке -> "))
    k = int(input("Введите число K (1 < K < N) -> "))
    if k > n or k < 1:
        while k > n or k < 1:
            k = int(input("Введите число K (1 < K < N) -> "))
    lst = []
    for i in range(n):  # наполняем список
        a = randint(1, 20)
        lst.append(a)
    print(lst)
    x = 0
    while x < k: # сдвиг влево
        lst.pop(0)
        lst.append(0)
        x += 1
    print(lst)
except:
    print("Error")
```

Студентки группы ИС-24 Кухарец И.А.

**Протокол программы №1:**

Введите значения списка -> 1 2 3 4 5 6
Нечетные элементы -> ['2', '4', '6']
Четные элементы -> ['3', '5']

**Протокол программы №2:**

Количество элементов в списке -> 5
Введите число ->11
[18, 11, 14, 18, 8]
Суммы соседних чисел -> [29, 25, 32, 26]
Соседние числа -> [11, 14]
Ближайшая сумма -> 25

**Протокол программы №3:**

Количество элементов в списке -> 9
Введите число K (1 < K < N) -> 4
[4, 2, 8, 3, 19, 7, 17, 10, 13]
[19, 7, 17, 10, 13, 0, 0, 0, 0]

**Вывод:** В процессе выполнения практического занятия выработала навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции *try/except, while, def(), for, list(), if.*

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды были выложены на GitHub.