

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 № 1.

Дан список A размера N. Вывести вначале его элементы с нечетными номерами в порядке возрастания номеров, а затем — элементы с четными номерами в порядке убывания номеров: A1, A3, A5, ..., A6, A4, A2. Условный оператор не использовать.

Текст программы:

```
# Дан список A размера N. Вывести вначале его элементы с нечетными номерами в
# порядке возрастания номеров, а затем — элементы с четными номерами в порядке
# убывания номеров: A1, A3, A5, ..., A6, A4, A2. Условный оператор не использовать.
from random import randint #импортирование модуля рандомных чисел
while True: #обработка исключений
    try:
        N = randint(10,20)
        A = [randint(1,10) for i in range(N)]
        print('Наш список:')
        print(A)

        print('Нечётные по порядку: ')
        print(A[::2])

        print("Чётные в обратном порядке: ")
        print(A[-1::-2])
        break
    except:
        print('Ошибка!')
```

Протокол работы программы:

Наш список:

[7, 2, 7, 9, 3, 3, 2, 4, 4, 3, 10, 5]

Нечётные по порядку:

[7, 7, 3, 2, 4, 10]

Чётные в обратном порядке:

[5, 3, 4, 3, 9, 2]

Process finished with exit code 0

Постановка задачи № 2.

Дан список A размера N. Сформировать два новых списка B и C: в список B записать все положительные элементы списка A, в список C — все отрицательные (сохраняя исходный порядок следования элементов). Вывести вначале размер и содержимое списка B, а затем — размер и содержимое списка C.

Текст программы:

```
# Дан список A размера N. Сформировать два новых списка B и C: в список B записать
# все положительные элементы списка A, в список C — все отрицательные (сохраняя
# исходный порядок следования элементов). Вывести вначале размер и содержимое
# списка B, а затем — размер и содержимое списка C.
from random import randint #импортирование модуля рандомных чисел
while True: #обработка исключений
    try:
        N = randint(10, 20)
        A = [randint(-10, 10) for i in range(N)]
        B = []
        C = []
        print('Наш список: ')
        print(A)
        for i in A:
            if i>0:
                B.append(i)
            else:
                C.append(i)
        print('Размер списка B:', len(B))
        print("Список B:", B)
        print("Размер списка C:", len(C))
        print("Список C:", C)
        break
    except:
        print('Ошибка!')
```

Протокол работы программы:

Наш список:

[-6, 6, 0, 2, 3, 3, 10, -10, 2, 8, 9, 2, 1, 7, 7, 10, 2, 2, -9, -3]

Размер списка B: 15

Список B: [6, 2, 3, 3, 10, 2, 8, 9, 2, 1, 7, 7, 10, 2, 2]

Размер списка C: 5

Список C: [-6, 0, -10, -9, -3]

Process finished with exit code 0

Постановка задачи № 3.

Дан список размера N. Осуществить циклический сдвиг элементов списка вправо на одну позицию (при этом A1 перейдет в A2, A2 — в A3,..., AN — в A1).

Текст программы:

```

# Дан список размера N. Осуществить циклический сдвиг элементов списка вправо на
# одну позицию (при этом A1 перейдет в A2, A2 — в A3,..., AN — в A1).
from random import randint #импортирование модуля случайных чисел
while True: #обработка исключений
    try:
        def shift(lst, steps):
            if steps < 0:
                steps = abs(steps)
                for i in range(steps):
                    lst.append(lst.pop(0))
            else:
                for i in range(steps):
                    lst.insert(0, lst.pop())
        N = randint(0, 10)
        A = [randint(1, 10) for i in range(N)]
        print("Наш список:")
        print(A)
        shift(A, 1)
        print("Список со сдвигом вправо:")
        print(A)
        break
    except:
        print('Ошибка!')

```

Протокол работы программы:

Наш список:

[6, 2, 10, 2, 2]

Список со сдвигом вправо:

[2, 6, 2, 10, 2]

Process finished with exit code 0

Вывод: Я выработал первичные навыки работы с IDE PyCharm Community, составил программу ветвящейся структуры. Были использованы языковые конструкции def, try, except, while, for, range, if, else.

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

Готовые программные кода были выложены на GitHub.