

Практическая работа №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 первый член A и разность D арифметической прогрессии. Сформировать и вывести список размера 10, содержащий 10 первых членов данной прогрессии: $A, A + D, A + 2*D, A + 3*D, \dots$
2. Дан список размера N . Найти номер его первого локального минимума (локальный минимум - это элемент, который меньше любого из своих соседей).
3. Дан список размера N (N - четное число). Поменять местами его первый элемент со вторым, третий - с четвертым и т. д.

Тип алгоритма: циклический, ветвление.

Текст программы:

Задача 1.

```
# Дан первый член A и разность D арифметической прогрессии. Сформировать и вывести
# список размера 10, содержащий 10 первых членов данной прогрессии: A, A + D, A +
# 2*D, A + 3*D, ... .

def inp(): # Функция обработки исключения
    a = input('Введите целое число: ')
    while type(a) != int:
        try:
            a = int(a)
            if a <= 0:
                a = input('Введите целое число: ')
        except ValueError:
            a = input('Введите целое число: ')
    return a

# Блок ввода A и D с использованием функции обработки исключений
print('Первый член последовательности')
A = inp()
print('-----')
print('Разность арифметической последовательности')
D = inp()

# Имеется список с 1 членом последовательности, цикл добавляет новые члены до 10
```

включительно

```
def summ():  
    lst = [A]  
    for _ in range(10 - 1):  
        lst.append(A + D * (_ + 1))  
    return lst  
  
# Вывод списка 10 членов прогрессии  
print(summ())
```

Задача 2:

Дан список размера N. Найти номер его первого локального минимума (локальный минимум — это элемент, который меньше любого из своих соседей).

```
from random import randint
```

```
def inp(): # Функция обработки исключения  
    a = input('Введите размер списка: ')  
    while type(a) != int:  
        try:  
            a = int(a)  
            if a <= 0:  
                a = input('Введите целое число: ')  
        except ValueError:  
            a = input('Введите целое число: ')  
    return a
```

def minObj(predel): # Функция со сравнением следующего и предыдущего элемента относительно текущего в списке

```
    lst = []  
    for _ in range(predel):  
        lst.append(randint(-10, 10))  
  
    for _ in range(len(lst)):  
        if lst[_] < lst[_ + 1] and lst[_] < lst[_ - 1]:  
            numba = _ + 1  
            return lst[_], lst, numba
```

Вывод локального минимума и его идентификатора

```
LocMin, AllList, numba = minObj(inp())  
print(f"Первый локальный минимум списка {AllList}: {LocMin} под номером {numba}")
```

Задача 3:

```
# Дан список размера N (N — четное число). Поменять местами его первый элемент со
# вторым, третий — с четвертым и т. д.

from random import randint

def inp(): # Функция обработки исключения
    a = input('Введите целое чётное число: ')
    while type(a) != int:
        try:
            a = int(a)
            if a <= 0 or a % 2 != 0: # Необходимо ввести чётное и положительное число для размера
списка
                a = input('Введите целое чётное число: ')
        except ValueError:
            a = input('Введите целое чётное число: ')

    return a

def lst(): # В функции определяется массив, в который будут записываться изменённые
элементы, точнее их последовательность
    lst1 = []
    for _ in range(inp()):
        lst1.append(randint(-10, 10))
    print(f'Список был: {lst1} \n-----')

    for _ in range(0, len(lst1), 2):
        lst1[_], lst1[_ + 1] = lst1[_ + 1], lst1[_]

    return lst1

print(f'Список стал: {lst()}') # Вывод списка с изменёнными элементами
```

Протокол работы программы:

Задача 1:

Первый член последовательности

Введите целое число: 2

Разность арифметической последовательности

Введите целое число: 4

[2, 6, 10, 14, 18, 22, 26, 30, 34, 38]

Process finished with exit code 0

Задача 2:

Введите размер списка: 5

Первый локальный минимум списка [-5, -9, 4, 9, -9]: -9 под номером 2

Process finished with exit code 0

Задача 3:

Введите целое чётное число: 4

Список был: [-1, -7, -8, -4]

Список стал: [-7, -1, -4, -8]

Process finished with exit code 0

Схема 1:

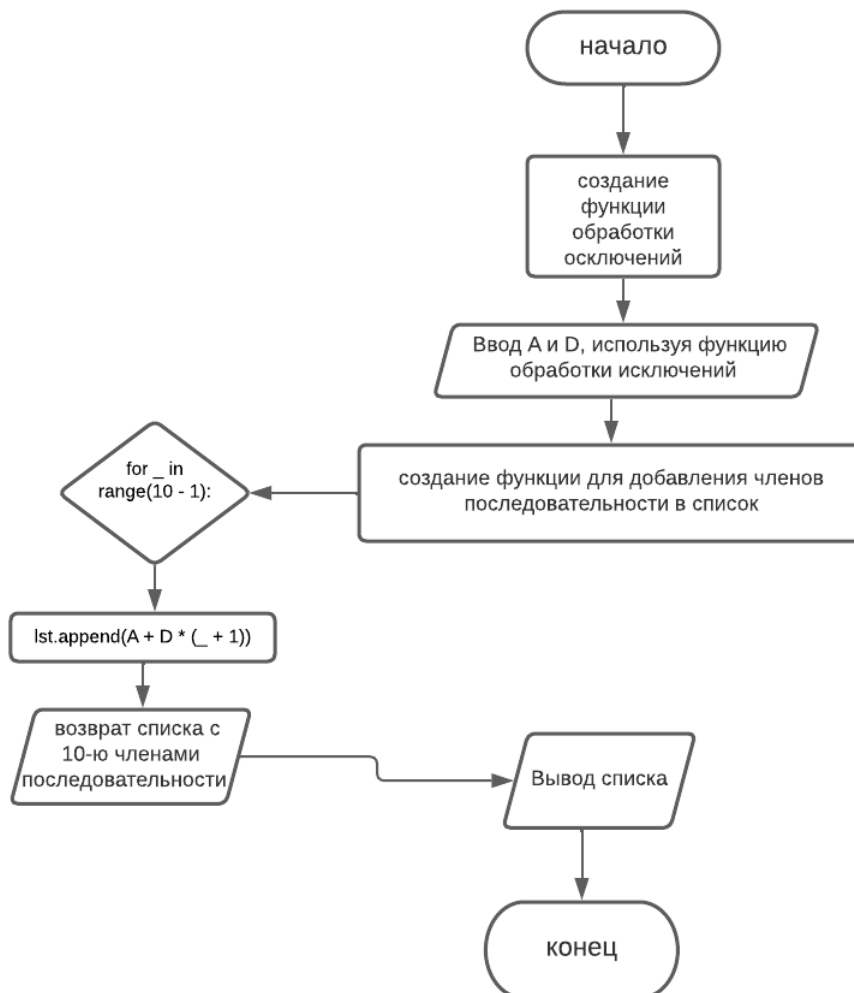


Схема 2:

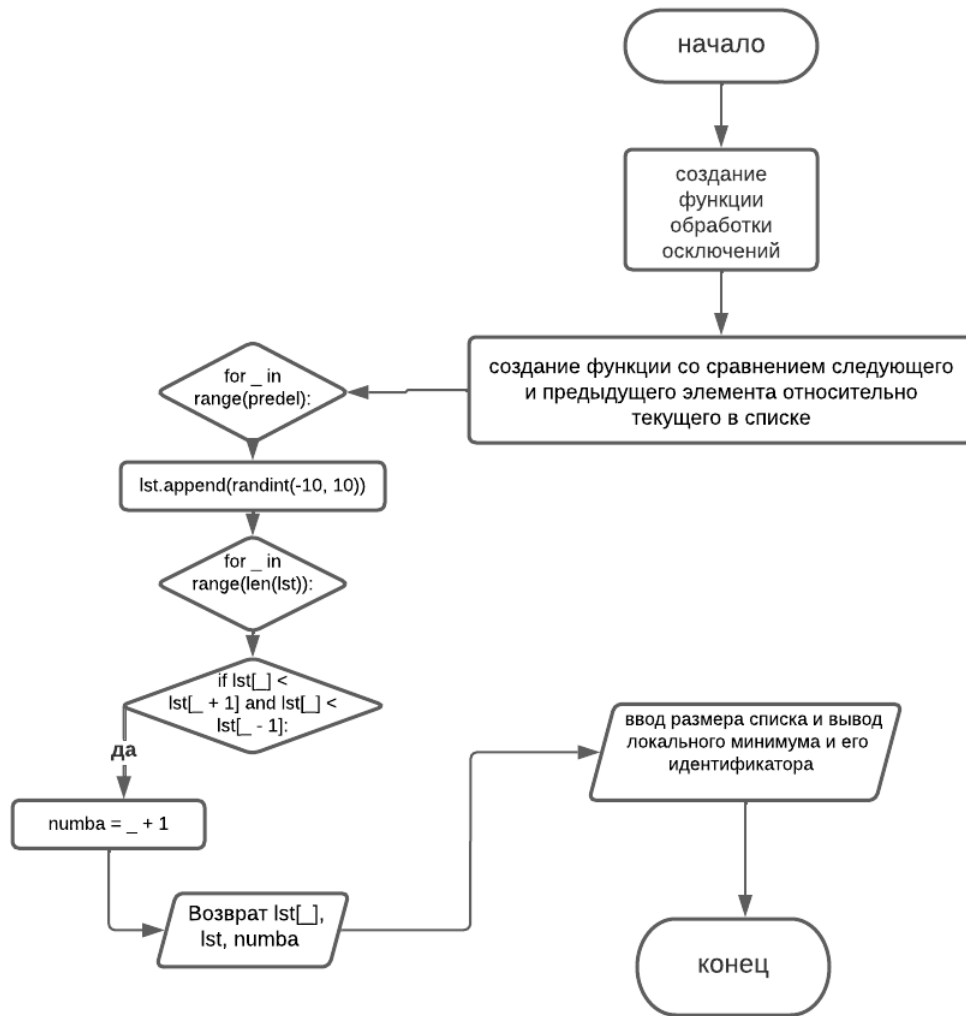
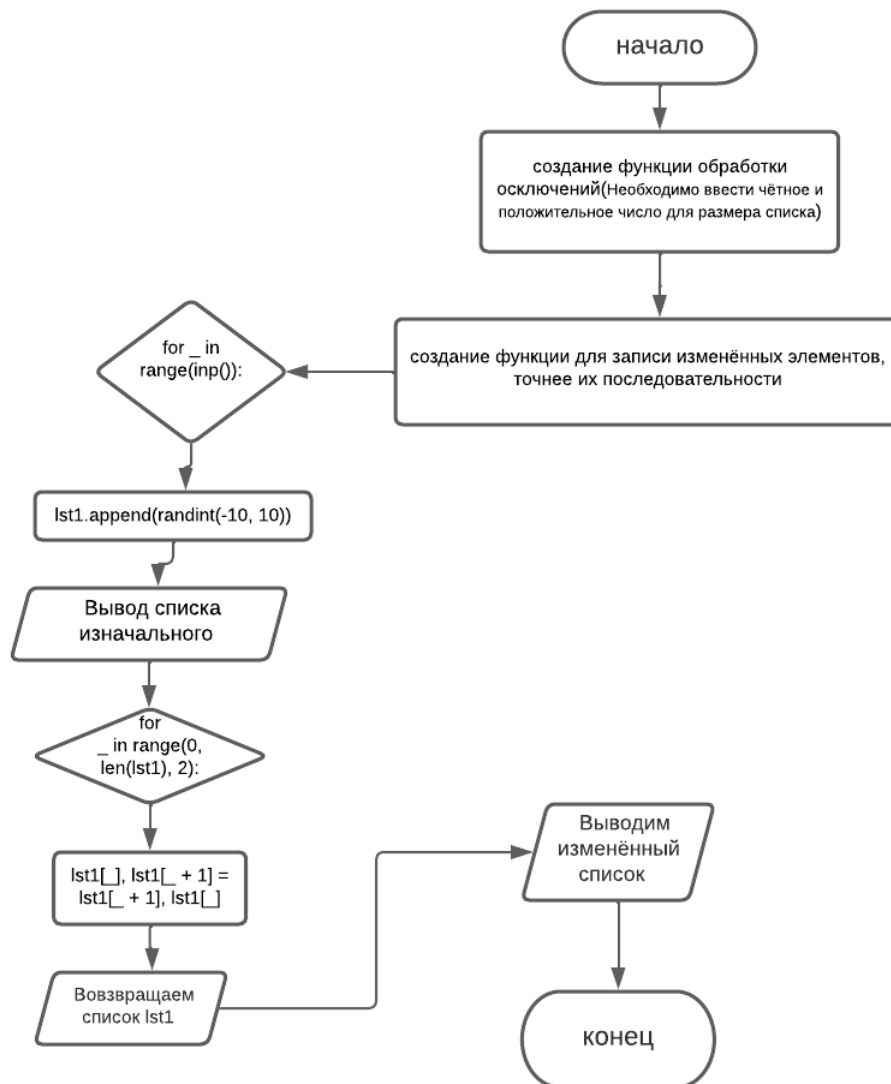


Схема 3:



Вывод: в процессе выполнения практического занятия я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ со списками в IDE Pycharm Community.