Студент группы ИС-25 Дудков Д.Ю.

Практическое занятие № 6

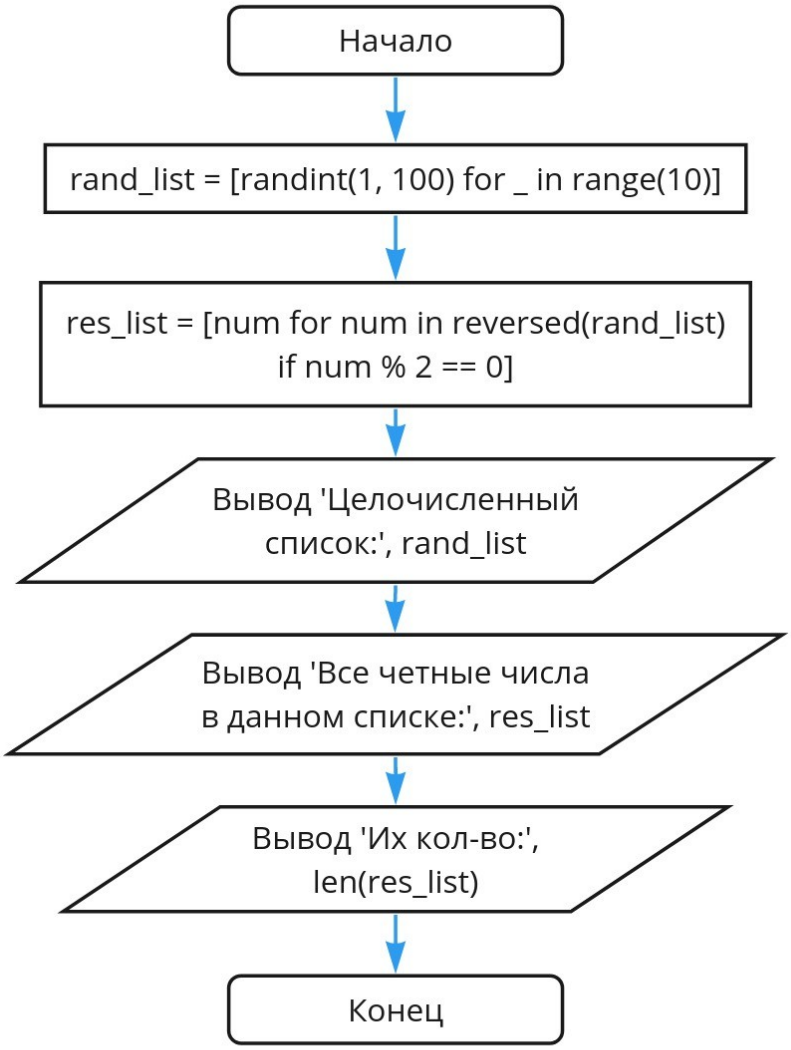
Тема: составление программ со списками в IDE PyCharm Community.

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

Постановка задачи.
Вывести все содержащиеся в данном целочисленном списке размера 10. четные числа в порядке убывания их индексов, а также их количество K.

Тип алгоритма: линейный

Блок-схема алгоритма:



Студент группы ИС-25 Дудков Д.Ю.

Текст программы:

```
from random import randint



def first_task():
    # генератором списков создаём список с рандомными числами
    rand_list = [randint(1, 100) for _ in range(10)]

    res_list = [num for num in reversed(rand_list) if num % 2 == 0]

    print('Целочисленный список:', rand_list)
    print('Все четные числа в данном списке:', res_list)
    print('Их кол-во:', len(res_list))



def main():
    first_task()



if __name__ == '__main__':
    main()
```

Протокол работы программы:

Целочисленный список: [60, 33, 89, 92, 84, 53, 76, 81, 19, 70]
Все четные числа в данном списке: [70, 76, 84, 92, 60]
Их кол-во: 5

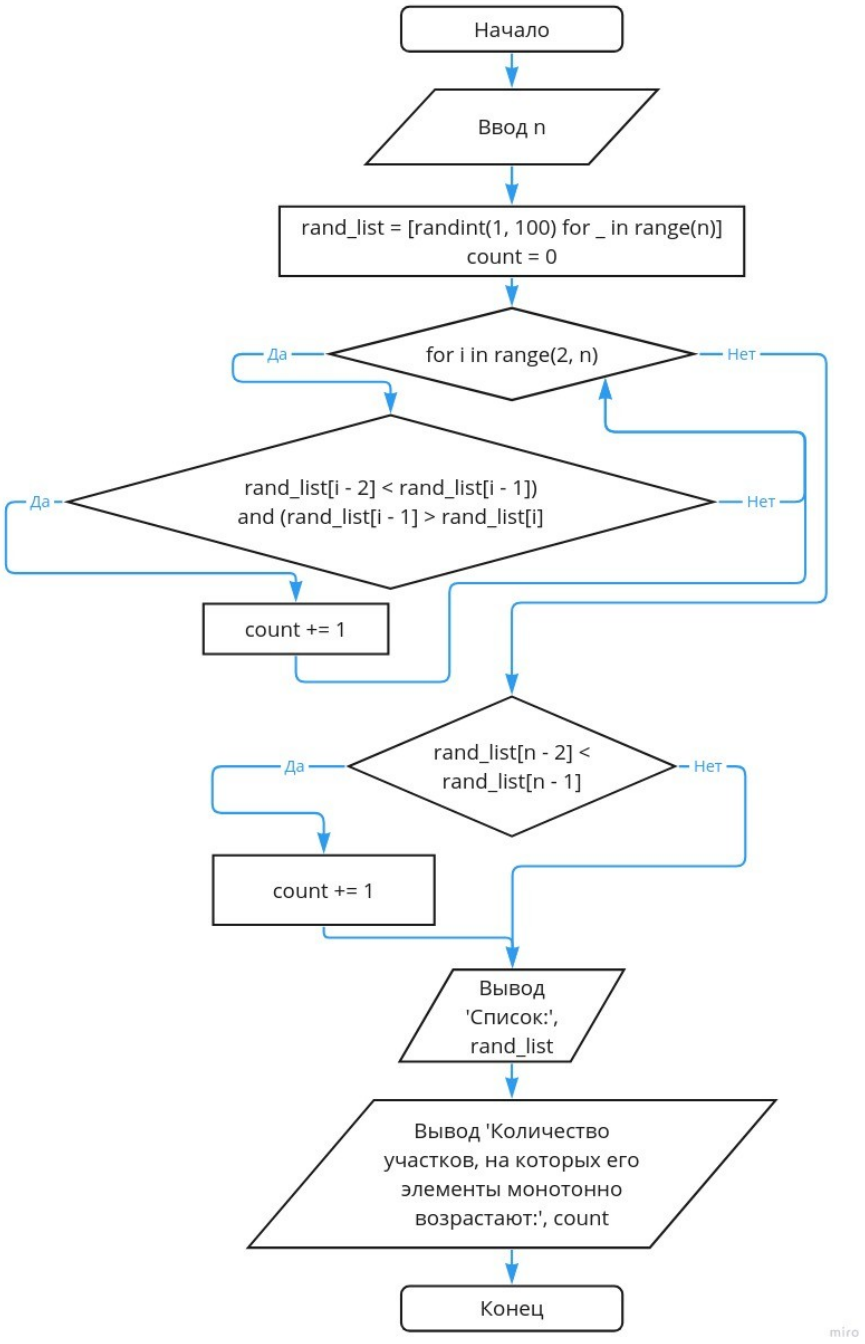
Process finished with exit code 0

Студент группы ИС-25 Дудков Д.Ю.

Постановка задачи.
Найти количество участков, на которых элементы списка размера N монотонно возрастают.

Тип алгоритма: циклический, ветвление.

Блок-схема алгоритма:



Студент группы ИС-25 Дудков Д.Ю.

Текст программы:

```
from random import randint



def second_task(n: int):
    # генератором списков создаём список с рандомными числами
    rand_list = [randint(1, 100) for _ in range(n)]

    count = 0

    for i in range(2, n):
        if (rand_list[i - 2] < rand_list[i - 1]) and (rand_list[i - 1] > rand_list[i]):
            count += 1

    # проверка предпоследнего и последнего элементов
    if rand_list[n - 2] < rand_list[n - 1]:
        count += 1

    print('Список:', rand_list)
    print('Количество участков, на которых его элементы монотонно возрастают:', count)



def main():
    try:
        n = int(input('Длина списка: '))

        # если n меньше 1, то возбуждаем ошибку
        if n < 1:
            raise ValueError

        second_task(n=n)
    except ValueError:
        print('\033[31mНеверное число!\033[0m')
        main()



if __name__ == '__main__':
    main()
```

Протокол работы программы:

Длина списка: 7
Список: [10, 67, 12, 43, 32, 42, 6]
Количество участков, на которых его элементы монотонно возрастают: 3

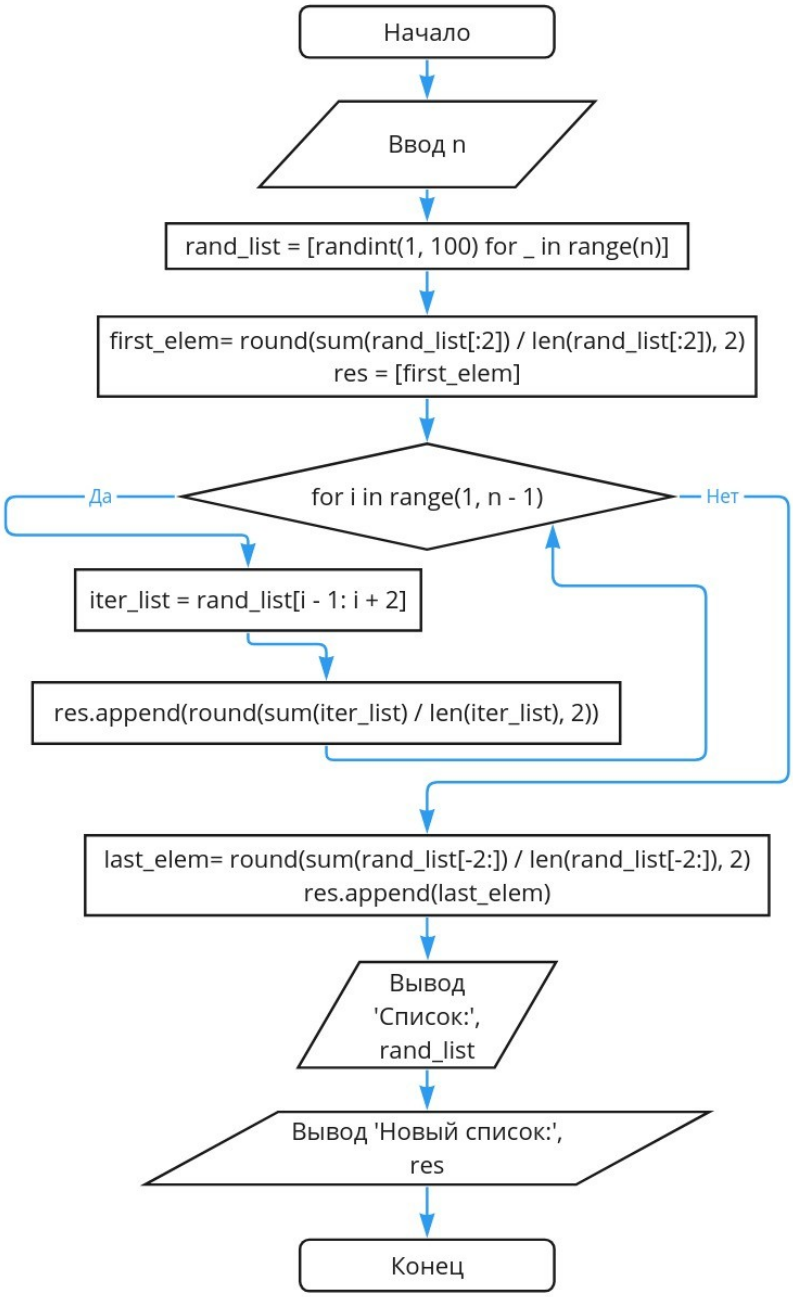
Process finished with exit code 0

Студент группы ИС-25 Дудков Д.Ю.

Постановка задачи.
Дан список размера N. Заменить каждый элемент списка на среднее арифметическое этого элемента и его соседей.

Тип алгоритма: циклический.

Блок-схема алгоритма:



Студент группы ИС-25 Дудков Д.Ю.

Текст программы:

```
from random import randint



# среднее арифметическое всех элементов списка
def avg(iter_list: list) -> float:
    return round(sum(iter_list) / len(iter_list), 2)



def third_task(n: int):
    # генератором списков создаём список с рандомными числами
    rand_list = [randint(1, 100) for _ in range(n)]

    # первый элемент
    res = [avg(rand_list[:2])]
    # все остальные элементы
    res += [
        avg(rand_list[i - 1: i + 2])
        for i in range(1, n - 1)
    ]
    # последний элемент
    res.append(avg(rand_list[-2:]))

    print('Список:', rand_list)
    print('Новый список:', res)



def main():
    try:
        n = int(input('Длина списка: '))

        # если n меньше 1, то возбуждаем ошибку
        if n < 1:
            raise ValueError

        third_task(n=n)
    except ValueError:
        print('\033[31mНеверное число!\033[0m')
        main()



if __name__ == '__main__':
    main()
```

Протокол работы программы:

Длина списка: 8
Список: [67, 96, 73, 62, 57, 62, 84, 35]
Новый список: [81.5, 78.67, 77.0, 64.0, 60.33, 67.67, 60.33, 59.5]

Студент группы ИС-25 Дудков Д.Ю.

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрёл навыки составления программ со списками в IDE PyCharm Community. Были использованы языковые конструкции def, if, for.
Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.
Готовые программные коды выложены на GitHub.