Министерство образования
Республики Беларусь
Учреждение образования
Белорусский государственный университет
информатики и радиоэлектроники

Кафедра: интеллектуальных информационных технологий

Отчёт
По дисциплине
«Метрология, стандартизация и сертификация (в информационных технологиях)»
Тема: Оценка программного продукта на основе метрик

Написал:
Астахов А.С., гр. 321701

Проверил
Волынец А.С.

Минск 2024

Цель: Написать алгоритм и на его основе научиться анализировать качество программного кода.

Задание: для своего варианта индивидуального задания разработать детализированную схему алгоритма, представленную в соответствии с положениями ГОСТ 19701-90. В алгоритме предусмотреть вывод на экран всех входных и выходных данных. На основании разработанного алгоритма создать программный код программы.

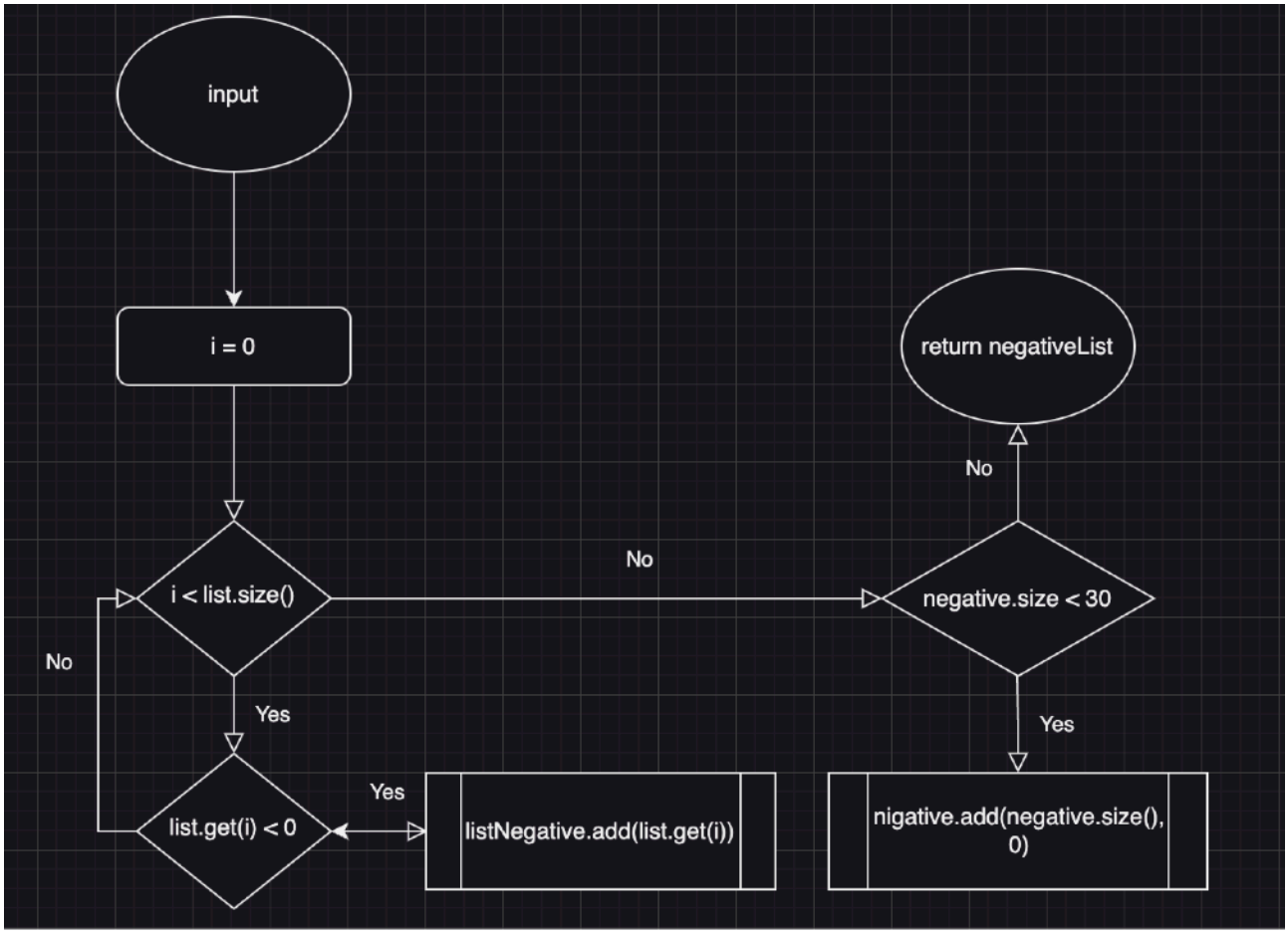
1. Из последовательности чисел A1, A2, ..., A30 выбрать отрицательные четные числа. Их значения поместить в массив B (30). Остаток массива B заполнить нулями. Вывести исходные числа и массив B.

Листинг кода:

```java
import java.util.ArrayList;
import java.util.Random;

public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> list = generateRandomList();
        ArrayList<Integer> NegativeList = getNegativeValuesFromList(list);

        for (Integer integer : NegativeList) {
            System.out.println(integer);
        }
    }

    public static ArrayList<Integer> generateRandomList() {
        ArrayList<Integer> list = new ArrayList<>();
        Random random = new Random();

        for (int i = 0; i < 30; i++) {
            int randomValue = random.nextInt(201) - 100; // Диапазон от -100 до 100
            list.add(randomValue);
```

```
        }

        return list;
    }

    public static ArrayList<Integer>
getNegativeValuesFromList(ArrayList<Integer> list) {
        ArrayList<Integer> negativeList = new ArrayList<>();
        for (Integer integer : list) {
            if (integer < 0) {
                negativeList.add(integer);
            }
        }

        while (negativeList.size() < 30) {
            negativeList.add(negativeList.size(), 0);
        }

        return negativeList;
    }

}
```

Блок-схема:



Результаты вычислений:

-39 -54 -74 -3 -44 -13 -45 -17 -16 -74 -85 -65 -43 -43 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Вывод: написал алгоритм и на его основе научился анализировать качество программного кода.

Метрика Маккейба

Цикломатическая сложность вычисляется по формуле V(G) = E - N + 2P, где:
- E: количество рёбер графа = 8,
- N: количество узлов графа = 8,
- P: количество компонент связности = 1.

V(G) = 8 - 8 + 2(1) = 2.

Метрика Джилба

Абсолютная сложность (CL): 2. Максимальный уровень вложенности (CLI): 1.
Относительная сложность: cl = CL / N ≈ 2 / 8 ≈ 0.25

Метрика граничных значений

Границы для случайных чисел в списке: от -100 до 100. Проверка показала корректную обработку всех значений.

Метрика Чепина

Группы переменных:
- P (параметры): 1 (входной список).
- M (модифицируемые): 2 (списки list и negativeList).
- C (управляющие): 1 (условие integer < 0).
- T (временные): 0.

Метрика Чепина: Q = P + 2M + 3C + 0.5T = 1 + 4 + 3 = 8.

Спен программы

Список переменных и их использование:
- list: 3 раза (создание, добавление, возврат).
- negativeList: 4 раза (создание, добавление, проверка, возврат).
- integer: 2 раза (итерация, проверка условия).

Итоговый спен: 3 + 4 + 2 = 9.

Заключение

Рассчитанные метрики демонстрируют умеренную сложность представленного алгоритма. Использование отрицательных значений и дополнительных списков увеличивает сложность в сравнении с базовыми линейными алгоритмами. Выбранный подход упрощает анализ, но допускает оптимизации для повышения эффективности.