

Для оценки сложности программы с помощью метрик Холстеда необходимо рассчитать следующие базовые метрики:

1. Длина программы (N) - общее количество операторов и операндов в программе.
2. Словарь программы (n) - количество уникальных операторов и операндов в программе.
3. Длина программы без дублирования (\tilde{N}) - сумма количества уникальных операторов и операндов в программе.
4. Объем программы (V) - произведение длины программы без дублирования (\tilde{N}) на логарифм основания 2 от словаря программы (n).

Давайте рассчитаем эти метрики для данной программы.

Анализ кода:

В данной программе имеется два класса: Main и Fraction. Поэтому мы должны анализировать оба класса для подсчета метрик Холстеда.

Класс Main:

В классе Main есть один метод main, который содержит последовательность операторов.

Операторы в методе main:

1. System.out.println - 14 раз (12 раз в различных формах и 2 раза в комментариях).
2. Создание объектов класса Fraction - 5 раз.
3. Вызов методов класса Fraction - 8 раз.
4. Цикл for - 2 раза.
5. Вызов метода Arrays.sort - 2 раза.

Таким образом, общее количество операторов в методе main составляет 31.

Класс Fraction:

В классе Fraction есть несколько методов и переменных.

Операторы в классе Fraction:

1. Операторы конструктора Fraction - 17 операторов.
2. Операторы метода validateFraction - 6 операторов.
3. Операторы метода parseFraction - 10 операторов.
4. Операторы метода GCD - 8 операторов.
5. Операторы метода add - 6 операторов.
6. Операторы метода subtract - 8 операторов.
7. Операторы метода multiply - 4 оператора.
8. Операторы метода divide - 4 оператора.

9. Операторы метода compareTo - 7 операторов.
10. Операторы метода iterator - 5 операторов.
11. Операторы метода reset - 2 оператора.
12. Операторы метода hasNext - 2 оператора.
13. Операторы метода remove - 1 оператор.
14. Операторы метода next - 7 операторов.
15. Операторы метода toString - 4 оператора.

Таким образом, общее количество операторов в классе Fraction составляет 93.

Расчет метрик:

1. Длина программы (N) - сумма операторов в классе Main и классе Fraction, то есть $31 + 93 = 124$.
2. Словарь программы (n) - количество уникальных операторов и операндов в программе. Для подсчета словаря программы, нужно найти все различные операторы и операнды в классах Main и Fraction. После анализа кода, были обнаружены следующие уникальные операторы и операнды:
 - System, out, println, Hard, code, constructor, Fraction, f1, toString, parseFraction, -15, -9, add, subtract, multiply, divide, foreach, i, fractions, setSortBy, compareTo, entry, numerator, denominator, `Продолжение расчета метрик:
 - GCD, validateFraction, iterator, hasNext, remove, next, reset, sort, sortFractions, int, i, n, j, gcd, temp, fraction, str, sb, valueOf, if, else, new, return, catch, Exception, e, Arrays, sort, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33.

Таким образом, количество уникальных операторов и операндов (словарь программы, n) равно 63.

3. Длина программы без дублирования (\tilde{N}) - сумма количества уникальных операторов и операндов в программе. $\tilde{N} = 124 + 63 = 187$.
4. Объем программы (V) - произведение длины программы без дублирования (\tilde{N}) на логарифм основания 2 от словаря программы (n). $V = 187 * \log_2(63) \approx 187 * 5.977 = 1118.399$.

Таким образом, для данной программы метрики Холстеда выглядят следующим образом:

- Длина программы (N): 124
- Словарь программы (n): 63
- Длина программы без дублирования (\tilde{N}): 187
- Объем программы (V): 1118.399

Эти метрики могут использоваться для оценки сложности программы и прогнозирования затрат на разработку и тестирование.