МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе №1

на тему: «Расчет метрических характеристик качества

разработки программ по метрикам Холстеда»

по дисциплине: «Качество и тестирование программного обеспечения»

Вариант 7

Выполнила: Карпикова С.П.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71ПГ

Проверили: Ужаринский А.Ю., Конюхова О.В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Орел, 2020 г.

**Тема:** расчет метрических характеристик качества разработки программ по метрикам Холстеда.

**Задание:** разработать алгоритм решения задания в соответствии с вариантом. Реализовать разработанный алгоритм на трёх языках программирования. Для каждой реализации рассчитать метрики качества программы Холстеда: словарь программы, длина реализации, объём программы, потенциальный объём программы, уровень качества программирования, сложность понимания, трудоёмкость кодирования, время кодирования, информационное содержание программы, уровень языка.

**Цель работы:** получение навыков расчета показателей качества программ.

**Вариант:** составить алгоритм и программу, которые в числовые переменные вводят два натуральных числа, являющиеся числителем и знаменателем простой дроби, и получают новые значения этих чисел, сократив дробь до несократимой дроби. Предусмотреть проверку ввода исходных данных.

**Алгоритм**

1. Найти наибольший общий делитель с помощью алгоритма Евклида, который можно описать следующей формулой:

 {\rm gcd}(a,b) = \cases{ a, & {\rm if} b=0 \cr {\[...]

1. Каждую из переменных разделить на наибольший общий делитель.

**Ход работы**

**Программа на С++:**

#include "stdafx.h"

#include <iostream>

using namespace std;

void ReduceFraction(int a, int b) {

int v = a;

int g = b;

int c;

while (v != 0 && g != 0) {

if (v > g) {

v = v % v;

}

else {

g = g % v;

}

}

if (a == 0) {

c = v;

}

else {

c = g;

}

cout << a / c << " " << b / c << endl;

}

int main()

{

int aa, bb;

cin >> aa >> bb;

if (aa > 0 && bb > 0) {

ReduceFraction(aa, bb);

} else {

cout << "Error";

}

system("pause");

return 0;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | = | 6 | 1 | a | 4 |
| 2 | != | 2 | 2 | b | 3 |
| 3 | && | 2 | 3 | v | 8 |
| 4 | while | 1 | 4 | g | 6 |
| 5 | if | 3 | 5 | c | 5 |
| 6 | else | 3 | 6 | aa | 4 |
| 7 | > | 3 | 7 | bb | 4 |
| 8 | % | 2 | 8 | 0 | 5 |
| 9 | == | 1 | 9 | " " | 1 |
| 10 | / | 2 | 10 | "Error" | 1 |
| 11 | cin | 1 | 11 | "pause" | 1 |
| 12 | cout | 2 |  |  |  |
| 13 | system | 1 |  |  |  |
| 14 | endl | 1 |  |  |  |
| 15 | return | 1 |  |  |  |
| 16 | { | 7 |  |  |  |
| 17 | } | 7 |  |  |  |
| 18 | ( | 5 |  |  |  |
| 19 | ) | 5 |  |  |  |
| 20 | int | 6 |  |  |  |

**Расчет метрик**

Число уникальных операторов η1 = 20

Общее число всех операторов N1 = 61

Число уникальных операндов η2 = 11

Общее число всех операндов N2 = 42

Словарь программы η = 31

Длина программы N = 103

Объем программы V = 103 \* log231 = 510.262

Теоретическая оценка длины N` = 20 \* log220 + 11 \* log211 = 124.489

Потенциальный объем программы V` = 124.489 \* log2(4 + 2) = 321.804

Уровень качества программирования L = 321.804/ 510.262 = 0,63

L` = 2 \* 11 / 20 \* 42 = 0,026

Сложность понимания EC = 510.262 / 0.026 \* 2 = 9812.73

Трудоемкость кодирования D = 1 / 0.026 = 38.46

Время кодирования Т = 9812.73 / 18 = 545.152

Информационное содержание программы I = 510.262 / 38.46 = 13.267

Уровень языка

**Программа на Python:**

def ReduceFraction(a, b):

v = a

g = b

while v != 0 and g != 0:

if v > g:

v = v % v

else:

g = g % v

if a == 0:

c = v

print(a // c, b // c)

else:

c = g

print(a // c, b // c)

aa = int(input())

bb = int(input())

if aa > 0 and bb > 0:

ReduceFraction(aa, bb)

else:

print("error")

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | = | 8 | 1 | a | 5 |
| 2 | != | 2 | 2 | b | 4 |
| 3 | > | 3 | 3 | c | 4 |
| 4 | % | 2 | 4 | v | 8 |
| 5 | // | 4 | 5 | g | 6 |
| 6 | == | 1 | 6 | aa | 3 |
| 7 | if | 3 | 7 | bb | 3 |
| 8 | else | 3 | 8 | 0 | 5 |
| 9 | while | 1 | 9 | "error" | 1 |
| 10 | int | 2 |  |  |  |
| 11 | input | 2 |  |  |  |
| 12 | print | 3 |  |  |  |

**Расчет метрик**

Число уникальных операторов η1 = 12

Общее число всех операторов N1 = 35

Число уникальных операндов η2 = 9

Общее число всех операндов N2 = 39

Словарь программы η = 21

Длина программы N = 74

Объем программы V = 74 \* log221 = 325.008

Теоретическая оценка длины N` = 12 \* log212 + 9 \* log29 = 71.55

Потенциальный объем программы V` = 71.55 \* log2(4 + 2) = 184.957

Уровень качества программирования L = 184.957/ 325.008 = 0,57

L` = 2 \* 9 / 12 \* 39 = 0,038

Сложность понимания EC = 325.008 / 0,038 \* 2 = 4276.421

Трудоемкость кодирования D = 1 / 0.038 = 26.316

Время кодирования Т = 4276.421 / 18 = 237.579

Информационное содержание программы I = 325.008 / 26.316 = 12.35

Уровень языка

**Программа на Java:**

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class Reduce

{

public static void ReduceFraction(int a, int b) {

int v = a;

int g = b;

int c;

while (v != 0 && g != 0) {

if (v > g) {

v = v % v;

}

else {

g = g % v;

}

}

if (a == 0) {

c = v;

}

else {

c = g;

}

System.out.println(a / c + " " + b / c);

}

public static void main (String[] args) throws java.lang.Exception

{

int aa, bb;

Scanner s = new Scanner(System.in);

aa = s.nextInt();

bb = s.nextInt();

if (aa > 0 && bb > 0) {

ReduceFraction(aa, bb);

} else {

System.out.println("error");

}

}

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | = | 9 | 1 | a | 4 |
| 2 | != | 2 | 2 | b | 3 |
| 3 | && | 2 | 3 | v | 8 |
| 4 | while | 1 | 4 | g | 6 |
| 5 | if | 3 | 5 | c | 5 |
| 6 | else | 3 | 6 | aa | 4 |
| 7 | > | 3 | 7 | bb | 4 |
| 8 | % | 2 | 8 | 0 | 5 |
| 9 | == | 1 | 9 | " " | 1 |
| 10 | / | 2 | 10 | "error" | 1 |
| 11 | nextInt() | 2 |  |  |  |
| 12 | System.out.println | 2 |  |  |  |
| 13 | Scanner | 1 |  |  |  |
| 14 | { | 7 |  |  |  |
| 15 | } | 7 |  |  |  |
| 16 | ( | 5 |  |  |  |
| 17 | ) | 5 |  |  |  |
| 18 | int | 6 |  |  |  |

**Расчет метрик**

Число уникальных операторов η1 = 18

Общее число всех операторов N1 = 63

Число уникальных операндов η2 = 10

Общее число всех операндов N2 = 41

Словарь программы η = 28

Длина программы N = 104

Объем программы V = 104 \* log228 = 499.928

Теоретическая оценка длины N` = 18 \* log218 + 10 \* log210 = 108.28

Потенциальный объем программы V` = 108.28 \* log2(4 + 2) = 279.903

Уровень качества программирования L = 279.903/ 499.928 = 0,56

L` = 2 \* 10 / 18 \* 41 = 0,027

Сложность понимания EC = 499.928/ 0,027 \* 2 = 9257.925

Трудоемкость кодирования D = 1 / 0.027 = 37.037

Время кодирования Т = 9257.925 / 18 = 514.329

Информационное содержание программы I = 499.928 / 37.037= 13.498

Уровень языка

**Выводы:** В ходе выполнения работы были разработаны, модифицированы программы реализации алгоритма на языках программирования Java, С++, Python. Для созданных программы были оценены метрические характеристики по Холстеду.