МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФГБОУ ВО «ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ И.С.ТУРГЕНЕВА»

Кафедра «Информационные системы и технологии»

**Отчет**

По лабораторной работе №1

На тему: «Pасчет метрических характеристик качества разработки программ  по метрикам Холстеда»

По дисциплине: «Требования к программному обеспечению и спецификации»

Работу выполнила: Куксенко Д. С.

Институт приборостроения, автоматизации и информационных технологий

Направление/специальность: 09.03.04 «Программная инженерия»

Группа: 22ПГ

Преподаватель Олькина Е.В.

Отчет защищен с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Орёл, 2024

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

**Цель работы: п**олучение навыков расчета показателей качества программ.

**Вариант №8:**

Составить алгоритм и программу, которые выводят в строку все натуральные простые числа, не превосходящие заданное натуральное трехзначное число М, а также находят количество этих чисел и их сумму.

**Алгоритм:**

1. Вводим число M
2. Проверяем, что введенное число является трехначным (принадлежит диапазону от 100 до 999 включительно), если да, переходим к п.3, иначе завершаем работу программы.
3. Инициализируем список numbers, используемый для хранения найденныйх простых чисел.
4. Проходим по всем числам x от 2 до M (включительно):
5. Для каждого числа x устанавливаем флаг flag\_is\_simple, который изначально равен True. Этот флаг будет указывать, является ли число простым.
6. Проверяем делимость числа x на все числа от 2 до квадратного корня из x (включительно). Если x делится на какое-либо из этих чисел, флаг flag\_is\_simple устанавливается в False, и происходит выход из внутреннего цикла.
7. Если после проверки флаг остался равным True, число x добавляется в список numbers.
8. Выводим результаты: найденный список numbers, его длину и сумму вошедших в него элементов.

**Программа на Python:**  
try:  
 M = int(input())  
 if 100 <= M <= 999:  
 numbers = []  
 for x in range(2, M + 1):  
 flag\_is\_simple = True  
 for i in range(2, int(x \*\* 0.5) + 1):  
 if x % i == 0:  
 flag\_is\_simple = False  
 break  
 if flag\_is\_simple:  
 numbers.append(x)  
 print(numbers)  
 print(len(numbers))  
 print(sum(numbers))  
except Exception as e:  
 print(e)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | try | 1 | 1 | numbers | 5 |
| 2 | except | 1 | 2 | 0 | 1 |
| 3 | input | 1 | 3 | 2 | 2 |
| 4 | <= | 2 | 4 | 0.5 | 1 |
| 5 | in | 2 | 5 | 100 | 1 |
| 6 | range | 2 | 6 | 999 | 1 |
| 7 | \*\* | 1 | 7 | M | 3 |
| 8 | % | 1 | 8 | i | 2 |
| 9 | = | 4 | 9 | x | 4 |
| 10 | == | 1 | 10 | 1 | 1 |
| 11 | print | 4 | 11 | e | 2 |
| 12 | break | 1 | 12 | flag\_is\_simple | 3 |
| 13 | len | 1 | 13 |  |  |
| 14 | sum | 1 | 14 |  |  |
| 15 | append | 1 | 15 |  |  |
| 16 | + | 1 | 16 |  |  |
| 17 | True | 1 | 17 |  |  |
| 18 | False | 1 | 18 |  |  |

**Расчет метрик для Python:**

Число уникальных операторов η1 = 18

Общее число всех операторов N1 = 27

Число уникальных операндов η2 = 12

Общее число всех операндов N2 = 26

Словарь программы η = 30

Длина программы N' = 53

Теоретическая оценка длины  N' = 118

Объём программы V = 260

Потенциальный объём программы V' =555

Уровень качества программирования:

L = 2,13

L' = 0,05

Сложность понимания EC = 2535,64

Трудоёмкость кодирования D = 19,50

Время кодирования T = 140,87

Информационное содержание программы I = 13,34

Уровень языка b = 1184,50

**Программа на Java:**

package system\_requirments\_lab\_1;  
  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class Program {  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.in);  
 int M;  
 try {  
 M = in.nextInt();  
 List<Integer> numbers = new ArrayList<>();  
 if (100 <= M && M <= 999) {  
 for (int x = 2; x <= M; x++) {  
 boolean flag\_is\_simple = true;  
 for (int i = 2; i <= (int)Math.sqrt(x); i++) {  
 if (x % i == 0) {  
 flag\_is\_simple = false;  
 break;  
 }  
 }  
 if (flag\_is\_simple)  
 numbers.add(x);  
 }  
 }  
 System.out.println(numbers);  
 System.out.println(numbers.size());  
 System.out.println(numbers.stream().mapToInt(Integer::intValue).sum());  
 }  
 catch (Exception e){  
 System.out.println(e);  
 }  
 }  
}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | if | 3 | 1 | 2 | 2 |
| 2 | <= | 4 | 2 | 0 | 1 |
| 3 | for | 2 | 3 | 100 | 1 |
| 4 | int | 1 | 4 | 999 | 1 |
| 5 | nextInt | 1 | 5 | i | 3 |
| 6 | Math.sqrt | 1 | 6 | in | 2 |
| 7 | % | 1 | 7 | M | 5 |
| 8 | == | 1 | 8 | numbers | 5 |
| 9 | try | 1 | 9 | x | 6 |
| 10 | catch | 1 | 10 | e | 2 |
| 11 | = | 7 | 11 | flag\_is\_simple | 3 |
| 12 | new | 1 | 12 |  |  |
| 13 | && | 1 | 13 |  |  |
| 14 | add | 1 | 14 |  |  |
| 15 | System.out.println() | 4 | 15 |  |  |
| 16 | stream() | 1 | 16 |  |  |
| 17 | mapToInt() | 1 | 17 |  |  |
| 18 | sum() | 1 | 18 |  |  |
| 19 | ++ | 2 | 19 |  |  |
| 20 | false | 1 | 20 |  |  |
| 21 | true | 1 | 21 |  |  |

**Расчет метрик для Java:**

Число уникальных операторов η1 = 21

Общее число всех операторов N1 = 37

Число уникальных операндов η2 = 11

Общее число всех операндов N2 = 31

Словарь программы η = 32

Длина программы N' = 68

Теоретическая оценка длины  N' = 130,29

Объём программы V = 340,00

Потенциальный объём программы V' = 645,49

Уровень качества программирования:

L = 1,90

L' = 0,03

Сложность понимания EC = 5030,45

Трудоёмкость кодирования D = 29,59

Время кодирования T = 279,47

Информационное содержание программы I = 11,49

Уровень языка b = 1225,48

**Программа на C++:**

#include <iostream>

#include <cmath>

int main()

{

int M;

std::cin >> M;

if (M >= 100 && M <= 999)

{

int\* numbers = new int[M];

int count = 0;

for (int x = 2; x <= M; x++)

{

bool flag\_is\_simple = true;

for (int i = 2; i <= sqrt(x); i++)

{

if (x % i == 0)

{

flag\_is\_simple = false;

break;

}

}

if (flag\_is\_simple) {

numbers[count] = x;

count++;

}

}

for (int i = 0; i < count; i++)

{

std::cout << numbers[i];

if (i < count - 1) std::cout << ", ";

}

std::cout << std::endl << count;

int sum = 0;

for (int i = 0; i < count; i++)

{

sum += numbers[i];

}

std::cout << std::endl << sum;

delete[] numbers;

};

return 0;

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Операторы | | | Операнды | | |
| N | Оператор | Число вхождений | N | Операнд | Число вхождений |
| 1 | if | 4 | 1 | 1 | 1 |
| 2 | <= | 3 | 2 | 2 | 2 |
| 3 | return | 1 | 3 | 100 | 1 |
| 4 | for | 4 | 4 | 999 | 1 |
| 5 | sqrt() | 1 | 5 | 0 | 5 |
| 6 | ++ | 5 | 6 | M | 6 |
| 7 | % | 1 | 7 | x | 6 |
| 8 | == | 1 | 8 | i | 8 |
| 9 | >= | 1 | 9 | count | 7 |
| 10 | && | 1 | 10 | sum | 3 |
| 11 | = | 8 | 11 | numbers | 5 |
| 12 | new | 1 | 12 | flag\_is\_simple | 3 |
| 13 | [] | 4 | 13 | 0 | 5 |
| 14 | < | 3 | 14 | int[] | 1 |
| 15 | += | 1 | 15 |  |  |
| 16 | delete | 1 | 16 |  |  |
| 17 | - | 1 | 17 |  |  |
| 18 | std::cin | 1 | 18 |  |  |
| 19 | std::cout | 4 | 19 |  |  |
| 20 | true | 1 | 20 |  |  |
| 21 | false | 1 | 21 |  |  |

**Расчет метрик для C++:**

Число уникальных операторов η1 = 21

Общее число всех операторов N1 = 48

Число уникальных операндов η2 = 14

Общее число всех операндов N2 = 54

Словарь программы η = 35

Длина программы N' = 102

Теоретическая оценка длины  N' = 145,54

Объём программы V = 523,19

Потенциальный объём программы V' = 837,58

Уровень качества программирования:

L = 1,60

L' = 0,02

Сложность понимания EC = 10594,53

Трудоёмкость кодирования D = 40,50

Время кодирования T = 588,59

Информационное содержание программы I = 12,92

Уровень языка b = 1340,88