Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра программного обеспечения и информационных технологий

Дисциплина «Метрология, стандартизация и сертификация»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«Метрики размера программ»**

БГУИР 6-05-0612-02

|  |
| --- |
| Выполнили студенты группы 353503  КОХАН Артём Игоревич  ШЕМЕТКОВ Ян Игоревич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. ПОИТ  БОЛТАК Светлана Владимировна |
|  |
| (дата, подпись преподавателя) |

Минск 2025

# 1 Индивидуальное задание

**Зaдание**

1 Изучить метрики размера программ на примере метрики Холстеда.

2 Взять код программы в консольном режиме (например, найти в Интернет) на указанном преподавателем языке программирования.

3 Написать свою программу (на любом языке программирования не в консоли), которая будет анализировать программу из пункта 2. На основе анализа программа должна рассчитать метрики Холстеда согласно примеру в файле из пункта 1. Результат работы программы: вывести на экран 6 основных метрик Холстеда и 3 расширенные в удобном для просмотра виде (например, в таблицу).

# 2 Выполнение работы

Ниже приведён код для расчёта метрик Холстеда

import kotlin.math.PI

import kotlin.math.abs

import kotlin.math.cos

import kotlin.math.ln

import kotlin.math.log

import kotlin.math.sin

import kotlin.math.sqrt

fun testFun(isTrue: Boolean) {

var a = 5

when(isTrue){

true -> {

a++

}

false -> {

a--

}

}

}

fun main() {

val a = 10; val b = 3

val c = 5.5

val d = 2.0

val sum = a + b

val difference = a - b

val product = a \* b

val quotient = a / b

val remainder = a % b

val sumDouble = c + d

val differenceDouble = c - d

val productDouble = c \* d

val quotientDouble = c / d

val andResult = a.and(b)

val orResult = a.or(b)

val xorResult = a.xor(b)

val shiftLeft = a.shl(2)

val shiftRight = a.shr(1)

val unsignedShiftRight = a.ushr(1)

var x = 20

x += 5

x -= 2

x \*= 3

x /= 2

x %= 4

var y = 7

val preIncrement = ++y

val postIncrement = y++

val preDecrement = --y

val postDecrement = y--

val negation = -a

val absoluteValue = abs(-c)

val sqrtValue = sqrt(16.0)

val logValue = ln(10.0)

val sinValue = sin(PI / 2)

val cosValue = cos(0.0)

if(sqrtValue != logValue){

print(sqrtValue)

}

}

object A {

val a: String = "Simple class"

fun func(something: String) {

repeat(5){

println(a + something)

}

}

}

Метрики рассчитанные вручную

|  |  |  |  |
| --- | --- | --- | --- |
| Оператор | Количество вхождений | Операнд | Количество вхождений |
| = | 33 | isTrue | 2 |
| ++ | 3 | a | 12 |
| -- | 3 | 5 | 3 |
| + | 3 | true | 1 |
| - | 4 | false | 1 |
| \* | 2 | 10 | 1 |
| / | 3 | b | 9 |
| % | 1 | 3 | 2 |
| += | 1 | c | 6 |
| -= | 1 | 5.5 | 1 |
| \*= | 1 | d | 5 |
| /= | 1 | 2.0 | 1 |
| %= | 1 | sum | 1 |
| != | 1 | difference | 1 |
| When | 1 | product | 1 |
| If | 1 | quotient | 1 |
| repeat | 1 | rmainder | 1 |
| ; | 1 | smDouble | 1 |
| testFun(isTrue: Boolean) | 1 | differenceDouble | 1 |
| or(b) | 1 | productDouble | 1 |
| xor(b) | 1 | quotientDouble | 1 |
| shl(2) | 1 | andResult | 1 |
| shr(1) | 1 | orResult | 1 |
| ushr(1) | 1 | xorResult | 1 |
| sqrt(16.0) | 1 | shiftLeft | 1 |
| ln(10.0) | 1 | 2 | 4 |
| sin(PI / 2) | 1 | shiftRight | 1 |
| cos(0.0) | 1 | 1 | 2 |
| print(sqrtValue) | 1 | unsignedShiftRight | 1 |
| println(a + something) | 1 | x | 6 |
| { | 9 | 20 | 1 |
| } | 9 | 4 | 1 |
|  |  | y | 5 |
|  |  | 7 | 1 |
|  |  | preIncrement | 1 |
|  |  | postIncrement | 1 |
|  |  | preDecrement | 1 |
|  |  | postDectement | 1 |
|  |  | negation | 1 |
|  |  | absoluteValue | 1 |
|  |  | sqrtValue | 3 |
|  |  | 16.0 | 1 |
|  |  | logValue | 2 |
|  |  | 10.0 | 1 |
|  |  | sinValue | 1 |
|  |  | PI | 1 |
|  |  | cosValue | 1 |
|  |  | 0.0 | 1 |
|  |  | A | 1 |
|  |  | something | 2 |
|  |  | “Simple class” | 1 |

Общее число операторов программы (N1): 92

Общее число операнд в программе (N2): 100

Длина программы (N = N1 + N2): 192

Число уникальных операторов программы (ɳ1): 32

Число уникальных операнд программы(ɳ2): 51

Cловарь программы(ɳ): 83

Объем программы(V = NLog2 ɳ): 1224.0076

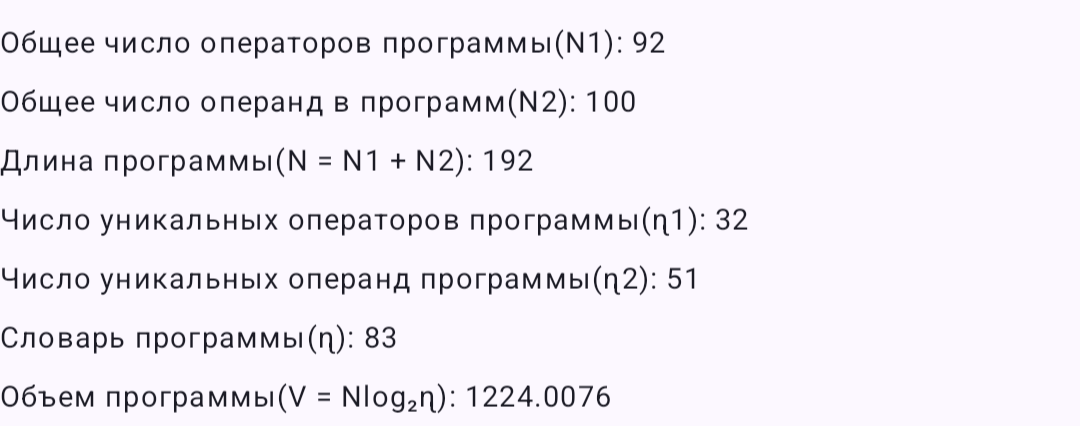


Рисунок 1 – Работа программы

# Вывод

В ходе лабораторной работы был проанализирован исходный текст программы и изучены базовые и расширенные метрики Холстеда размера программ на примере языка программирования Kotlin.