Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования «Сибирский государственный университет

телекоммуникаций и информатики» (СибГУТИ)

Кафедра прикладной математики и кибернетики

Лабораторная работа

«Порядок расчёта метрических характеристик ПО»

Выполнил:

Студент группы ИП-017

Костин А.В.

Работу проверил:

ассистент кафедры ПМиК

Агалаков А.А.

Новосибирск 2023 г.

Оглавление

[1. Задание 3](#_Toc153310543)

[2. Результат 4](#_Toc153310544)

[3. Исходные тексты программ на языке Python 6](#_Toc153310545)

[4. Исходные тексты программ на языке C# 8](#_Toc153310546)

1. **Задание**

Приобретение практических навыков расчета метрических характеристик

ПС: трудоемкости реализации, начальной надежности, структурных

параметров на основе постановки задачи.

Задание

1. Написать программу на двух языках программирования для расчета следующих метрических характеристик ПС:

* структурных параметров ПС:
  + числа уровней иерархии в схеме иерархии логических модулей;
  + количества модулей на каждом уровне иерархии;
  + общего числа модулей в ПС;
* календарного времени программирования;
* начальной надежности ПС.

2. На основе постановки задачи рассчитать метрические характеристики программы.

3. Сопоставить расчетные метрические характеристики с характеристиками, полученной в результате реализации программы.

4. С помощью написанной программы рассчитать метрические характеристики для следующих значений η\*2: 300, 400, 512. При расчете

начального количества ошибок принять г = 0.5 Tk. При расчете

календарного времени принять число программистов n = 5, число

отлаженных в день команд ассемблера v = 20.

1. **Результат**

В ходе работы был разработан класс, в котором были реализованы методы для расчета метрических характеристик ПО. Для сравнения и анализа метрических характеристик был также написан код на языке Python с реализацией таких же алгоритмов.

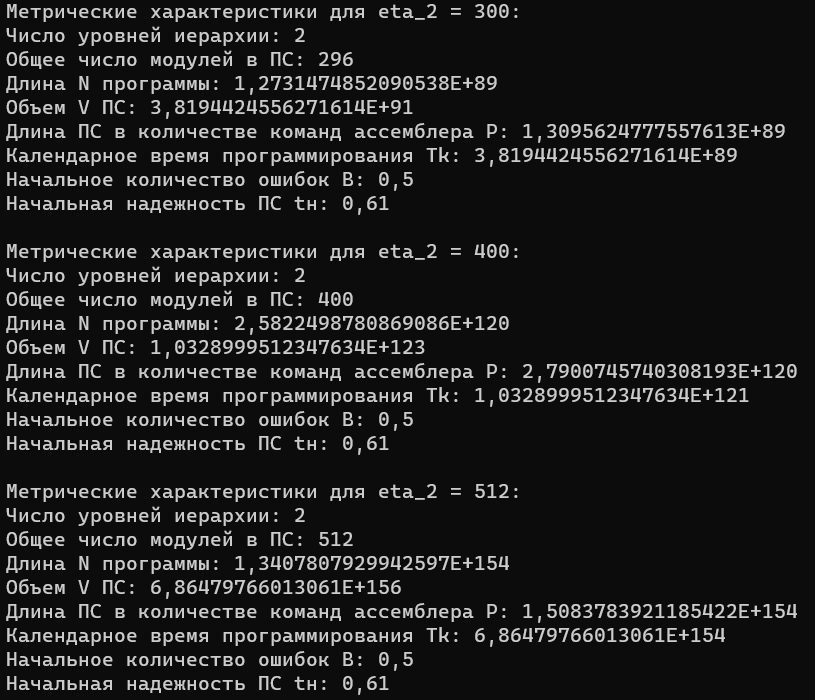


Рисунок 1. Результат работы на С#

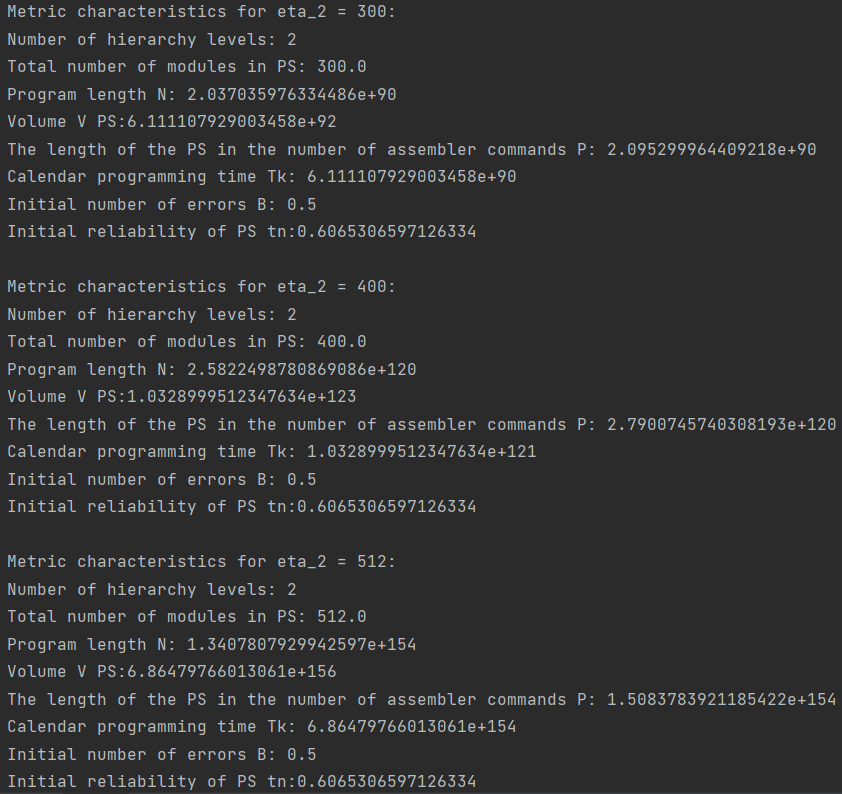


Рисунок 2. Результат работы на Python

**3. Исходные тексты программ на языке Python**

import math  
  
  
def calculate\_metrics(eta\_2):  
 k = eta\_2 / 8  
 i = 0  
 total\_modules = 0  
  
 if k > 8:  
 i = 1  
 while k > 8:  
 k /= 8  
 i += 1  
 total\_modules = k \* (8 \*\* i)  
 else:  
 total\_modules = eta\_2  
  
 N = 2 \*\* total\_modules  
 V = N \* eta\_2  
  
 k\_2 = eta\_2  
 P = N \* (math.log(k\_2, 2) / 8)  
  
 n = 5  
 v = 20  
  
 Tk = V / (n \* v)  
 tau = 0.5 \* Tk  
 B = tau / Tk  
  
 t\_n = math.exp(-B)  
  
 return i, total\_modules, N, V, P, Tk, B, t\_n  
  
  
def run():  
 values\_of\_eta\_2 = [300, 400, 512]  
  
 for eta\_2 in values\_of\_eta\_2:  
 i, total\_modules, N, V, P, Tk, B, t\_n = calculate\_metrics(eta\_2)  
 print(f"Metric characteristics for eta\_2 = {eta\_2}:")  
 print(f"Number of hierarchy levels: {i}")  
 print(f"Total number of modules in PS: {total\_modules}")  
 print(f"Program length N: {N}")  
 print(f"Volume V PS:{V}")  
 print(f"The length of the PS in the number of assembler commands P: {P}")  
 print(f"Calendar programming time Tk: {Tk}")  
 print(f"Initial number of errors B: {B}")  
 print(f"Initial reliability of PS tn:{t\_n}")  
 print("")

**4. Исходные тексты программ на языке C#**

Program.cs

using System;

using System.Linq;

using System.Collections.Generic;

namespace lab17

{

class Program

{

static void Main()

{

int[] valuesOfEta2 = { 300, 400, 512 };

int n = 5; // Число программистов

int v = 20; // Число отлаженных в день команд ассемблера

foreach (int eta\_2 in valuesOfEta2)

{

int i = 0;

double totalModules = 0;

double k = eta\_2 / 8;

if (k > 8)

{

i = 1;

while (k > 8)

{

k /= 8;

i++;

}

totalModules = k \* Math.Pow(8, i);

}

else

{

totalModules = eta\_2;

}

double N = Math.Pow(2, totalModules);

double V = N \* eta\_2;

double P = N \* Math.Log(eta\_2, 2) / 8;

double Tk = V / (n \* v);

double tau = 0.5 \* Tk;

double B = tau / Tk;

double t\_n = Math.Exp(-B);

Console.WriteLine($"Метрические характеристики для eta\_2 = {eta\_2}:");

Console.WriteLine($"Число уровней иерархии: {i}");

Console.WriteLine($"Общее число модулей в ПС: {Math.Round(totalModules,2)}");

Console.WriteLine($"Длина N программы: {Math.Round(N,2)}");

Console.WriteLine($"Объем V ПС: {Math.Round(V,2)}");

Console.WriteLine($"Длина ПС в количестве команд ассемблера P: {Math.Round(P,2)}");

Console.WriteLine($"Календарное время программирования Tk: {Math.Round(Tk,2)}");

Console.WriteLine($"Начальное количество ошибок B: {Math.Round(B,2)}");

Console.WriteLine($"Начальная надежность ПС tн: {Math.Round(t\_n,2)}");

Console.WriteLine();

}

}

}

}