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

Современные технологии программирования

Практическая работа №13

«Порядок расчета метрических характеристик ПС»

Выполнил: студент 4 курса

группы ИП-111 Кузьменок Денис Витальевич

Проверил преподаватель: Зайцев Михаил Георгиевич

Новосибирск, 2024 г.

**Цель:**

Приобретение практических навыков расчета метрических характеристик ПС: трудоемкости реализации, начальной надежности, структурных параметров на основе постановки задачи.

**Задание**

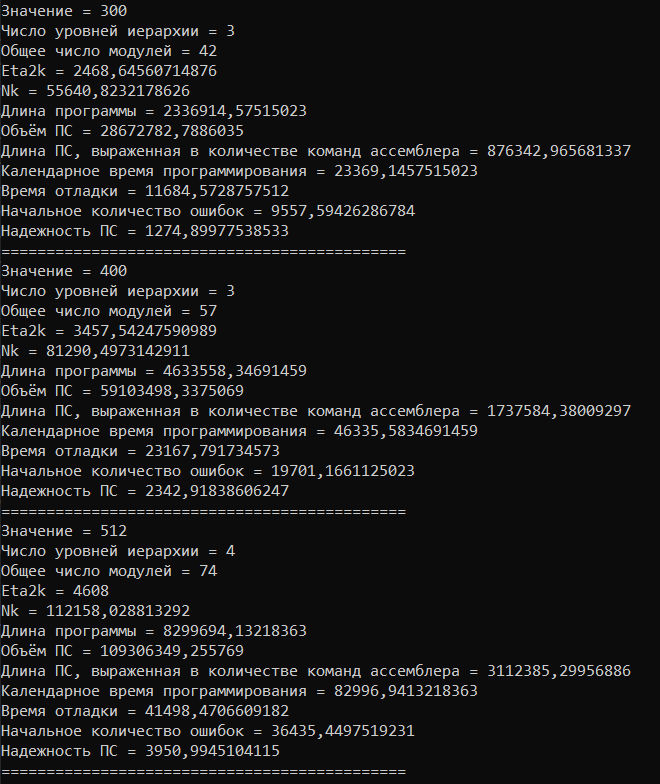
1. Написать программу на двух языках программирования для расчета следующих метрических характеристик ПС:

* 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.

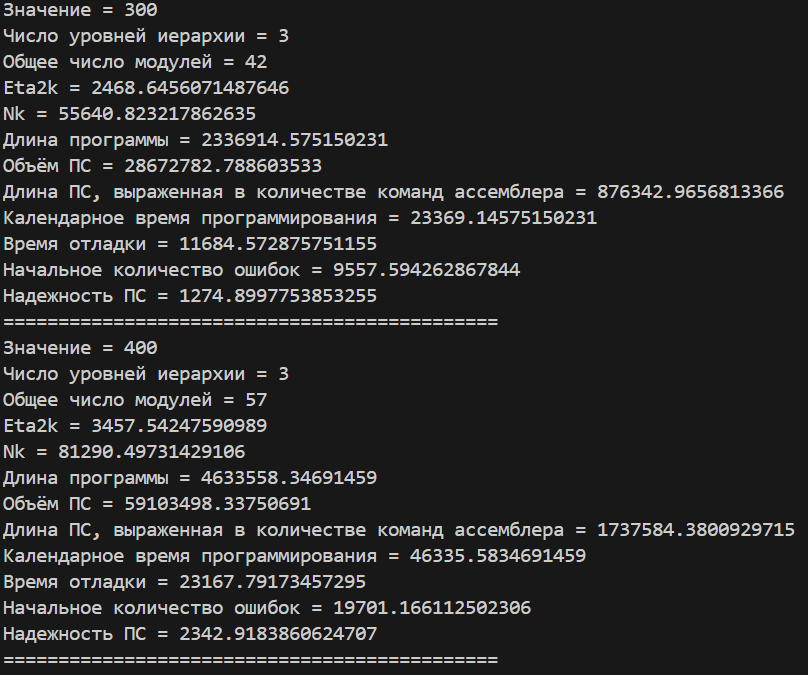
1. На основе постановки задачи рассчитать метрические характеристики программы.
2. Сопоставить расчетные метрические характеристики с характеристиками, полученной в результате реализации программы.
3. С помощью написанной программы рассчитать метрические характеристики для следующих значений **η\* 2: 300, 400, 512**. При расчете начального количества ошибок принять **τ = 0.5 Tk**. При расчете календарного времени принять число программистов **n = 5**, число отлаженных в день команд ассемблера **v = 20**.

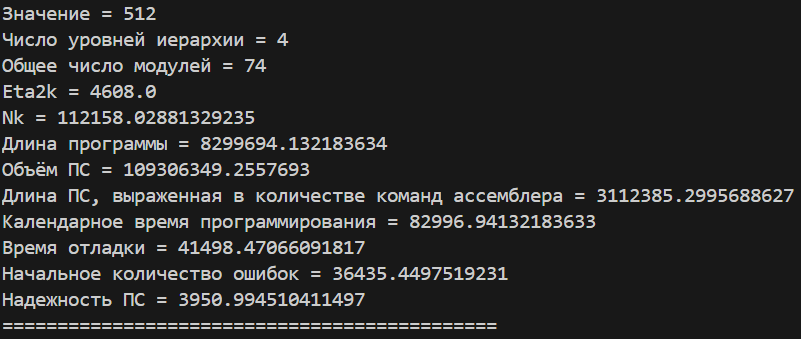
**Ход работы**

Программа на языке C# показала следующие результаты:



Программа на языке Python показала следующие результаты:





**Вывод:**

В ходе выполнения лабораторной работы были рассчитаны основные метрические характеристики программного обеспечения (ПС), включая структурные параметры, календарное время программирования и начальную надежность системы.

**Листинг программ**

**Program.cs**

**using** **System**;

**using** **System.Collections.Generic**;

**using** **System.Linq**;

**using** **System.Text**;

**using** **System.Threading.Tasks**;

**namespace** **lab13**

{

**class** **Program**

{

**private** **static** **void** **Main**(**string**[] args)

{

List<**int**> etas = **new** List<**int**>()

{

**300**,

**400**,

**512**

};

**foreach**(**int** element **in** etas)

{

**int** i = I(element);

**int** k = K(element, i);

**double** eta2k = Eta2k(element);

**double** nk = Nk(eta2k);

**double** n = N(k, nk);

**double** v = V(k, nk, eta2k);

**double** p = P(n);

**double** tk = Tk(n);

**double** t = T(tk);

**double** b0 = B0(v);

**double** tn = Tn(b0, t);

Console.WriteLine($"Значение = {element}");

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

Console.WriteLine($"Общее число модулей = {k}");

Console.WriteLine($"Eta2k = {eta2k}");

Console.WriteLine($"Nk = {nk}");

Console.WriteLine($"Длина программы = {n}");

Console.WriteLine($"Объём ПС = {v}");

Console.WriteLine($"Длина ПС, выраженная в количестве команд ассемблера = {p}");

Console.WriteLine($"Календарное время программирования = {tk}");

Console.WriteLine($"Время отладки = {t}");

Console.WriteLine($"Начальное количество ошибок = {b0}");

Console.WriteLine($"Надежность ПС = {tn}");

Console.WriteLine("=============================================");

}

}

**private** **static** **int** **I**(**int** eta)

{

**return** (**int**)(Math.Log(eta, **2**) / **3** + **1**);

}

**private** **static** **int** **K**(**int** eta, **int** i)

{

**int** total = **1**;

**for**(**int** j = **1**; j < i; j++)

{

total += (**int**)(eta / Math.Pow(**8**, j));

}

**return** total;

}

**private** **static** **double** **N**(**int** k, **double** nk)

{

**return** k \* nk;

}

**private** **static** **double** **Nk**(**double** eta2k)

{

**return** **2** \* eta2k \* Math.Log(eta2k, **2**);

}

**private** **static** **double** **Eta2k**(**int** eta)

{

**return** eta \* Math.Log(eta, **2**);

}

**private** **static** **double** **V**(**int** k, **double** nk, **double** eta2k)

{

**return** k \* nk \* Math.Log((**2** \* eta2k), **2**);

}

**private** **static** **double** **P**(**double** n)

{

**return** **3** \* n / **8**;

}

**private** **static** **double** **Tk**(**double** p)

{

**return** p / (**5** \* **20**);

}

**private** **static** **double** **T**(**double** tk)

{

**return** tk / **2**;

}

**private** **static** **double** **B0**(**double** v)

{

**return** v / **3000**;

}

**private** **static** **double** **Tn**(**double** b0, **double** t)

{

**return** t / Math.Log(b0);

}

}

}

**Main.py**

**import** **math**

**def** **i**(eta):

**return** int(math.log2(eta) / **3** + **1**)

**def** **k**(eta, i):

total = **1**

**for** j **in** range(**1**, i):

total += int(eta / (**8**\*\*j))

**return** total

**def** **n**(k, nk):

**return** k \* nk

**def** **nk**(eta2k):

**return** **2** \* eta2k \* math.log2(eta2k)

**def** **eta2k**(eta):

**return** eta \* math.log2(eta)

**def** **v**(k, nk, eta2k):

**return** k \* nk \* math.log2(**2** \* eta2k)

**def** **p**(n):

**return** **3** \* n / **8**

**def** **tk**(n):

**return** n / (**5** \* **20**)

**def** **t**(tk):

**return** tk / **2**

**def** **b0**(v):

**return** v / **3000**

**def** **tn**(b0, t):

**return** t / math.log(b0)

etas = [**300**, **400**, **512**]

**for** element **in** etas:

i\_val = i(element)

k\_val = k(element, i\_val)

eta2k\_val = eta2k(element)

nk\_val = nk(eta2k\_val)

n\_val = n(k\_val, nk\_val)

v\_val = v(k\_val, nk\_val, eta2k\_val)

p\_val = p(n\_val)

tk\_val = tk(n\_val)

t\_val = t(tk\_val)

b0\_val = b0(v\_val)

tn\_val = tn(b0\_val, t\_val)

print(f"Значение = {element}")

print(f"Число уровней иерархии = {i\_val}")

print(f"Общее число модулей = {k\_val}")

print(f"Eta2k = {eta2k\_val}")

print(f"Nk = {nk\_val}")

print(f"Длина программы = {n\_val}")

print(f"Объём ПС = {v\_val}")

print(f"Длина ПС, выраженная в количестве команд ассемблера = {p\_val}")

print(f"Календарное время программирования = {tk\_val}")

print(f"Время отладки = {t\_val}")

print(f"Начальное количество ошибок = {b0\_val}")

print(f"Надежность ПС = {tn\_val}")

print("=============================================")