# Цель

Разработка программы, имитирующей процесс написания программы программистом, и расчет статистических оценок метрических характеристик полученных программ. Для имитации используется вероятностная модель выборки с возвратом из генеральной совокупности, состоящей из символов, пока последняя не будет исчерпана, а также проведение серий испытаний при разных значениях .

# Задание

1. Разработать программу для вероятностного моделирования процесса написания программы программистом с длиной словаря программы = 16, 32, 64, 128.

2. С помощью разработанной программы получить статистические оценки:

* длины программы ,
* дисперсии длины ,
* среднеквадратического отклонения,
* относительной ожидаемой погрешности .

3. С помощью приведенных формул получить теоретические значения и сравнить их с результатами моделирования.

4. По тексту разработанной программы посчитать длину ее словаря и длину программы. Рассчитать длину программы по размеру ее словаря с помощью приведенных формул. Сравнить посчитанное по тексту значение длины текста программы, с длиной текста программы, полученной по формуле.

5. По первому и второму пунктам задания определить – число единых по смыслу входных и выходных параметров представленных в сжатой без избыточной формы. Сравнить прогнозируемую длину программы с длиной программы, рассчитанной по тексту программы.

# Исходные тексты программ на языке C#

using System;

using System.Collections.Generic;

using System.Linq;

using System.Linq.Expressions;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1 {

internal class Program{

public static Random rnd = new Random();

static void Main(string[] args) {

int[] nus = { 16, 32, 64, 128 };

for (int i = 0; i < nus.Length; i++) {

double[] Ls = new double[100];

double teorL;

double D, teorD;

double Std, teorStd;

double Delta, teorDelta;

for (int j = 0; j < 100; j++) {

List<int> programText = generateProgramText(nus[i]);

Ls[j] = calcPractL(programText);

}

D = calcPractDisp(Ls);

Std = calcPractStd(Ls);

Delta = calcPractDelta(Ls);

teorL = calcL(nus[i]);

teorD = calcDisp(nus[i]);

teorStd = calcStd(nus[i]);

teorDelta = calcDelta(nus[i]);

}

}

public static List<int> generateProgramText(int nu) {

List<int> programText = new List<int>();

int[] isTicketInText = new int[nu];

while (isTicketInText.Sum() != nu) {

int tmp = rnd.Next(nu);

programText.Add(tmp);

isTicketInText[tmp] = 1;

};

return programText;

}

static double calcL(int etta) {

return 0.9 \* etta \* Math.Log(etta, 2);

}

static double calcPractL(List<int> programText) {

return programText.Count;

}

static double calcDisp(int etta) {

return Math.Pow(Math.PI, 2) \* Math.Pow(etta, 2);

}

static double calcPractDisp(double[] Ls) {

double disp = 0;

double mean = Ls.Average();

for (int i = 0; i < Ls.Length; i++)

disp += Math.Pow(Ls[i] - mean, 2);

disp /= Ls.Length;

return disp;

}

static double calcStd(int etta) {

return Math.Sqrt(calcDisp(etta));

}

static double calcPractStd(double[] Ls) {

return Math.Sqrt(calcPractDisp(Ls));

}

static double calcDelta(int etta) {

return 1 / (2 \* Math.Pow(etta, 2));

}

static double calcPractDelta(double[] Ls) {

return calcPractStd(Ls) / Ls.Average();

}

}

}

# Статистические и расчётные характеристики

Для каждого значения размера словаря n генерировалось 10 000 различных программ. По этом данным были рассчитаны метрические характеристики, которые были сравнены с теоретическими, посчитанными по следующим формулам:

* длина программы
  + дисперсия длины программы
  + среднеквадратическое отклонение
  + относительная ожидаемая погрешность

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Длина программы | | Дисперсия длины  программы | | Среднеквадратичное отклонение | | Относительная погрешность | |
|  |  |  |  |  |  |  |  |
| 16 | 57,6 | 56,42 | 421,1 | 442,5 | 20,5 | 21,03 | 0,36 | 0,37 |
| 32 | 144 | 127,85 | 1684,41 | 1249,59 | 41,04 | 35,35 | 0,29 | 0,27 |
| 64 | 345,6 | 314,34 | 6737,65 | 6289,22 | 82,08 | 79,30 | 0,23 | 0,25 |
| 128 | 806,4 | 707,01 | 26950,6 | 28822,3 | 164,17 | 169,77 | 0,2 | 0,24 |

Символ \* обозначает теоретические значения характеристик.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 16 | 1.18 | 21.4 | 0.53 | 0.01 |
| 32 | 16.15 | 434.82 | 5.69 | 0.02 |
| 64 | 31.26 | 448.43 | 2.78 | 0.02 |
| 128 | 99.39 | 1871.7 | 5.6 | 0.04 |

По результатам исследования можно увидеть, что с увеличением размера словаря выборочные оценки приближаются к теоретическим.

# Анализ разработанной программы

Перечислим и подсчитаем количество операторов и операндов в разработанной программе.

|  |  |
| --- | --- |
| Операторы | Число вхождений |
| ; | 45 |
| = | 22 |
| [] | 16 |
| + | 4 |
| <> | 5 |
| \* | 4 |
| new | 4 |
| < | 3 |
| ++ | 3 |
| / | 3 |
| for () | 3 |
| while () | 1 |
| != | 1 |
| - | 1 |
| /= | 1 |
| += | 1 |
| {} | 17 |
| return | 9 |
| f(x) – вызов функции | 31 |

|  |  |
| --- | --- |
| Операнды | Число вхождений |
| Ls | 15 |
| i | 12 |
| etta | 11 |
| nus | 7 |
| programText | 7 |
| D | 3 |
| Std | 3 |
| Delta | 3 |
| j | 4 |
| disp | 4 |
| teorL | 2 |
| teorD | 2 |
| teorStd | 2 |
| teorDelta | 2 |
| isTicketInText | 3 |
| tmp | 3 |
| rnd | 2 |
| 16 | 1 |
| 32 | 1 |
| 64 | 1 |
| 128 | 1 |
| 0 | 4 |
| 100 | 2 |
| 1 | 1 |
| 2 | 4 |
| 6 | 1 |
| Math.PI | 1 |
| 0.9 | 1 |

По составленным словарям операндов и операторов: , , .

Теоретическая длина программы .

Практическая длина программы .

Определим число для разработанной программы из выражения:

# Вывод

Была произведена разработка программы, имитирующей процесс написания программы и рассчитаны для неё соответствующие статистические оценки метрических характеристик. Были получены фактическая и теоретическая длина написанной программы и вычислено число единых по смыслу входных и выходных параметров.