|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | **3** |

**Название:**

«Исследование псевдослучайных чисел»

**Дисциплина:** Моделирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-76Б |  |  | А. А. Петрова |
|  | (Группа) |  |  | (И.О. Фамилия) |
|  |  |  |  |  | |  |
| Преподаватель |  |  |  | И. В. Рудаков |
|  |  |  |  | (И.О. Фамилия) |

2022 г.

**Задание**

Реализовать алгоритмический и табличный методы генерирования псевдослучайных чисел, а также критерий оценки случайной последовательности. Сравнить результаты работы этого критерия на одноразрядных, двухразрядных и трехразрядных последовательностях целых чисел.

**Математическая формализация**

В качестве алгоритмического способа генерации псевдослучайных чисел был выбран линейный конгруэнтный метод, который был предложен Д. Г. Лемером. Суть метода заключается в вычислении последовательности чисел Xn, полагая:

,

где .

Для оценки случайности генерируемых последовательностей был выбран критерий, основанный на расчете частотности единиц в двоичном представлении чисел последовательности. Цель данного теста — выяснить действительно ли число нулей и единиц в последовательности приблизительно одинаковы, как это можно было бы предположить в случае истинно случайной бинарной последовательности. Тест оценивает, насколько близка доля единиц к 0,5.

Таким образом, процедура проверки последовательности заключается в следующем:

1. Перевести каждое число последовательности в двоичный вид.
2. Занести все полученные двоичные числа в одну последовательность.
3. Подсчитать количество единиц в последовательности *n\_ones*.
4. Подсчитать общее число 0 и 1 в последовательности (длину последовательности) *len*.
5. *P = n\_ones / len* – искомая доля единиц в последовательности.
6. Если , то последовательность случайна. Иначе – неслучайна.

**Реализация**

В листингах ниже представлена реализация алгоритмического метода генерации и критерия оценки случайности.

Листинг 1: алгоритмический метод генерации последовательности

|  |
| --- |
| **def** rand\_alg**(**self**,** low**,** high**,** col**):** # Лемер  m **=** 2. **\*\*** 31  a **=** 1664525  c **=** 1013904223  current **=** 10  seq **=** **[]**  **for** i **in** **range(**COUNT**):**  current **=** **(**a **\*** current **+** c**)** **%** m  result **=** **int(**low **+** current **%** **(**high **-** low**))**  seq**.**append**(**result**)**  **for** i **in** **range(**0**,** COUNT**,** STEP**):**  self**.**table\_2**.**setItem**(**i **//** STEP**,** col**,** QTableWidgetItem**(str(**seq**[**i**])))** |

Листинг 2: критерий оценки случайности последовательности

|  |
| --- |
| **def** \_criterion**(**self**,** seq**):**  bin\_seq **=** ""  **for** num **in** seq**:**  bin\_seq **+=** **bin(**num**)[**2**:]**  ones **=** 0  **for** digit **in** bin\_seq**:**  **if** digit **==** '1'**:**  ones **+=** 1  p **=** ones **/** **len(**bin\_seq**)**  **return** p |

**Результаты работы**

На рисунках ниже приведены результаты работы программы при разных последовательностях.

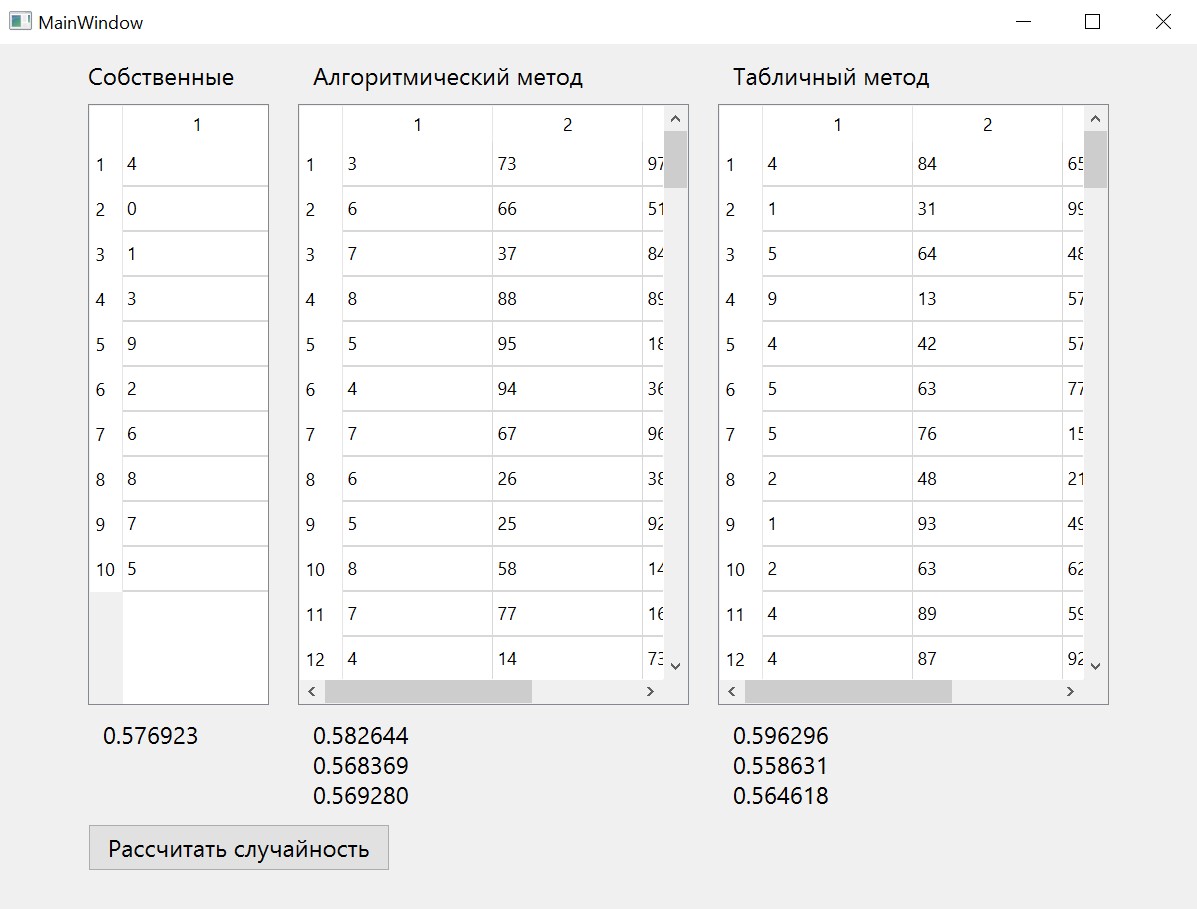
****

Рисунок 1: случайные последовательности (

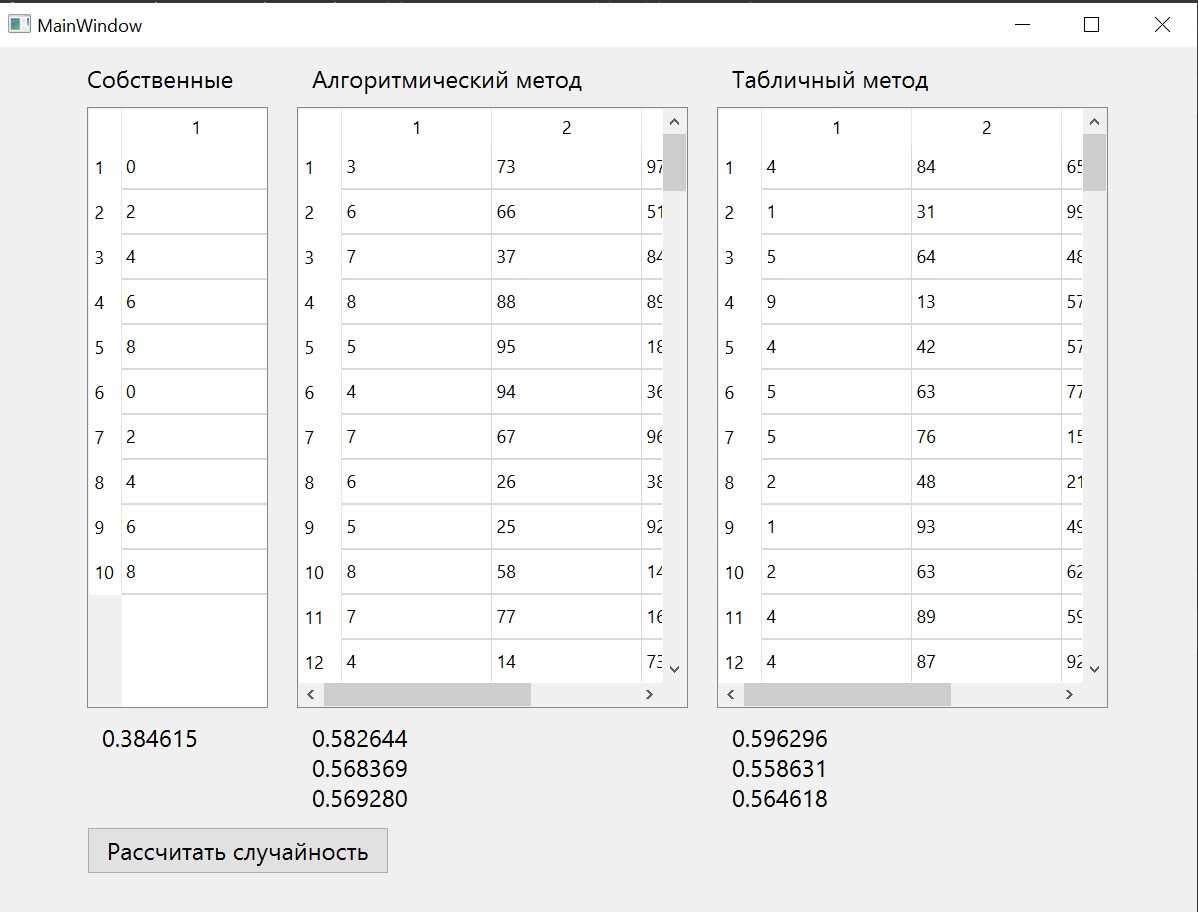


Рисунок 2: неслучайная собственная последовательность (p значительно меньше 0,5)

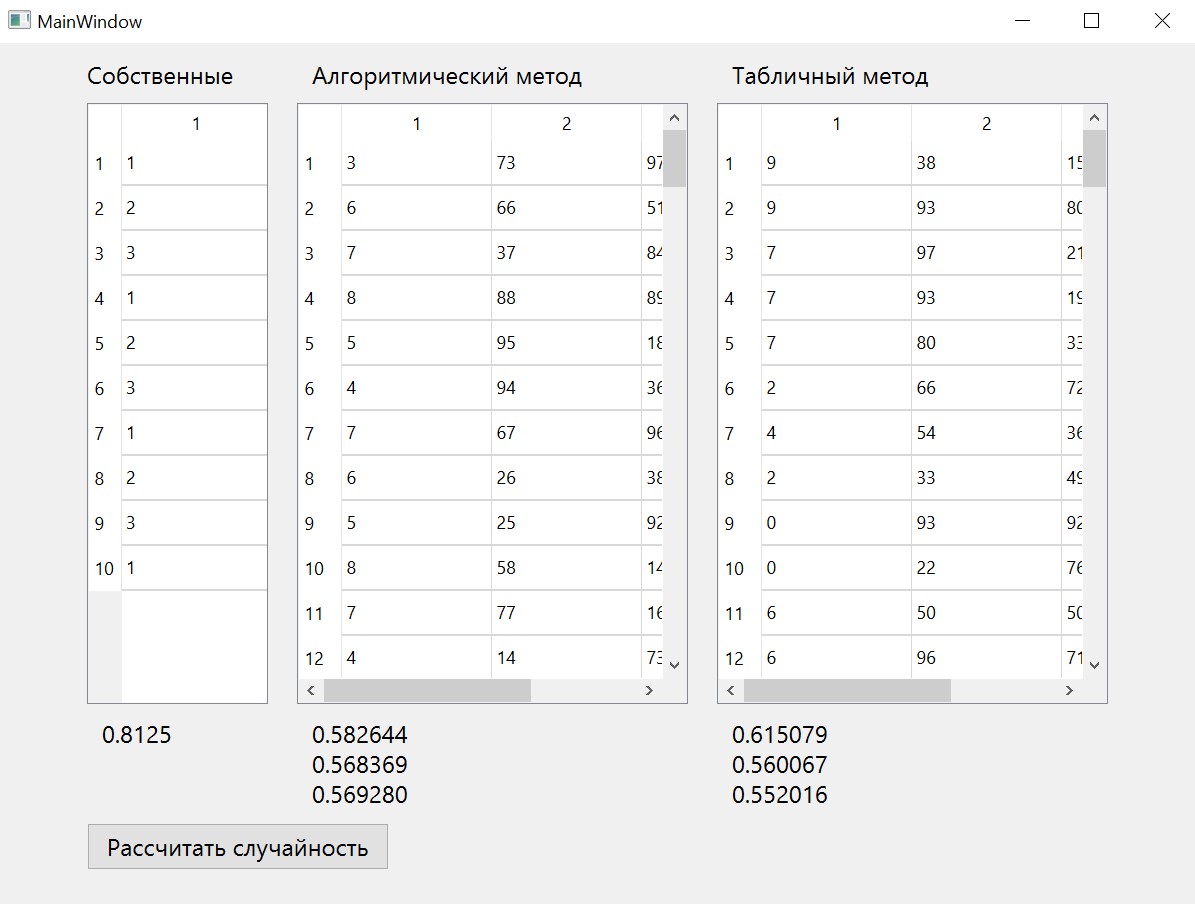


Рисунок 3: неслучайная собственная последовательность (p значительно больше 0,5)