**МИНОБРНАУКИ РОССИИ**

**ФГБОУ ВО «ТУЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт прикладной математики и компьютерных наук

Кафедра вычислительной техники

Отчет по лабораторной работе № 3

«Модерирование дискретных случайных событий»

по дисциплине «Основы моделирования систем»

Выполнил студент группы 220681:

Шайхаттаров Д.В.

Проверил:

доц. Семенчев Е.А.

Тула 2020

# ЦЕЛЬ РАБОТЫ

Практическое освоение метода моделирования на ЭВМ дискретных случайных величин и событий.

# ЗАДАНИЕ

1. Имитировать бросание симметричной монеты. Результаты первых 100 бросаний вывести на печать в строку в виде последовательности букв "О" и "Р". Составить алгоритм и программу подсчёта частоты, с которой в последовательности из бросаний встречается заданная комбинация орлов и решек: P. Результаты для вывести на печать и сравнить с теоретической вероятностью. Программу составить для заданного номера варианта.
2. Имитировать бросание игральной кости. Результаты первых 100 бросаний вывести строкой на печать. Составить алгоритм и программу расчёта частоты события, состоящего в том, что сумма очков при двух последовательных бросаниях равна заданному К. (К принимает значения от 2 до 12. Число бросаний равно 1000). Сравнить рассчитанную частоту с теоретической вероятностью.
3. При работе ЭВМ время от времени возникают неисправности (сбои). Ежедневное количество сбоев описывается как случайная величина X, распределённая по закону Пуассона с параметром a =1.5. Составить алгоритм и программу имитации сбоев в ЭВМ в течение 365 дней и вычисления частоты события: сумма числа сбоев в двух последовательных днях равна K. (K-номер студента по списку группы). Число сбоев за каждый из первых 100 дней вывести на печать в строку.

# ХОД РАБОТЫ

## Бросание симметричной монеты

Вероятность выпадения орла или решки равна 0,5 и т.к. наша последовательность является «Р», то вероятность её выпадения тоже равняется 0,5.

На рисунке 1 представлен алгоритм работы программы.

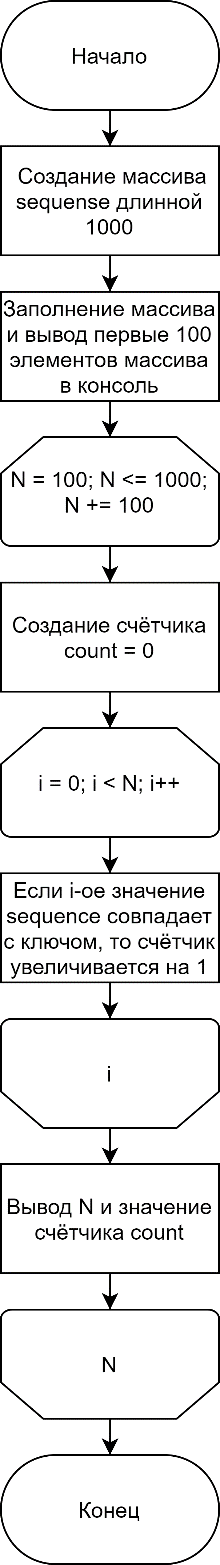
****

Рисунок 1 – Алгоритм работы программы броска игральной кости

# ТЕСТИРОВАНИЕ

На рисунке 2 представлены результаты работы программы. Код программы представлен в «Приложении».

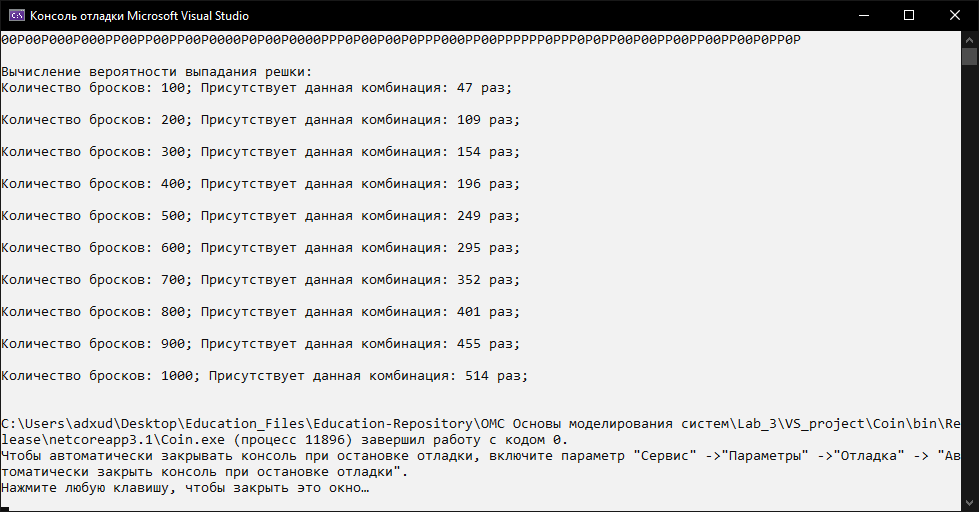


Рисунок 2 – Тестирование программы

# ВЫВОДЫ

Был освоен метод моделирования на ЭВМ дискретных случайных величин и событий. По полученным данным можно сделать вывод, что при увеличении количества испытаний частота сходится по вероятности к теоретически вычисленной вероятности.

# Приложение

## Coin.cs

using System;

namespace Coin {

class Coin {

readonly byte[] sequence = new byte[1000];

readonly byte key = 1;

byte Flip() => (byte) new Random().Next(0,2);

public Coin() { }

public void Test() {

Console.WriteLine("Программная имитация броска монетки:");

InitSequence();

Console.WriteLine("Вычисление вероятности выпадания решки:");

FindKey();

}

void InitSequence() {

for(int i = 0; i < 1000; i++) {

sequence[i] = Flip();

if (i < 100) {

if (sequence[i] == 0)

Console.Write("0");

else

Console.Write("Р");

}

}

Console.WriteLine("\n");

}

void FindKey() {

for (int N = 100; N <= 1000; N += 100) {

int count = 0;

for (int i = 0; i < N; i++) {

if (sequence[i] == key) { count++; }

}

Console.WriteLine("Количество бросков: {0}; Присутствует данная комбинация: {1} раз;\n", N, count);

}

}

}

}

## Program.cs

namespace Coin {

class Program {

static void Main() {

new Coin().Test();

}

}

}