

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | Информатика и вычислительная техника | | | |
| Направление подготовки | | 09.03.03 | Прикладная информатика в химии | | | |
| Направленность (профиль) | |  | САПР | | | |
|  | |  |  | | | |
| Факультет | |  | Информационных технологий и управления | | | |
| Кафедра | |  | Систем автоматизированного  проектирования и управления | | | |
| Учебная дисциплина | |  | Методы поддержки принятия решений | | | |
|  |  | | |  |  |  |

Курс \_\_\_3\_\_\_ Группа 475

**Лабораторная работа №3**

**Вероятностное программирование**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Андрианова К.И.\_\_

(подпись, дата) (инициалы, фамилия)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Овчинников Р.С.

(подпись, дата) (инициалы, фамилия)

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пекер В.А.

(подпись, дата) (инициалы, фамилия)

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Уланов В.Н.

(подпись, дата) (инициалы, фамилия)

Санкт-Петербург

2019

# Цель работы

Целью данной работы является создание программы, которая осуществляет решение вероятностного программирования. Предлагается практическое решение задачи. Для удобства использования программы разработан эргономичный интерфейс.

**Задание**

Предложено практическое решение задачи: у нас есть два типа билетов (в одну сторону или в обе стороны), при условии, что билет в обе стороны стоит больше, чем билет в одному сторону, но меньше, чем покупать два билета в одну сторону. Покупка билета зависит от погоды, для этого мы задаем вероятность хорошей погоды в каждый из двух дней. В следствии чего, на экран должна выводиться вероятность выгодной покупки, каждого из предложенных варианта покупки билетов.

**План выполнения**

1. Знакомство с вероятностным программированием
2. Знакомство с Байесовским программированием
3. Блок-схема программы
4. Разработка интерфейса
5. Тестирование
6. Анализ тестирования
7. Вывод

**Ход выполнения работы**

1. Возможность автоматизации части логического вывода привела к разработке вероятностных языков программирования (PPL), которые позволяют четко разделить создание модели и логический вывод. Использую PPL пользователи могут задать вероятностную модель, написав несколько строк кода, после чего вывод делается автоматически.

Вероятностное программирование — способ описания статистических моделей в виде декларативного программного кода, с последующим применением алгоритмов анализа и статистического вывода. Родственно методам Байесовской вероятности (например, Байесовым сетям).

В отличие от классического программирования, конечной целью которого является выполнение программы, вероятностное программирование делает упор на анализе вероятностных программ. Типовой сценарий — описание зависимостей наблюдаемых данных от скрытых параметров в виде программы, после чего используются встроенные алгоритмы для обратного вероятностного вывода значений на входе от известных данных на выходе.

Идея вероятностного программирования:

• Вероятностное программирование предоставляет язык описания Байесовских моделей

• С помощью этого языка исследователь тратит основное время на обдумывание структуры модели, а не на разработку программного кода

• Вероятностное программирование компилирует сгенерированную модель и выполняет процедуры определения оптимальных параметров наиболее приемлемым методом

2. Байесовская вероятность — это интерпретация понятия вероятности, используемая в байесовской теории. Вероятность определяется как степень уверенности в истинности суждения. Для определения степени уверенности в истинности суждения при получении новой информации в байесовской теории используется теорема Байеса. Байесовское программирование также можно рассматривать как формальную алгебраическую систему для задания графических моделей, таких как, например, байесовские сети, динамические байесовские сети, фильтры Калмана или скрытые марковские модели. Действительно, байесовское программирование обобщает байесовские сети и имеет выразительную мощность эквивалентную фактор-графам.

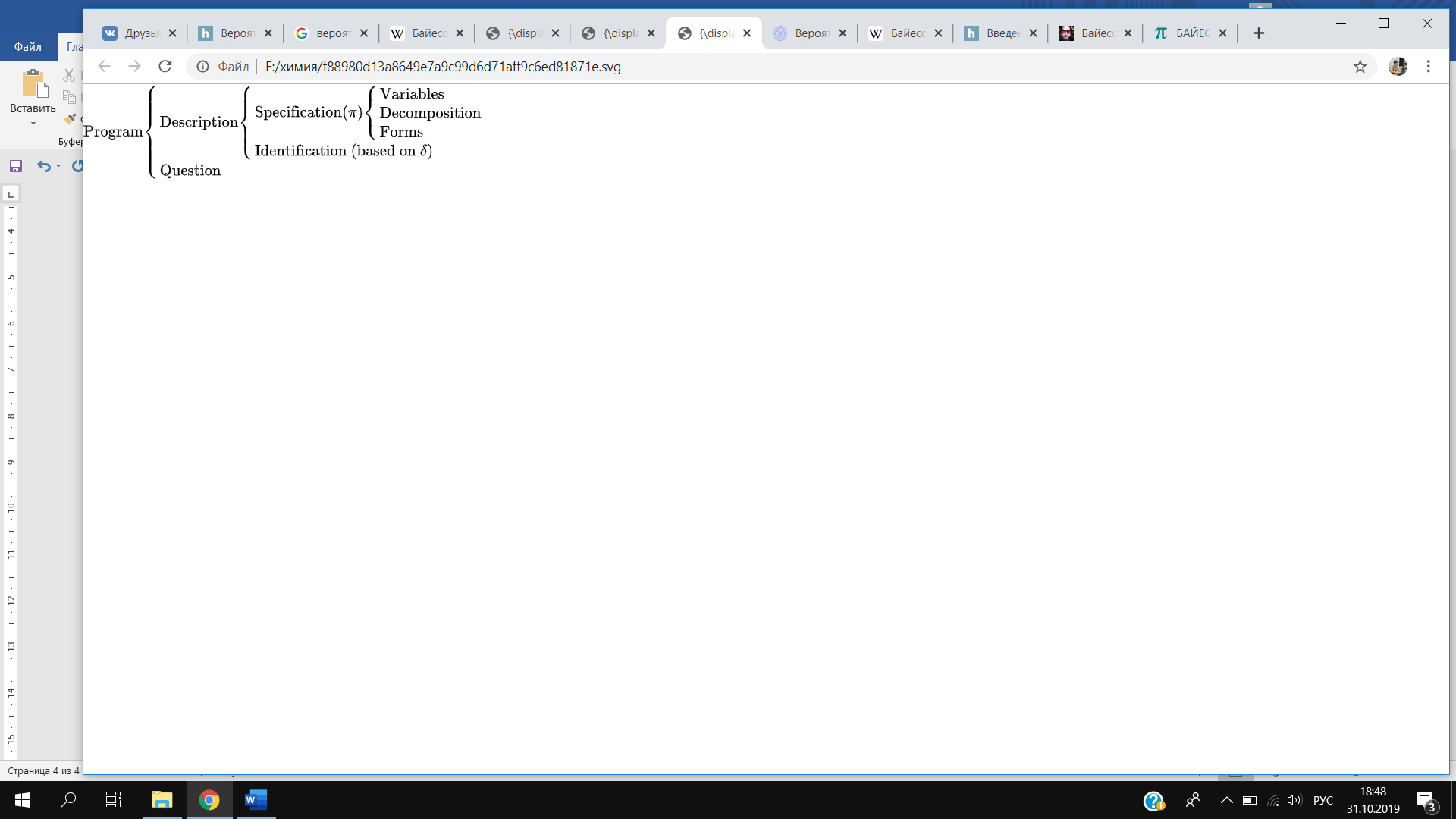
Байесовское программирование также можно рассматривать как формальную алгебраическую систему для задания графических моделей, таких как, например, байесовские сети, динамические байесовские сети, фильтры Калмана или скрытые марковские модели. Действительно, байесовское программирование обобщает байесовские сети и имеет выразительную мощность эквивалентную фактор-графам.

Байесовская вероятность противопоставляется частотной, в которой вероятность определяется относительной частотой появления случайного события при достаточно длительных наблюдениях.

Неприятная деталь, связанная с использованием байесовской вероятности в том, что задания вероятности недостаточно для того, чтобы понять её природу.

Байесовская программа является средством задания семейства распределений вероятности.

Ниже представлены составляющие элементы байесовской программы:



Программа строится из описания и вопроса.

1. Описание строится с помощью какого-либо определения, заданного программистом, и идентификации или процесса обучения для параметров, не полностью описанных в определении, с применением набора данных.
2. Определение строится из набора значимых переменных, декомпозиции и набора форм.
3. Формы являются или параметрическими формами, или вопросами к другим байесовским программам.
4. Вопрос задает распределение вероятности, которое необходимо вычислить.

Сравнение двух видов программирования:

Цель вероятностного программирования — объединение сферы классических языков программирования с вероятностным моделированием (особенно с байесовскими сетями) для того, чтобы быть в состоянии иметь дело с неопределенностью и в то же время пользоваться выразительной силы языков программирования для описания сложных моделей.

Расширенные классические языки программирования включают в себя логические языки, как предлагается в Вероятностной абдукции Горна, Логике независимого выбора, PRISM и ProbLog, являющимся расширением языка Prolog.

Оно также может быть расширением функциональных языков программирования (по сути LISP и Scheme), такими как IBAL или Church[en]\*. Языки, лежащие в основе расширения, также могут быть объектно-ориентированными, как в случае BLOG и FACTORIE, или более стандартными, как в CES и FIGARO.

Цель байесовского программирования несколько другая. Установка Джейнса о «вероятности как логике» отстаивает мнение, что вероятность является расширением и альтернативой логике, поверх которой может быть выстроена заново вся теория рациональности, алгоритмов и программирования. Байесовское программирование не ищет способа расширить классические языки, оно стремится заменить их новым подходом к программированию на основе вероятности, который учитывает неполноту и неопределенность.

Точное сравнение семантики и выразительной мощности байесовского и вероятностного программирования пока остается открытым вопросом.

**Выбор инструментария**

Программа написана в Visual Studio 2017 на языке C#. Так как язык C# обладает достаточными функциями для вероятностного программирования, такими как random и т д.

**Интерфейс**

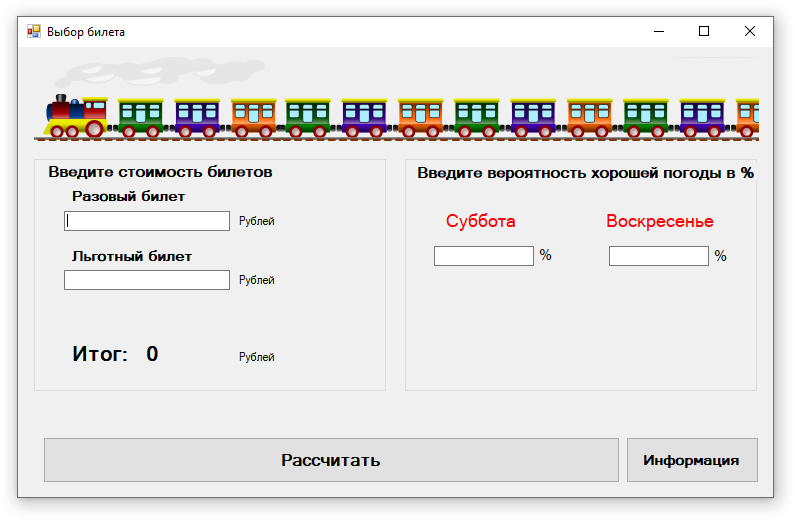
****

Рисунок 1 – Интерфейс программы

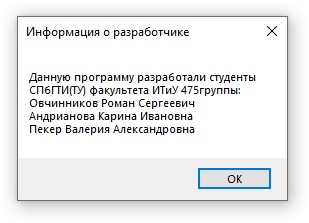
****

Рисунок 2 – Информация о разработчиках

**Блок-схемы программы**

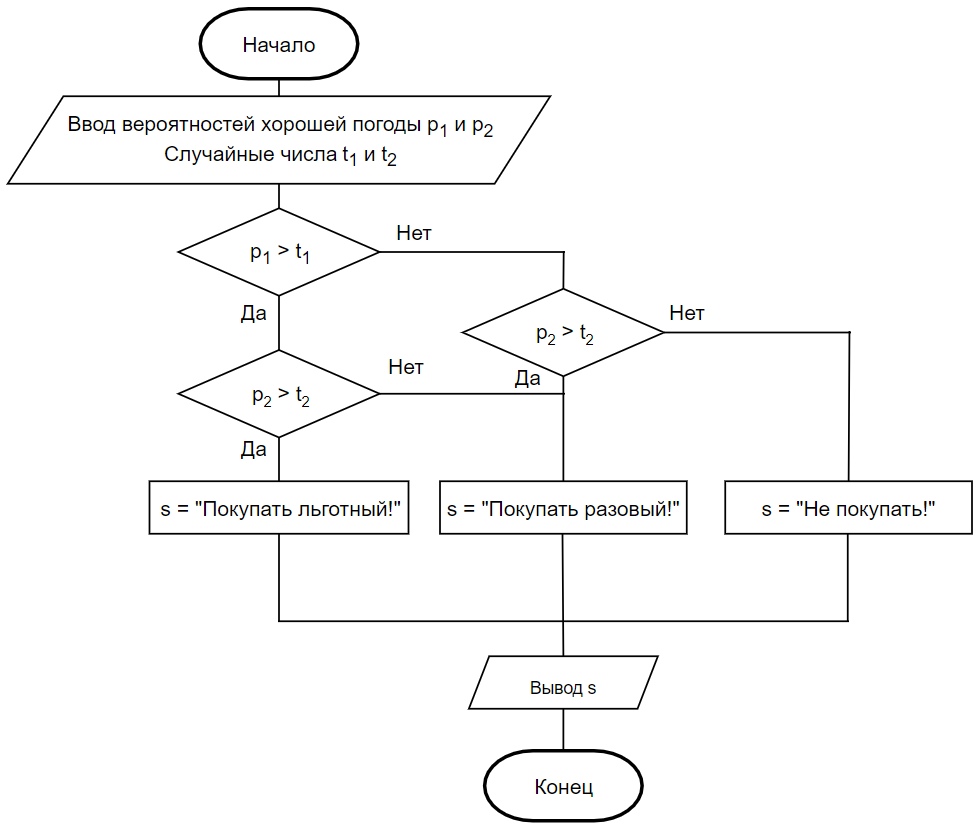


Рисунок 3 – Блок-схема программы

**Тестирование**

Сначала мы заполняем поля со стоимостью разового и льготного билета, а также вводим вероятность хорошей погоды в каждый из дней (рис.4). Затем на экран выводится результат в виде красного креста или зеленой галочки рядом с полем билета. (рис.5).

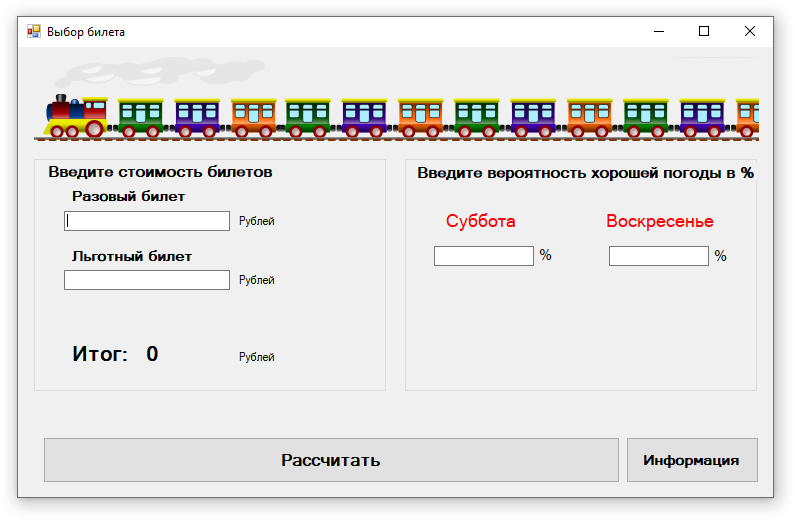


Рисунок 4 – Заполнение всех полей

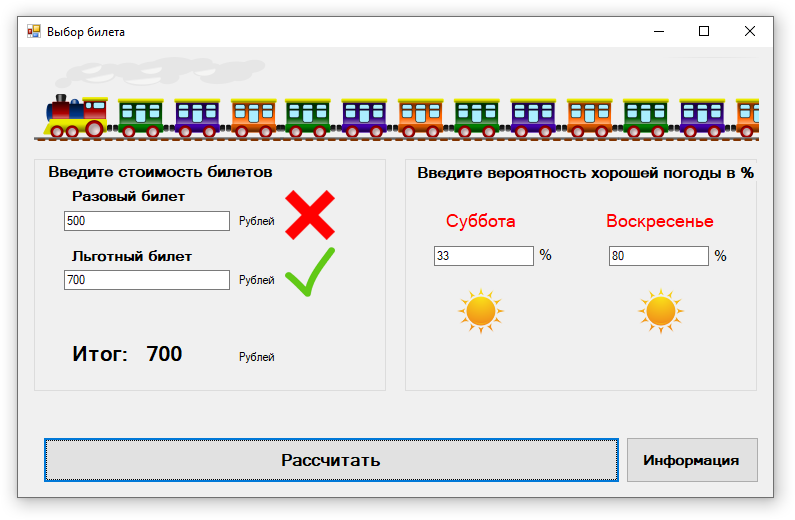


Рисунок 5 – Вывод результата программы

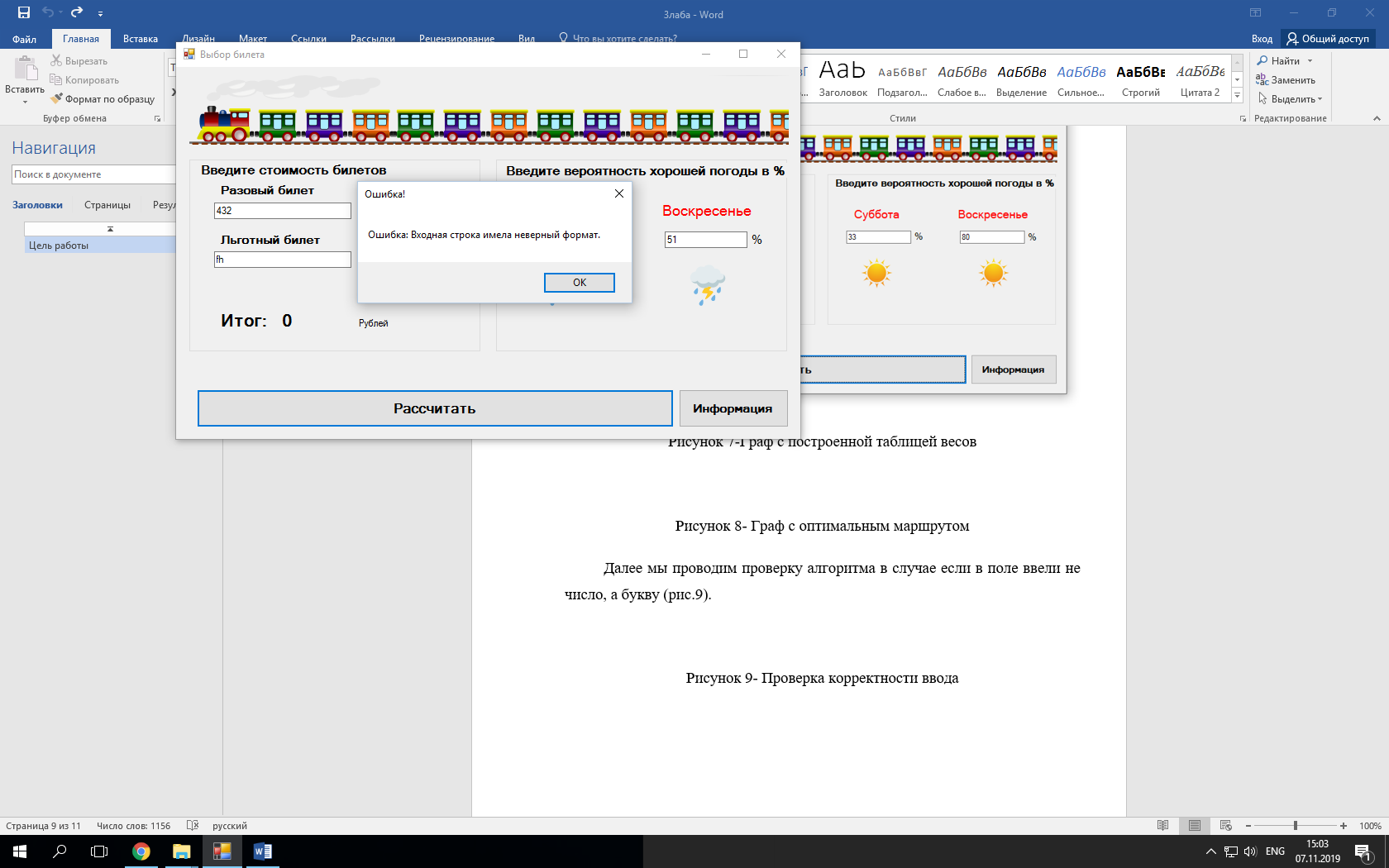
Далее мы проводим проверку работы программы в случае, если в поле ввели не число, а букву (рис.6).

Рисунок 6 – Проверка корректности ввода

Далее, мы проводим тестирование, где задаем количество повторов для расчета, чтобы программа вывела вероятность, с которой выгоднее купать каждый из билетов. (рис. 7, 8, 9)

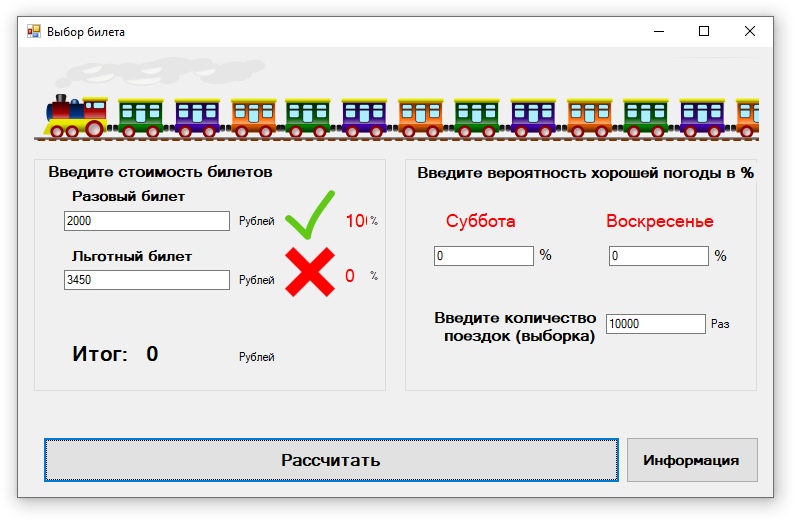


Рисунок 7 – Расчет вероятности выгодной покупки билета при заданном количестве повторений

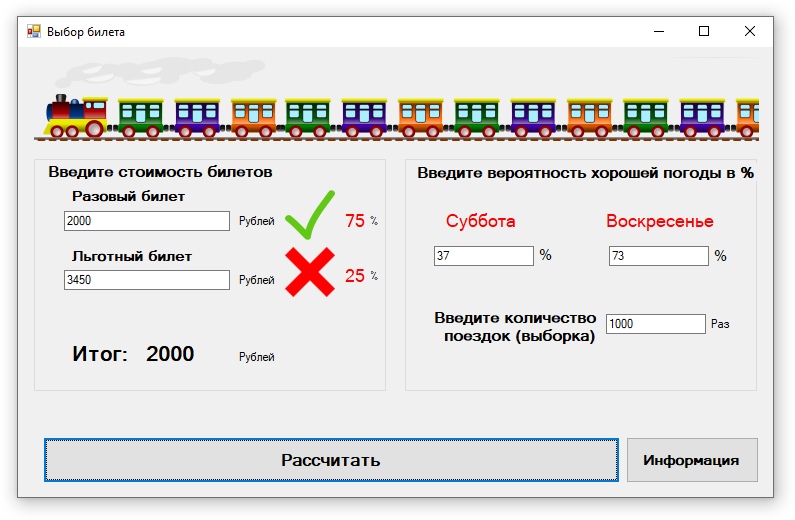


Рисунок 8 – Расчет вероятности выгодной покупки билета при заданном количестве повторений

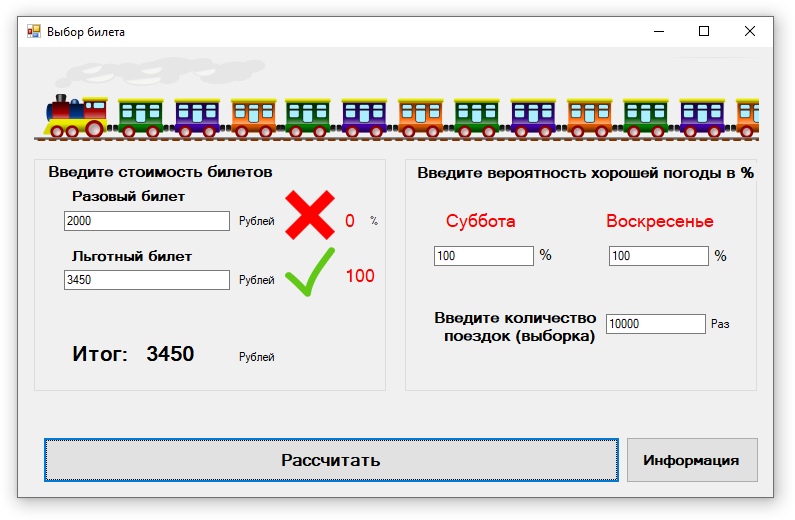


Рисунок 9 – Расчет вероятности выгодной покупки билета при заданном количестве повторений

Далее, проводим следующие виды тестирования:

1. Выставляем маленькие проценты хорошей погоды и рассматриваем результаты при выборке 10, 100,1000.

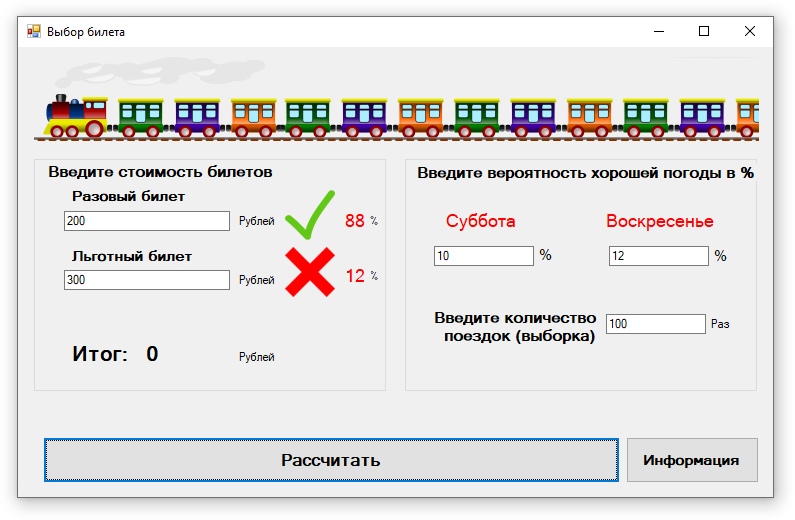


Рисунок 10 – Расчет вероятности выгодной покупки билета при маленькой вероятности хорошей погоды с количеством повторений 100.

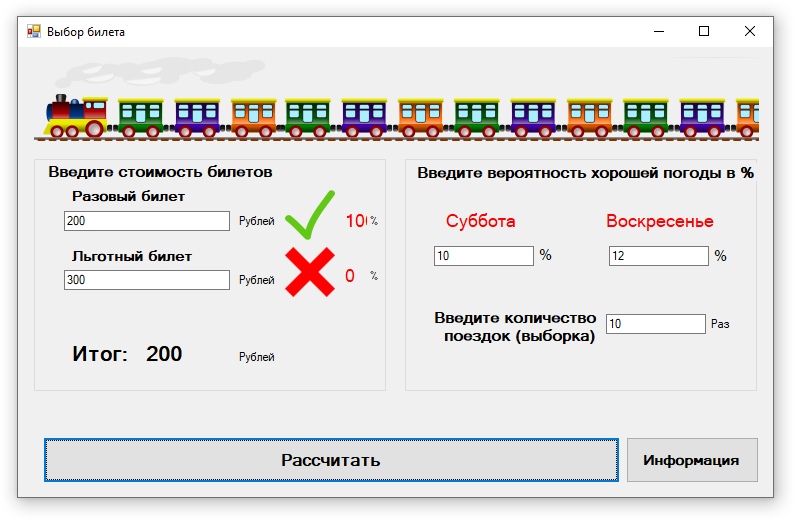


Рисунок 11 – Расчет вероятности выгодной покупки билета при маленькой вероятности хорошей погоды с количеством повторений 10.

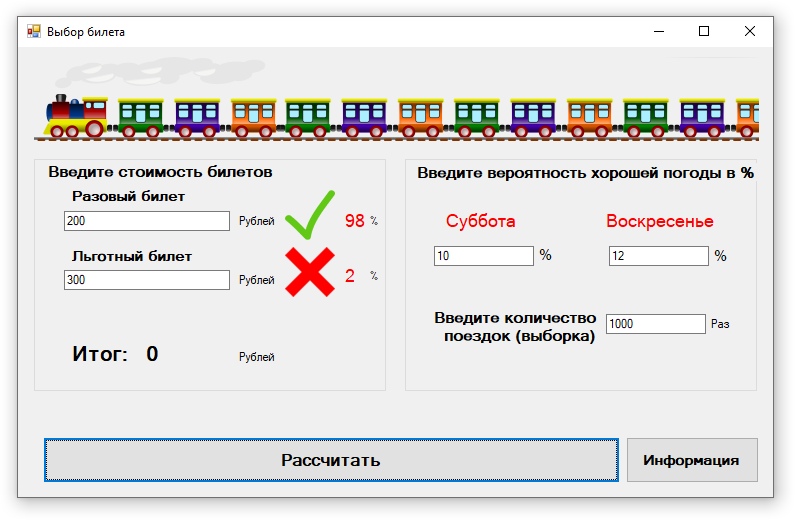


Рисунок 12 – Расчет вероятности выгодной покупки билета при маленькой вероятности хорошей погоды с количеством повторений 1000.

1. Выставляем большие проценты хорошей погоды и рассматриваем результаты при выборке 10, 100,1000.

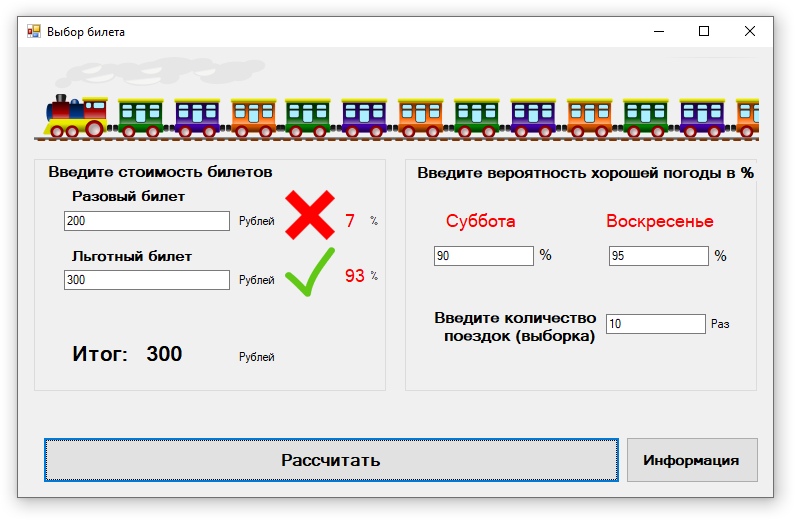


Рисунок 13 – Расчет вероятности выгодной покупки билета при большой вероятности хорошей погоды с количеством повторений 10.

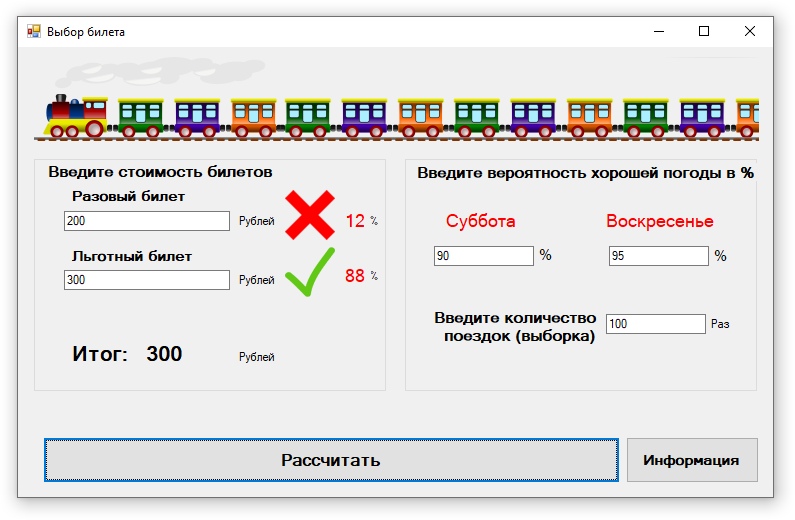


Рисунок 14 – Расчет вероятности выгодной покупки билета при большой вероятности хорошей погоды с количеством повторений 100.

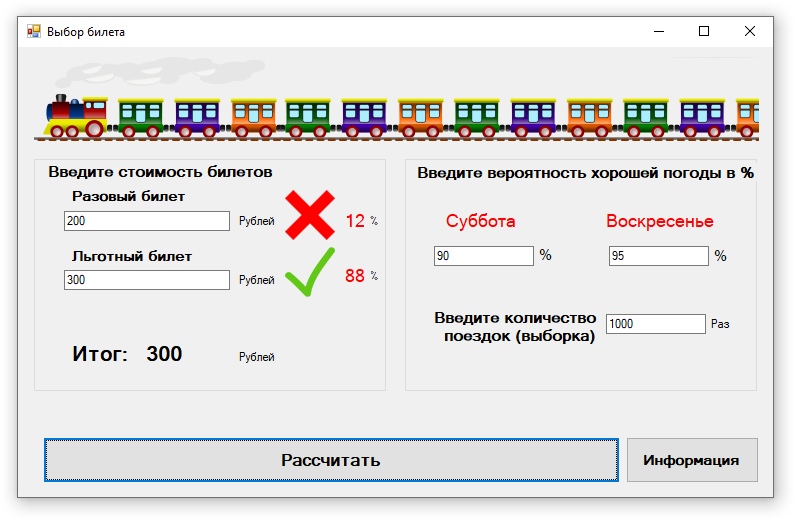


Рисунок 15 – Расчет вероятности выгодной покупки билета при большой вероятности хорошей погоды с количеством повторений 1000.

1. Выставляем вероятность хорошей погоды 50 на 50 и рассматриваем результаты при выборке 10, 100,1000.

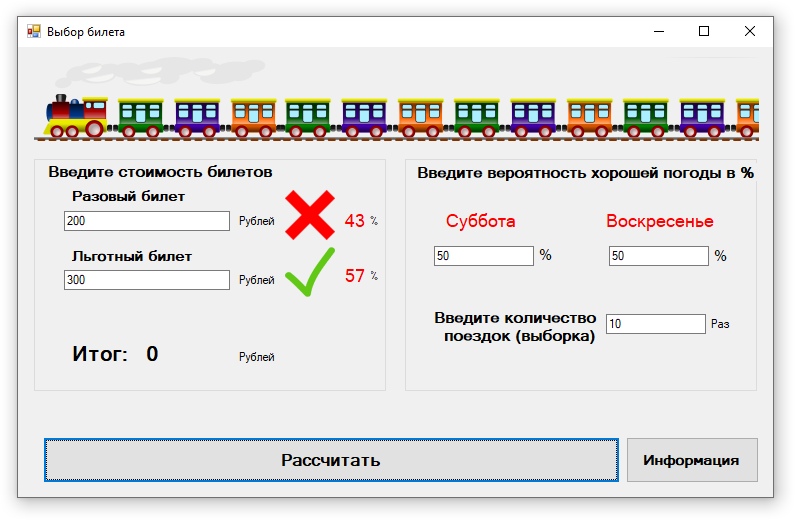


Рисунок 16 – Расчет вероятности выгодной покупки билета при вероятности хорошей погоды 50 на 50 с количеством повторений 10.

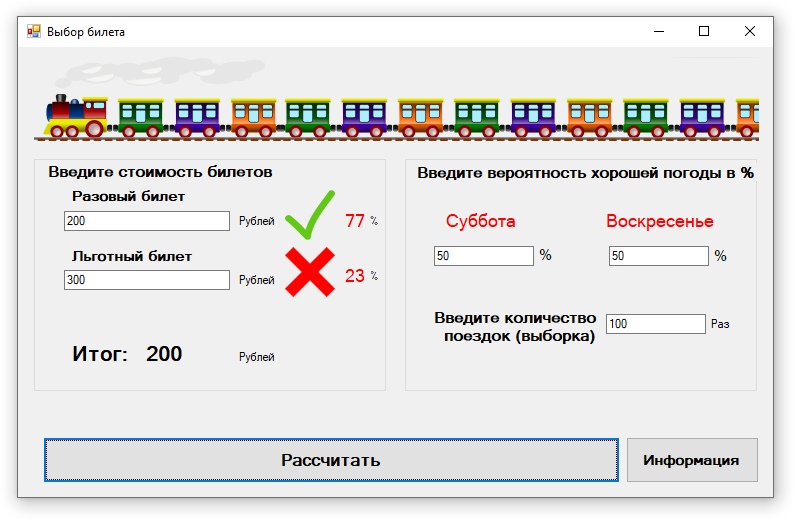


Рисунок 17 – Расчет вероятности выгодной покупки билета при вероятности хорошей погоды 50 на 50 с количеством повторений 100.

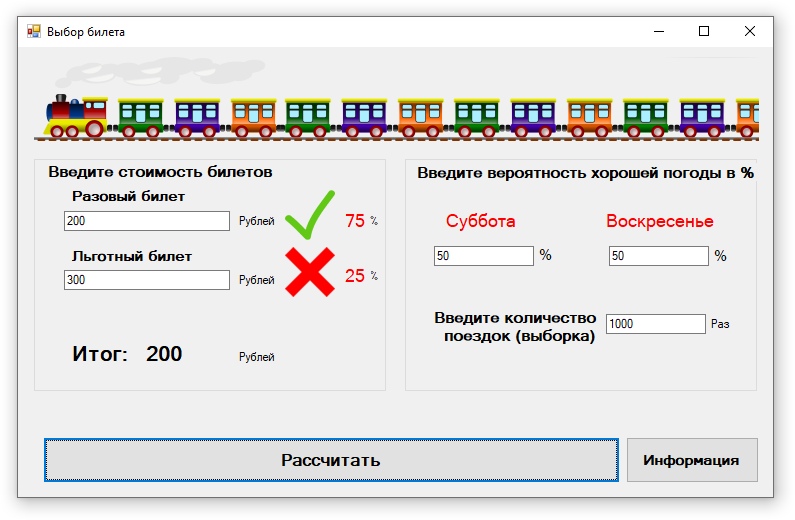


Рисунок 18 – Расчет вероятности выгодной покупки билета при вероятности хорошей погоды 50 на 50 с количеством повторений 1000.

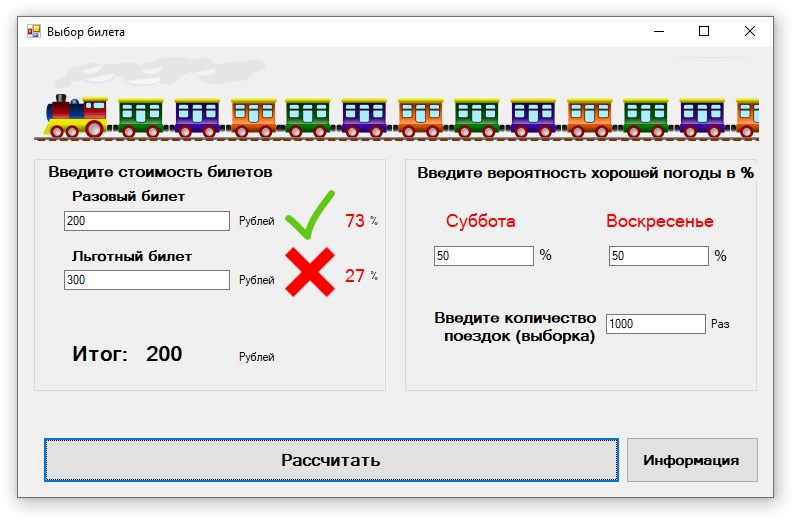


Рисунок 19 – Расчет вероятности выгодной покупки билета при вероятности хорошей погоды 50 на 50 с количеством повторений 1000.

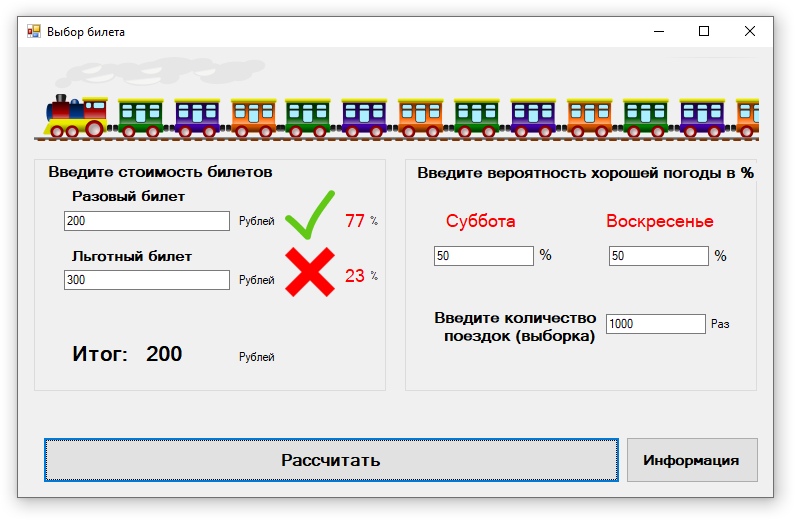


Рисунок 20 – Расчет вероятности выгодной покупки билета при вероятности хорошей погоды 50 на 50 с количеством повторений 1000.

**Анализ тестирования**

В ходе тестирования мы убедились, что написанная нами программа работает. Интерфейс, предложенный нами эргономичен и удобен для использования. Функция не полностью случайна, так как мы вывели некоторую закономерность появления выгодной покупки билета.

**Код программы (алгоритм)**

public static void program()

{

Random rnd = new Random();

int valueSaturday = rnd.Next(0, 100);

int valueSunday = rnd.Next(0, 100);

Obrab.weatherSaturday = false;

Obrab.weatherSunday = false;

// Если заданное число больше рандомного, то погода хорошая

if (Obrab.saturday >= valueSaturday)

{

Obrab.weatherSaturday = true;

}

if (Obrab.sunday >= valueSunday)

{

Obrab.weatherSunday = true;

}

if(Obrab.weatherSaturday && Obrab.weatherSunday) // хорошая хорошая

{

Obrab.result = Obrab.ticketPreferential;

Obrab.vivod = 1;

}

else if (!Obrab.weatherSaturday && Obrab.weatherSunday) // плохая хорошая

{

Obrab.result = Obrab.ticketOrdinary;

Obrab.vivod = 2;

}

else if (Obrab.weatherSaturday && !Obrab.weatherSunday) // хорошая плохая

{

Obrab.result = Obrab.ticketOrdinary;

Obrab.vivod = 3;

}

else if (!Obrab.weatherSaturday && !Obrab.weatherSunday) // плохая плохая

{

Obrab.result = 0;

Obrab.vivod = 4;

}

}

**Вывод**

В ходе лабораторной работы мы изучили вероятностное программирование и методы его реализации. Написали программу, с помощью которой мы выясняли какой билет нам выгоднее приобрести, это зависело от вероятности погоды, которую мы задавали. Так же добавили функцию, которая рассчитывает, при заданных количествах повторов , вероятность, с которой выгоднее купить каждый из предложенных вариантов билетов.