|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  Факультет среднего профессионального образования | |
| **Курсовая работа**  по междисциплинарному курсу «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей»    **Разработка программного обеспечения для автоматизации расчётов**  **в теории вероятностей**  Пояснительная записка  ОГУ 09.02.07. 3024. 019 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 22ИСП-2  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Дерипалов  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

|  |
| --- |
| Утверждаю  Председатель предметно-цикловой комиссии дисциплин профессионального цикла |
| \_\_\_\_\_\_\_\_\_\_\_\_ Ж.В. Михайличенко  подпись |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту \_\_\_\_\_Дерипалову Артёму Вячеславовичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

по специальности \_\_\_09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по междисциплинарному курсу \_\_Технология разработки программного обеспечения\_\_\_\_\_\_\_\_\_\_

1. Тема работы: \_\_\_Разработка программного обеспечения для автоматизации расчётов в теории вероятностей \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_\_июня\_\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Разработать программный продукт, позволяющий решать 7-10 задач из теории вероятностей с использованием различных входных данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и практикумы по теории вероятностей\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Перечень вопросов, подлежащих разработке: \_ а) изучить предметную область, выполнить анализ требований к программному обеспечению, составить техническое задание на разработку; б) выполнить проектирование системы с помощью CASE-средств; в) для решения поставленной задачи реализовать оконное приложение на языке C# и протестировать его; г) сформулировать предложения по внедрению, эксплуатации и сопровождению разработанного программного обеспечения. Сделать выводы по результатам проделанной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Перечень графического (иллюстративного) материала: таблицы, графики, рисунки, схемы, отражающие теоретический материал и программную реализацию поставленной задачи\_\_\_\_\_\_\_

Дата выдачи и получения задания

Руководитель «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ж.В. Михайличенко\_\_\_

подпись инициалы, фамилия

Студент «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_А.В. Дерипалов\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 019 ПЗ

Разраб.

Дерипалов А.В.

Провер.

Михайличенко Ж

Реценз.

Н. Контр.

Утверд.

Разработка программного обеспечения для автоматизации расчётов в теории вероятностей

Лит.

Листов

47

22ИСП-2

В курсовой работе по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществление интеграции программных модулей» проведена разработка автоматизированной системы для автоматизации расчётов в теории вероятностей.

В первой главе курсовой работы «Анализ требований проектирования программного продукта», описываются ключевые этапы разработки программного продукта, начиная с анализа предметной области и технического задания на разработку автоматизированной системы в котором формулируются конкретные требования к программному продукту и его функциональности для оптимизации расчётов в теории вероятностей, заканчивая проектированием информационной системы, где рассматривается проектирование архитектуры программного продукта, выбор технологий и создание дизайна системы.

Во второй главе курсовой работе «Разработка и тестирование программного продукта», речь идёт о важных аспектах создания программного продукта, включая выбор языка программирования, на котором будет написана программа и выбор среды разработки, а также разработка интерфейса. Рассматривается алгоритмизация задач и их программная реализация, что включает в себя написание, тестирование и отладку кода, а также подготовку к последующему тестированию продукта.

В третьей главе курсовой работы «Рекомендации по внедрению, эксплуатации и сопровождению программного продукта» представлены рекомендации, которые помогут обеспечить успешное внедрение, эффективную эксплуатацию и сопровождение программного продукта.

Пояснительная записка содержит 47 страницы, в том числе 17 рисунков, 3 таблиц, 8 источников, 1 приложение.

Разработка приложения выполнена студентом группы 22ИСП-2 А. В. Дерипаловым.

**Содержание**

[Введение 5](#_Toc168396663)

[1 Анализ требований и проектирование программного продукта 6](#_Toc168396664)

[1.1 Анализ предметной области 6](#_Toc168396665)

[1.2 Техническое задание на разработку 8](#_Toc168396666)

[1.3 Проектирование 11](#_Toc168396667)

[2 Разработка и тестирование программного продукта 15](#_Toc168396668)

[2.1 Обоснование программных средств реализации 15](#_Toc168396669)

[2.2 Разработка пользовательского интерфейса 16](#_Toc168396670)

[2.3 Алгоритмизация и программирование 22](#_Toc168396671)

[2.4 Тестирование 24](#_Toc168396672)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного продукта 27](#_Toc168396673)

[3.1 Руководство пользователя 27](#_Toc168396674)

[3.2 План внедрения и сопровождения 33](#_Toc168396675)

[Заключение 35](#_Toc168396676)

[Список использованных источников 36](#_Toc168396677)

[Приложение А 37](#_Toc168396678)

# Введение

Актуальность темы курсовой работы:

1. Широкий спектр применения:

Теория вероятностей используется во многих областях, таких как:

* Финансы: оценка рисков, анализ инвестиций, страхование;
* Наука: физика, математика, статистика, биология, медицина;
* Инженерия: обработка сигналов, теория управления, машинное обучение;
* Искусственный интеллект: разработка алгоритмов, анализ данных;
* Экономика: прогнозирование, моделирование, принятие решений.

2. Снижение трудоемкости и повышение точности:

Ручные вычисления в теории вероятностей могут быть трудоемкими, утомительными и подверженными ошибкам. Программное обеспечение может:

* Автоматизировать рутинные задачи;
* Ускорить процесс вычислений;
* Повысить точность и надежность результатов;
* Освободить время для более сложных задач.

Цель курсовой работы:

Разработать программное обеспечение, которое автоматизирует расчёты в теории вероятностей.

Цель и задачи для достижения поставленной цели:

1. Анализ существующих решений:

* Изучить существующие программные решения для расчётов в теории вероятностей;
* Определить их преимущества и недостатки;
* Выявить потребности пользователей в данном ПО.

2. Проектирование ПО:

* Определить функциональные возможности ПО;
* Разработать архитектуру ПО.

3. Разработка ПО:

* Реализовать функциональные возможности ПО;
* Обеспечить удобный интерфейс пользователя;
* Провести тестирование ПО.

Разработка программного обеспечения для автоматизации расчётов в теории вероятностей является актуальной задачей, так как реализация данного проекта позволит повысить эффективность работы пользователей и расширить их возможности для решения задач в этой области.

# Анализ требований и проектирование программного продукта

## Анализ предметной области

Теория вероятностей – это раздел математики, что изучает закономерности случайных явлений. Расчёты в теории вероятностей применяются в различных областях науки и техники, таких как:

* Физика: статистическая механика, квантовая механика, теория информации;
* Экономика: моделирование финансовых рынков, страхование, оценка рисков;
* Биология: генетика, эпидемиология, популяционная динамика;
* Инженерия: теория надежности, теория управления, обработка сигналов;
* Искусственный интеллект: машинное обучение, распознавание образов, обработка естественного языка.

В настоящее время наблюдается рост потребности в программном обеспечении (ПО) для автоматизации расчётов в теории вероятностей. Это связано с:

* Увеличением объёма данных: современные научные исследования и инженерные задачи оперируют огромными массивами данных, обработка которых вручную невозможна;
* Сложностью расчётов: многие задачи теории вероятностей требуют сложных вычислений, которые трудно выполнить вручную;
* Необходимостью экономии времени: автоматизация расчётов позволяет экономить время и ресурсы.

Целью разработки ПО для автоматизации расчётов в теории вероятностей является:

* Автоматизация вычислений: автоматизация рутинных вычислений, освобождение времени для более творческих задач;
* Повышение точности: снижение ошибок, связанных с человеческим фактором;
* Увеличение скорости: ускорение выполнения расчётов;
* Повышение доступности: предоставление возможности использовать методы теории вероятностей более широкому кругу пользователей.

ПО для автоматизации расчётов в теории вероятностей должно обладать следующими функциональными возможностями:

* Выполнение основных расчетов: вычисление вероятностей, средних значений, дисперсий, медиан, квантилей и других статистических характеристик;
* Визуализация данных: построение графиков, диаграмм и других визуальных представлений данных;
* Статистическое моделирование: генерация случайных чисел, реализация различных стохастических процессов;
* Анализ данных: проведение статистических тестов, оценка параметров моделей;
* Экспорт данных: экспорт результатов расчётов в различные форматы.

ПО для автоматизации расчётов в теории вероятностей должно соответствовать следующим требованиям:

* Простота использования: интерфейс ПО должен быть простым и понятным для пользователей с различным уровнем подготовки;
* Наглядность: результаты расчетов должны быть представлены в наглядной форме;
* Точность: ПО должно обеспечивать высокую точность вычислений;
* Производительность: ПО должно работать быстро и эффективно;
* Гибкость: ПО должно быть достаточно гибким, чтобы его можно было использовать для решения различных задач;
* Открытость: ПО должно быть открытым, чтобы его можно было модифицировать и расширять.

Для разработки программного обеспечения, осуществляющего автоматизацию расчётов в теории вероятности, выбраны следующие задачи:

1) В магазине 5 яблок, 3 груши и 2 банана. Какова вероятность того, что покупатель, выбрав наугад один фрукт, возьмет яблоко?

2) В группе 20 человек. Из них 10 мальчиков и 10 девочек. Наугад выбирают двух человек. Какова вероятность того, что это будут мальчик и девочка.

3) В прямоугольник 5×4 см2 вписан круг радиуса 1,5 см. Какова вероятность того, что точка, случайным образом поставленная в прямоугольник, окажется внутри круга?

4) Вероятность того, что изготовленная на первом станке деталь будет первосортной, равна 0,7. При изготовлении такой же детали на втором станке эта вероятность равна 0,8. На первом станке изготовлены две детали, на втором три. Найти вероятность того, что все детали первосортные.

5) Из 1000 ламп 380 принадлежат к 1 партии, 270 – ко второй партии, остальные к третьей. В первой партии 4% брака, во второй - 3%, в третьей – 6%. Наудачу выбирается одна лампа. Определить вероятность того, что выбранная лампа – бракованная.

6) Для сигнализации об аварии установлены два независимо работающих сигнализатора. Вероятность того, что при аварии сигнализатор сработает, равна 0,95 для первого сигнализатора и 0,9 для второго. Найти вероятность того, что при аварии сработает только один сигнализатор.

7) Трое учащихся на экзамене независимо друг от друга решают одну и ту же задачу. Вероятности ее решения этими учащимися равны 0,8, 0,7 и 0,6 соответственно. Найдите вероятность того, что хотя бы один учащийся решит задачу.

## Техническое задание на разработку

Техническое задание на разработку программного обеспечения – это документ, который содержит подробное описание требований к программному продукту. Техническое задание на разработку программного обеспечения для автоматизации расчётов в теории вероятностей, составлено согласно ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» и представлено ниже.

1 Общие сведения.

а) Полное наименование автоматизированной системы (АС): «Автоматизированная система расчётов в теории вероятности»;

б) Наименование заказчика: факультет среднего профессионального образования Орского гуманитарно-технологического института(филиала) ОГУ в лице преподавателя высшей категории Михайличенко Ж. В.;

в) Наименование разработчика: студент второго курса группы 22ИСП-2 Дерипалов А. В.;

г) Документ, на основании которого создаётся АС: протокол закрепление тем курсовых работ по дисциплине «Технология разработки программного обеспечения» от 19.02.2024;

д) Дата начала работ: 19.02.2024;

е) Дата окончания работ: 10.06.2024.

2 Цели и назначение создания автоматизированной системы.

а) Цели создания АС:

* Повышение производительности и эффективности при проведении расчетов в теории вероятностей;
* Снижение вероятности ошибок за счёт автоматизации рутинных операций;
* Расширение возможностей для проведения сложных расчётов, которые трудно или невозможно выполнить вручную;
* Сокращение времени на проведение расчётов;
* Повышение доступности инструментов для расчёта вероятностей для широкого круга пользователей;
* Создание унифицированной среды для разработки и применения программного обеспечения для расчётов в теории вероятностей.

б) Назначение АС:

* Автоматизация расчёта различных вероятностных распределений, таких как биномиальное, нормальное, пуассоновское;
* Вычисление различных вероятностных характеристик, таких как математическое ожидание, дисперсия, медиана;
* Моделирование случайных величин и процессов;
* Проведение статистических тестов;
* Анализ данных;
* Визуализация результатов расчётов.

3 Характеристика объекта автоматизации.

Объект автоматизации представляет собой систему, предназначенную для выполнения расчётов в области теории вероятностей, также имеет большое значение для развития науки и техники. Система может быть использована для решения различных задач в различных областях, что может привести к улучшению качества жизни людей.

4 Требования к автоматизированной системе.

а) Требования к структуре АС в целом: разрабатываемая система должна состоять из подсистемы ввода данных (загрузка данных из различных источников, обеспечение данных), подсистемы расчётов: (реализация алгоритмов расчётов в теории вероятностей, поддержка различных методов и моделей, обеспечение точности и производительности расчётов), подсистема визуализации (предоставление результатов расчётов в удобной и понятной форме, создание графиков, диаграмм).

б) Требования к функциям выполнения АС:

* Расчёт дискретных вероятностей;
* Расчёт непрерывных вероятностей;
* Расчёт вероятностей по формулам комбинаторики;
* Расчёт условных вероятностей;
* Расчёт совместных вероятностей.

Статистический анализ:

* Вычисление выборочных средних и дисперсий;
* Оценка параметров распределений;
* Проверка статистических гипотез;
* Моделирование случайных величин.

в) Требования к видам обеспечивания АС:

* Требования к математическому обеспечению: АС должна использовать методы теории вероятностей и математической статистики, соответствующие задачам, решаемым системой; АС должна использовать алгоритмы обработки информации, обеспечивающие точность и эффективность расчётов.
* Требования к информационному обеспечению: хранение и обработка данных о теории вероятностей.
* Требования к лингвистические требованиям: АС должна иметь интерфейс на русском языке; АС должна поддерживать различные способы организации диалога с пользователем, например, графический интерфейс, диалоговый интерфейс.
* Требования к программному обеспечению: для обеспечения работы в теории вероятностей требуется использовать определенное программное обеспечение среда программирование Visual Studio 2019, построение визуальных диаграмм Ramus, документирование MS Word, руководство пользователя Dr.Explain.
* Требования к техническому обеспечению: для полноценного функционирования автоматизированной системы в области теории вероятностей необходим современный персональный компьютер с установленной операционной системой Windows 10.
* Требования к организационному обеспечению: планирование и контроль выполнения расчетов в теории вероятностей.

г) Общие технические требования к АС.

* Требования к численности и квалификации пользователей: пользователь с персональным компьютером.
* Требования к защите данных: защита на неправильный ввод символов.

5 Состав и содержание работ по созданию автоматизированной системы.

В таблице 1 представлены этапы разработки АС расчётов в теории вероятности:

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 Анализ требования | 19.02.2024 – 10.03.2024 | Анализ предметной области, изучение программных аналогов, разработки технического задания нас создании АС |
| 2 Проектирование | 10.03.2024 – 17.03.2024 | Разработка архитектуры системы, модулей системы, интерфейса пользователя |
| 3 Программирования | 17.03.2024 – 20.04.2024 | Реализация модулей системы, отладка и тестирование модулей |
| 4 Тестирование | 20.04.2024 – 21.05.2024 | Нагрузочное тестирование, интеграционное тестирование, функциональное тестирование |
| 5 Внедрение | 21.05.2024 – 08.06.2024 | Установка и настройка системы, обучение пользователей, опытная эксплуатация |
| 6 Эксплуатация и сопровождение | 10.06.2024 | Техническая поддержка пользователей, доработка и обновление системы |

1. Порядок разработки автоматизированной системы.

* Определение требований заказчиком;
* Анализ и проектирование;
* Разработка и тестирование

7 Порядок контроля и приёмки автоматизированной системы.

* Функциональное тестирование;
* Тестирование безопасности;
* Приёмка заказчиком

8 Требования к составу и содержанию работ по подготовки объекта автоматизации к вводу автоматизированной системы в действии.

* Установка программного обеспечения;
* Тестирование и проверка работоспособности.

9 Требования к документированию.

* Документирование программного кода;
* Техническое задание на разработку системы;
* Руководство пользователя системы;
* Пояснительная записка к курсовой работе.

10 Источники разработки.

* Протокол закрепление тем курсовых работ по дисциплине «Технология разработки программного обеспечения» от 19.02.2024;
* ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»
* Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с.
* ГОСТ-19.701-90 (ИСО 5807–85) «Схемы алгоритмов, программ, данных и систем».
* ГОСТ 34.601–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».

## Проектирование

Проектирование информационной системы расчётов в теории вероятностей является крайне важным, поскольку это позволяет эффективно и точно проводить расчёты электрических и магнитных процессов, которые широко применяются в различных областях науки и техники.

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ). Стандарт IDEF0 представляет организацию как набор модулей. Наиболее важная функция находится в верхнем левом углу. Модель используется при организации бизнес-процессов и проектов, основанных на моделировании всех процессов: как административных, так и организационных.

Преимущества IDEF0:

* Комплексность при декомпозиции (туннелирование стрелок обеспечивает связность создаваемых диаграмм между собой);
* Возможность агрегирования и детализации потоков данных и информации (разделение и слияние стрелок);
* Наличие жестких требований методологии, обеспечивающих получение моделей процессов стандартного вида;
* Простота документирования процессов;
* Соответствие подхода к описанию процессов в IDEF0 МС ИСО серии 9000, что позволяет выбирать IDEF0 в качестве внутреннего стандарта организации, регламентирующего описание бизнес-процессов;
* Полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи).

Ramus Educational – это бесплатный программный продукт, который может быть использован для создания диаграмм в формате IDEF0 и DFD. Формат файлов Ramus Education полностью совместим с форматом файла коммерческой версии Ramus.

Для построения функциональной модели IDEF0 разрабатываемой автоматизированной системы расчётов в теории вероятностей, прежде всего, необходимо создать контекстную диаграмму, состоящую из единственного функционального блока A0 «Разработка программного обеспечения для автоматизации расчётов в теории вероятностей».

Входными данными контекстной диаграммы являются исходные данные к задачам; выходными данными являются решённые задачи; в качестве механизмов учувствуют пользователь и персональный компьютер (ПК); управлением являются законы математики.

Контекстная диаграмма показана на рисунке 1.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 1 – Контекстная диаграмма IDEF0

Для того чтоб лучше понять функции системы необходимо провести декомпозицию контекстной диаграммы, выделив следующие управляющие функции:

A1 – «Выбор задачи» включает в себя вход, где указан список номеров задач; механизмы - пользователь и ПК; а также на выходе мы получаем номера задач.

A2 – «Ввод исходных данных к задаче» на входе подаётся условие задачи, а на выходе получаем известные значения; механизмы - пользователь и ПК.

A3 – «Выполнение расчётов по формулам» на вход подаются формулы и известные значения, а на выходе результат расчётов; механизм – ПК; управление – законы математики.

A4 – «Вывод результатов» на вход подаётся результаты расчёта, на выходе решённые задачи; механизм – ПК.

Диаграмма декомпозиции IDEF0 первого уровня для показана на рисунке 2.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 2 – Диаграмма декомпозиции первого уровня IDEF0

Рассмотрим декомпозицию функционального блока А3 «Выполнение расчётов по формулам» на примере задачи №1. Для этого выделим следующие блоки декомпозиции второго уровня:

A31 – «Анализирование условия задачи» включает в себя вход, где указаны исходные данные к задаче, механизмы – пользователь и ПК, они используются и в следующем блоке.

A32 – «Ввод числовых значений» получаем данные к задаче для дальнейшего решения, на вход подаётся известные значения задачи, на выходе подстановка переменных.

A33 – «Подставление формулы для задачи по астрономии» на входе получаем подстановку переменных, а на выходе проверку введённых данных к задаче, механизм – ПК.

A34 – «Выполнение расчётов в теории вероятностей» получаем решенную задачу, на вход подаётся проверка введённых данных к задаче, на выходе результаты расчётов, механизм – ПК.

Диаграмма декомпозиции IDEF0 второго уровня для показана на рисунке 3.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 3 – Диаграмма декомпозиции второго уровня IDEF0

Все диаграммы, построенные в результате проектирования, будут использованы при разработке программного обеспечения системы расчётов в теории вероятностей.

# Разработка и тестирование программного продукта

## Обоснование программных средств реализации

Средством реализации разрабатываемого программного обеспечения для автоматизации расчётов в теории вероятности является язык программирования C# и среда программирования Visual Studio 2022.

C# — это объектно-ориентированный язык программирования, разработанный компанией Microsoft. Он является частью платформы .NET Framework и используется для создания различных типов приложений, включая веб-приложения, настольные приложения, игры и мобильные приложения.

Возможности и преимущества С#:

1. Объектно-ориентированный: C# поддерживает объектно-ориентированное программирования, которое позволяет создавать модульные и повторно используемые программы.
2. Безопасный: C# — это безопасный язык программирования, который помогает предотвращать ошибки, такие как нулевые ссылки и переполнение буфера.
3. Многофункциональный: C# можно использовать для разработки различных типов приложений, включая приложения для Windows, веб-приложения, игры, мобильные приложения.
4. Производительный: C# — это высокопроизводительный язык программирования, который может генерировать эффективный машинный код.
5. Простота изучения и использования: C# — это относительно простой язык для изучения и использования. Он имеет понятный синтаксис и множество библиотек и фреймворков, которые могут помочь вам начать работу.

Visual Studio — это интегрированная среда разработки (IDE), созданная компанией Microsoft. Она используется для разработки приложений на различных языках программирования, включая C#, C++, Visual Basic и F#.

Возможности и преимущества Visual Studio:

1. Редактор кода: Visual Studio имеет мощный редактор кода, который поддерживает множество функций, таких как подсветка синтаксиса, автозаполнение кода и отладка.
2. Инструменты разработки: Visual Studio включает в себя множество инструментов разработки, которые могут помочь вам в разработке приложений, таких как дизайнер пользовательского интерфейса, отладчик и тестировщик.
3. Расширяемость: Visual Studio можно расширить с помощью плагинов, которые добавляют новые функции и возможности.
4. Профессиональная IDE: Visual Studio — это профессиональная IDE, которая предлагает широкий спектр функций для разработки приложений.
5. Интеграция с .NET: Visual Studio тесно интегрирована с платформой .NET, что делает её идеальной средой разработки для приложений .NET.

C# и Visual Studio являются идеальным выбором для разработки программного обеспечения для автоматизации расчётов в теории вероятности. Они обладают всеми необходимыми возможностями для создания надежных, эффективных и масштабируемых приложений.

## Разработка пользовательского интерфейса

Пользовательский интерфейс (UI – User Interface) программного обеспечения – это точка соприкосновения между пользователем и программой. Он представляет собой набор элементов, с помощью которых пользователь взаимодействует с программой, получая от неё информацию и управляя её работой.

Особенности пользовательского интерфейса:

1. Интерактивность: UI должен обеспечивать взаимодействие между пользователем и программой. Это означает, что пользователь должен иметь возможность отдавать команды и получать мгновенную обратную связь.
2. Удобство использования: UI должен быть удобным и простым в использовании. Пользователь должен легко ориентироваться в нем и выполнять необходимые действия без особых усилий.
3. Эстетичность: UI должен быть эстетичным и привлекательным. Это повышает его привлекательность для пользователей и делает их работу с программой более приятной.
4. Эффективность: UI должен быть эффективным и помогать пользователю быстро и легко достигать своих целей.
5. Соответствие платформе: UI должен быть разработан с учётом платформы, на которой работает программа.

Существует ряд требований, которым должен соответствовать UI:

1. Простота использования: UI должен быть интуитивно понятным и лёгким для освоения.
2. Наглядность: интерфейс должен отображать информацию и обеспечивать лёгкое восприятие.
3. Гибкость: UI должен быть гибким и адаптивным к различным устройствам и разрешениям экрана.
4. Обратная связь: UI должен предоставлять пользователю обратную связь о результатах его действий.
5. Эффективность: интерфейс должен помогать пользователям выполнять задачи быстро и без лишних усилий.

Важной частью UI является его доступность, пользовательский интерфейс должен быть доступен для всех пользователей, включая людей с ограниченными возможностями. Это означает, что UI должен быть разработан с учётов использования вспомогательных технологий, таких как скринридеры. Таким образом, пользовательский интерфейс должен быть эстетичным и соответствовать общему стилю продукта.

Программный интерфейс автоматизированной системы расчётов в теории вероятности должен соответствовать всем вышеперечисленным требованиям. Структура программного интерфейса показана на рисунке 4.



Рисунок 4 – Структура основного программного интерфейса

Кнопки расположенные на основном программном интерфейсе окна отвечает за открытие следующих окон с задачами, которые соответствуют им по названию.

Окно с Задачей №1 показано на рисунке 5.

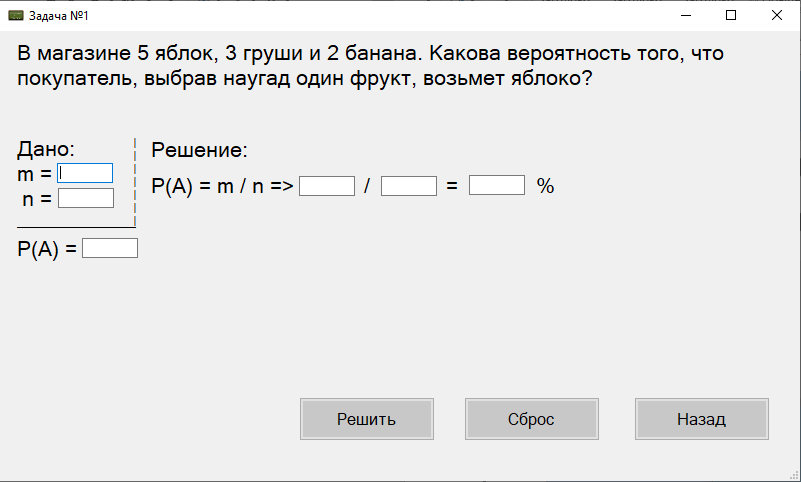


Рисунок 5 – Окно с Задачей №1

В окне «Задача №1» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №2 показано на рисунке 6.

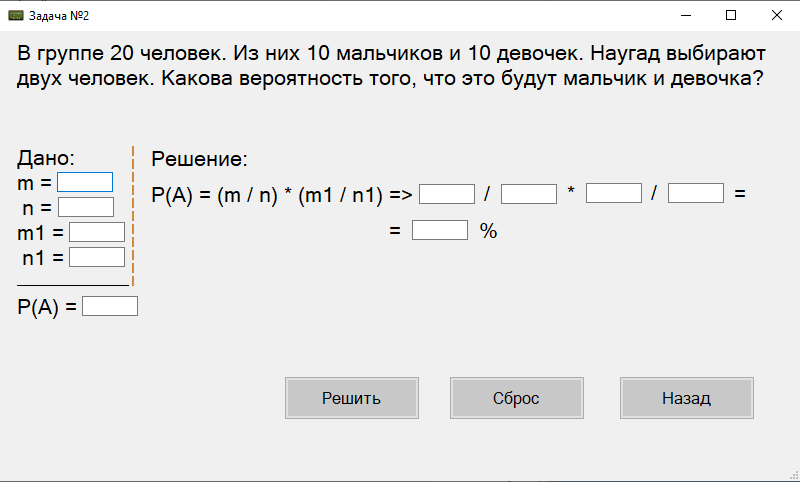


Рисунок 6 – Окно с Задачей №2

В окне «Задача №2» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №3 показано на рисунке 7.

В окне «Задача №3» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №4 показано на рисунке 8.

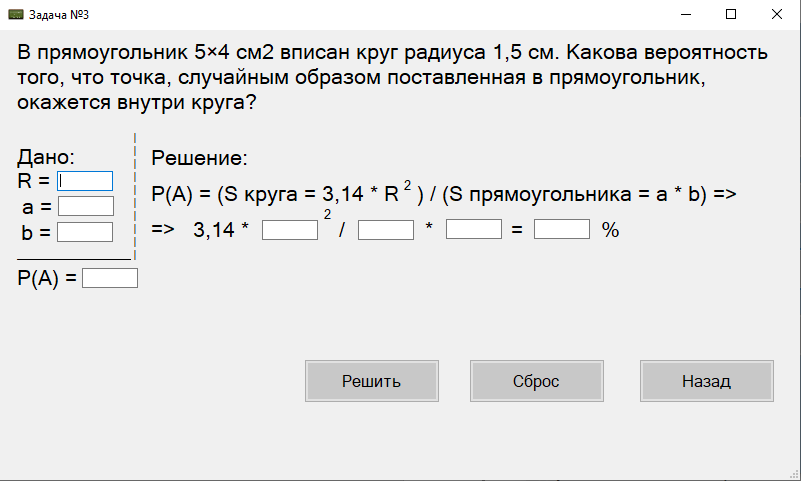


Рисунок 7 – Окно с Задачей №3

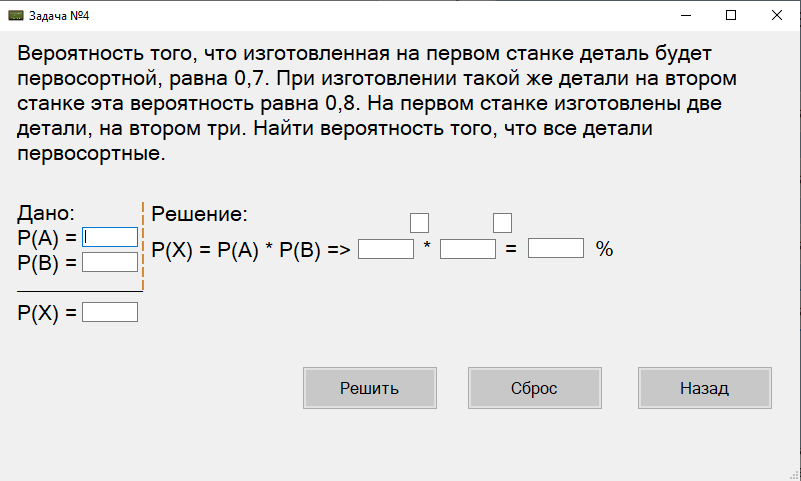


Рисунок 8 – Окно с Задачей №4

В окне «Задача №4» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №5 показано на рисунке 9.

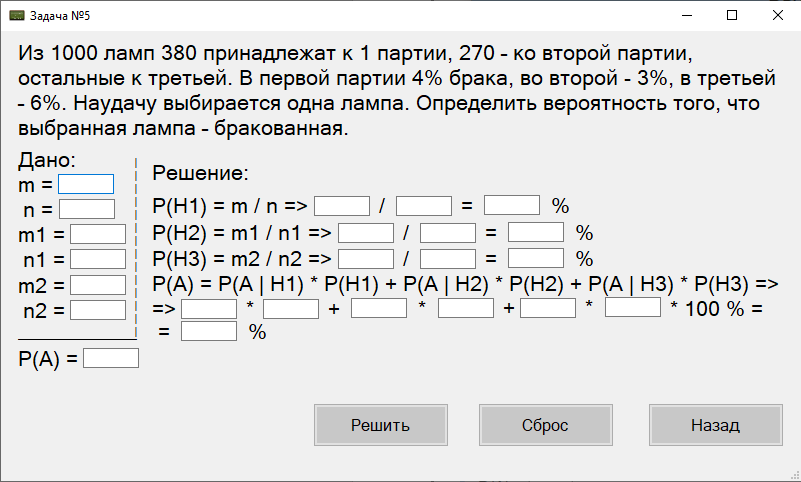


Рисунок 9 – Окно с Задачей №5

В окне «Задача №5» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №6 показано на рисунке 10.

В окне «Задача №6» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

Окно с Задачей №7 показано на рисунке 11.

В окне «Задача №7» находится условие задачи, поля для ввода данных, формула для этой задачи, а также кнопки: кнопка «Назад» отвечает за переход на главное окно где можно выбрать задачи, кнопка «Решить» отвечает за произведения расчётов исходя из введённых данных для данной задачи, кнопка «Сброс» отвечает за сброс всех значений.

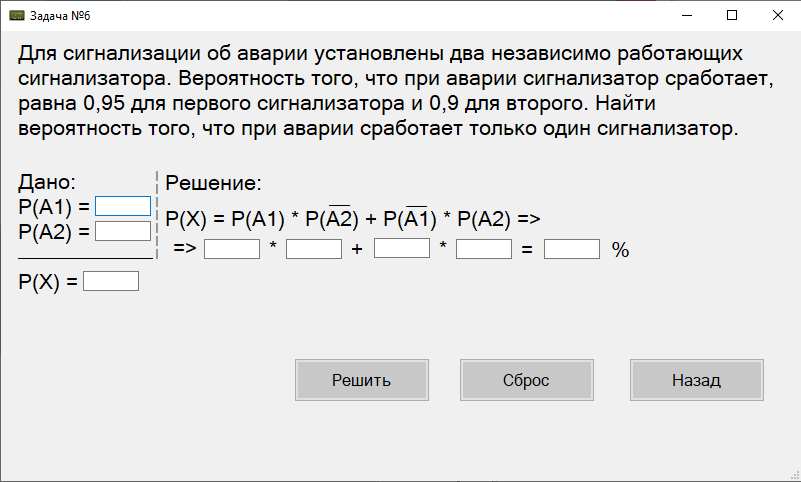


Рисунок 10 – Окно с Задачей №6

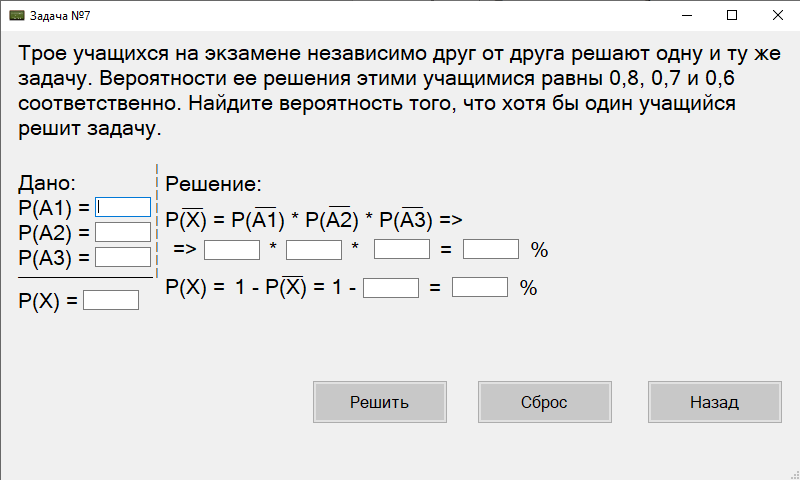


Рисунок 11 – Окно с Задачей №7

## Алгоритмизация и программирование

Алгоритм – это последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу. Алгоритм должен описываться на формальном языке, исключающем неоднозначность толкования.

Алгоритм обладает следующими свойствами:

1. Дискретность. Процесс решения задачи должен быть разбит на последовательность отдельных шагов-команд, которые выполняются одна за другой. Только после завершения одной команды начинается выполнение следующей.
2. Понятность. Алгоритм должен содержать только те команды, которые известны исполнителю.
3. Детерминированность. Каждый шаг и переход от шага к шагу должны быть точно определены, чтобы его мог выполнить любой другой человек или механическое устройство. У исполнителя нет возможности принимать самостоятельное решение (алгоритм исполняется формально).
4. Конечность. Обычно предполагают, что алгоритм заканчивает работу за конечное число шагов. Результат работы алгоритма также должен быть получен за конечное время. Можно расширить понятие алгоритма до понятия процесса, который по различным каналам получает данные, выводит данные и потенциально может не заканчивать свою работу.
5. Массовость. Алгоритм должен решать не одну частную задачу, а класс задач. Не имеет смысла строить алгоритм нахождения наибольшего общего делителя только для чисел 10 и 15.

Алгоритмы чаще всего представляются в виде блок-схем по стандарту: ГОСТ 19.701–90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения».

Для демонстрации алгоритмов в виде блок-схем выбраны задачи № 1, 2, 4.

Задача №1. В магазине 5 яблок, 3 груши и 2 банана. Какова вероятность того, что покупатель, выбрав наугад один фрукт, возьмет яблоко?

Задача №2. В группе 20 человек. Из них 10 мальчиков и 10 девочек. Наугад выбирают двух человек. Какова вероятность того, что это будут мальчик и девочка.

Задача №4. Вероятность того, что изготовленная на первом станке деталь будет первосортной, равна 0,7. При изготовлении такой же детали на втором станке эта вероятность равна 0,8. На первом станке изготовлены две детали, на втором три. Найти вероятность того, что все детали первосортные.

Блок-схемы алгоритмов решения задач №1, 2, 4 показаны на рисунках 12, 13, 14.

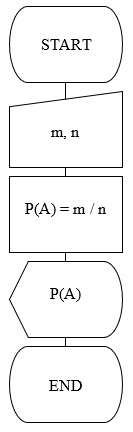


Рисунок 12 – Блок-схема алгоритма решения задачи №1

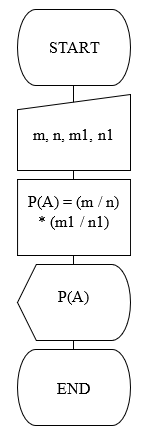


Рисунок 13 – Блок-схема алгоритма решения задачи №2

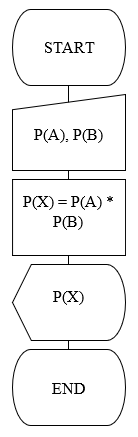


Рисунок 14 – Блок-схема алгоритма решения задачи №4

Для программной реализации разработаны следующие методы для выполнения функции автоматизированной системе расчётов в теории вероятностей.

1) textBox1\_TextChanged, textBox2\_TextChanged, textBox3\_TextChanged, textBox4\_TextChanged - эти методы вызываются при изменении текста в соответствующих текстовых полях textBox1, textBox2, textBox3 и textBox4.

2) button1\_Click: - этот метод вызывается при нажатии на кнопку button1. Внутри метода происходит вычисление значения по заданной формуле и отображение результата на форме Задачи №1.

3) button2\_Click: - этот метод вызывается при нажатии на кнопку button2. Внутри метода происходит очистка всех текстовых полей на форме.

4) button3\_Click: - этот метод вызывается при нажатии на кнопку button3. Внутри метода происходит скрытие текущей формы и переход к главной форме.

## Тестирование

Тестирование ПО – важный этап производства ПО, направленный на детальное исследование программного кода и выявление ошибок в работе систем.

Цели тестирования информационной системы в теории вероятностей:

* Проверка корректности вычислений: убедиться, что система правильно рассчитывает задачи связанные с вероятностостью.
* Оценка точности и надежности: определить, насколько точно система производит расчёты по сравнению с эталонными значениями.
* Тестирование функциональности: проверить, что система корректно реализует все необходимые функции для работы с теорией вероятностей, расчет вероятностей.
* Анализ производительности: оценить, насколько эффективно система справляется с вычислительной нагрузкой, особенно при работе с большими объемами данных.
* Проверка интерфейса и удобства использования: убедиться, что пользовательский интерфейс системы интуитивно понятен и удобен для работы с вероятностными расчётами, оценить, насколько легко пользователи могут выполнять типовые операции в системе, комплексное тестирование системы по этим направлениям позволяет обеспечить ее надежность, точность и функциональность в области теории вероятностей.

Виды тестирования, которые могут использоваться при приёмке информационной системы, включают в себя:

1. Функциональное тестирование:

* Проверка правильности обработки входных данных и корректности вывода результатов.
* Проверка соответствия функциональности системы заявленным требованиям.

1. Компонентное (модульное) тестирование:

* Проверка взаимодействия компонентов между собой.
* Тестирование отдельных компонентов (модулей) системы на корректность их работы.

1. Интеграционное тестирование:

* Тестирование взаимодействия между различными частями системы.
* Проверка работы системы в целом после объединения всех компонентов.

1. Нагрузочное тестирование:

* Оценка производительности системы при различных нагрузках.
* Проверка стабильности работы системы при высоких нагрузках.

1. Решение эргономика решений (интерфейсов):

* Проверка соответствия интерфейса стандартам и требованиям пользователя.
* Оценка удобства использования пользовательского интерфейса.

1. Тестирование надёжности:

* Тестирование на наличие ошибок и их обработку.
* Оценка способности системы к восстановлению после сбоев.
* Проверка стабильности работы системы в различных условиях.

Результаты приёмки АС расчётов в теории вероятностей представлено в таблице 2.

Таблица 2 – Результат приёмки АС

|  |  |
| --- | --- |
| Вид тестирования | Оценка по десятибалльной шкале |
| 1. Функциональное |  |
| 1. Компонентное (модульное) |  |
| 1. Интеграционное |  |
| 1. Нагрузочная |  |
| 1. Эргономика решений (интерфейсы) |  |
| 1. Надёжность |  |

# Рекомендации по внедрению, эксплуатации и сопровождению программного продукта

## Руководство пользователя

Обзор возможностей программы

Программа для решения задач в теории вероятностей — это программа, которая помогает решать задачи при помощи формул.

Расчёты в теории вероятностей предназначены для:

Типовые задачи, решаемые с помощью Расчётов в теории вероятностей:

Основными выгодами от использования Расчётов в теории вероятностей являются:

Системные требования

Для стабильной и эффективной работы расчётов в теории вероятностей рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): 2.5 GHz

Количество ядер процессора (CPU): 4

Объем оперативной памяти (RAM): 16 GB

Объем свободного места на диске (HDD): 3 GB

Операционная система (OS): Windows 11, Windows 10

Браузер: Google Chrome, Microsoft Edge

Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с расчётами в теории вероятностей.

Перед началом работы в автоматизированной системе для расчётов в теории вероятностей рекомендуем ознакомиться с основными понятиями и терминами:

Основные понятия и термины

Перед началом работы с расчётами в теории вероятностей рекомендуем ознакомиться с основными понятиями и терминами:

Понятие 1

Теория вероятностей – это раздел математики, что изучает закономерности случайных явлений. Расчёты в теории вероятностей применяются в различных областях науки и техники, таких как:

* Физика: статистическая механика, квантовая механика, теория информации;
* Экономика: моделирование финансовых рынков, страхование, оценка рисков;
* Биология: генетика, эпидемиология, популяционная динамика;
* Инженерия: теория надежности, теория управления, обработка сигналов;
* Искусственный интеллект: машинное обучение, распознавание образов, обработка естественного языка.

Понятие 2

Статистическая механика – это механика больших ансамблей относительно простых систем, таких как атомы в кристалле, молекулы в газе, фотоны в лазерном пучке, звёзды в галактике, автомобили на шоссе.

Понятие 3

Моделирование финансовых рынков – это процесс создания математических моделей, которые представляют финансовые аспекты бизнеса или проекта.

Понятие 4

Популяционная динамика – это изменение её биологических и групповых свойств во времени.

Понятие 5

Теория надежности – это наука, изучающая закономерности распределения отказов технических устройств и конструкций, причины и модели их возникновения.

Понятие 6

Машинное обучение – это наука о разработке алгоритмов и статистических моделей, которые компьютерные системы используют для выполнения сложных задач без четких инструкций.

Понятие 7

Компьютер – представляет собой электронное устройство, которое работает с информацией и данными.

Понятие 8

Программа – это последовательность машинных команд, предназначенная для достижения конкретного результата.

Понятие 9

Visual Studio – это интегрированная среда разработки, является творческой стартовой площадкой, которую можно использовать для редактирования, отладки и сборки кода, а также для публикации приложения.

Понятие 10

Программный код — это набор инструкций для компьютера. Его пишут на языке программирования сами разработчики или генерируют автоматически. С помощью кода создают программы: отдают компьютеру команды, которые он выполняет.

Понятие 11

Ramus — это программа для построения визуальных диаграмм, используемых для наглядного отображения различных бизнес-процессов.

Понятие 12

Dr.Explain — это приложение для быстрого создания файлов справки (help-файлов), справочных систем, online руководств пользователя, пособий и технической документации к программному обеспечению и техническим системам.

Документация — это совокупность документов, посвященных какому-либо вопросу (задаче, проекту, изделию и др.). Документирование — это процесс отбора, классификации, использования и распространения документов.

Установка

Для установки Visual Studio, пожалуйста, загрузите дистрибутива последней версии 5.7.1, доступный по адресу <https://learn.microsoft.com/ru-ru/visualstudio/releases/2019/release-notes-v16.10>

Перед установкой ознакомьтесь с системными требованиями и лицензионным соглашением.

В процессе установки, пожалуйста, убедитесь, что системные требования соответствует программному продукту.

Настройка

Для начала работы в Visual Studio рекомендуем предварительно выполнить следующие настройки окружения:

1. Установить Visual Studio
2. Настроить Visual Studio

Запуск

Для запуска Расчётов в теории вероятностей нажмите на ярлык программы «Visual Studio» в меню Пуск либо наберите в командной строке «Visual Studio».

При первом запуске программы «Открывается главное окно программы с интерфейсом»

Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса расчётов в теории вероятностей: основных режимов работы, предназначение окон и экранов, доступные операции.

Главное окно программы

Главное окно программы Расчётов в теории вероятностей представлено на рисунке 15 и позволяет выполнять следующие операции:

* Открыть задачи по теории вероятностей
* Расчёты по формулам



Рисунок 15 – Главное окно программы

Настройки программы

Для начала работы в теории вероятностей следует использовать следующие настройки окружения:

1. Создать проект в среде Visual Studio;
2. Распаковать файлы курсовой в папке проекта;
3. Завершить проект;

Режимы работы с задачами по молекулярной физики

Пользовательский интерфейс расчётов молекулярной физики обеспечивает работу в нескольких режимах: Программа работает только в режиме пользователя. Нет ограничений.

Режимы работы с задачами по молекулярной физики

Пользовательский интерфейс расчётов молекулярной физики обеспечивает работу в нескольких режимах: Программа работает только в режиме пользователя. Нет ограничений.

Работа с кнопками

Данный раздел описывает работу с кнопками программы

В частности, рассматриваются наиболее частые операции:

* Кнопка «Задача 1–7», при нажатии на неё откроется новое окно, в котором можно будет решить задачу по формуле;
* Кнопка «Назад», при нажатии на неё переносит на главную форму;
* В окнах «Задача 1–7», есть кнопки "Сброс" "Решение" "Назад"
* Кнопка «Сброс» после нажатия этой кнопки сбросится все введённые данные и решение

Горячие клавиши

Следующий раздел содержит все сочетания клавиш и способы управления при помощи мыши, поддерживаемые в программном продукте.

Общие

F5 - запуск программы.

Редактирование

Shift + Delete – вырезать.

Shift + Insert – вставить.

Ctrl + C – копировать.

Ctrl + V – вставить.

Ctrl + A – выбрать все.

Ctrl + X – вырезать.

Сtrl + Z – отменить.

Примеры использования

В данном разделе собраны примеры реального использования расчёта молекулярной физики, демонстрирующие применения продукта в различных отраслях.

Работа с задачами

Задача

В магазине 5 яблок, 3 груши и 2 банана. Какова вероятность того, что покупатель, выбрав наугад один фрукт, возьмет яблоко?

Проблема

Если пользователь вводит некорректные данные, например символы букв, «-», «\_», «!», «?», то строки с данными не будут выводить результат.

Результат

После того как пользователь заполнил все данные поля данными

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 16 – Окно Задачи для ввода данных

Ему следует нажать кнопку «Рассчитать», после чего программ решит данную задачу по формуле.

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Автоматически созданное описание

Рисунок 17 – Окно Задачи с решением

Вывод

На примере этой задачи можно сделать вывод что программа способна решать задачи по теории вероятностей по формулам.

Устранение типовых проблем

Описание проблемы ввод неправильные данных

Решение: перезапустить программу. И ввести правильные данные задачи

Частые вопросы (FAQ)

Как открыть программу?

Ответ: чтобы открыть программу, нужно запустить Visual Studio 2019

Как скачать программу?

Ответ: найти в браузере Visual Studio 2019 community.

Если вы не нашли ответа на свой вопрос, пожалуйста, свяжитесь с нами, написав на почту: jusedruwotblitz@gmail.com.

Контактная информация

Расчёты в теории вероятностей разрабатывается и поддерживается компанией 22ИСП-2, являющейся правообладателем.

Техническая поддержка

Вы можете направить вопросы по функциональности программы расчётов в теории вероятностей следующими способами:

* Email: jusedruwotblitz@gmail.com
* Телефон: +7 961 904 70-54
* Мессенджеры: ВК: <https://vk.com/tyomalaw>
* Форма обратной связи: Заочная.
* Сайт проекта: <https://github.com/godbtw/1337>

## План внедрения и сопровождения

Внедрение информационной системы – процесс настройки ПО и технических средств под определёнными условиями использования, а также обучение пользователей работе системы.

Сопровождение информационной системы – процесс улучшения оптимизации и устранение дефектов ПО после передачи информационной системы на эксплуатацию.

Цели внедрения и сопровождения:

* Улучшение скорости и точности расчётов: целью внедрения информационной системы может быть повышение скорости и точности вычислений необходимых для решения задач на вероятности. Это поможет пользователям проводить расчёты за меньшее время.
* Автоматизация процесса расчётов: информационная система может быть разработана для автоматизации процесса расчётов в теории вероятностей, упрощая работу и снижая вероятность ошибок.
* Повышение удобства использования: целью сопровождения системы может быть обеспечение удобного и интуитивно понятного интерфейса для пользователей, что сделает работу с программой более эффективной.
* Обеспечение надёжности и безопасности данных: важной целью сопровождения информационной системы является обеспечение надёжности хранения данных, а также защиты информации от утечек и несанкционированного доступа.
* Постоянное обновление и развитие: для системы, используемой в расчётах связанной с теорией вероятностей, важно постоянное обновление и развитие функциональности, чтобы она соответствовала современным требованиям и стандартам.
* Обучение пользователей: важной целью сопровождения является обучение пользователей работе с системой, чтобы они могли максимально эффективно использовать её возможности для своих целей.

В таблице 3 представлен план внедрения и сопровождения расчётов в теории вероятностей.

Таблица 3 – План внедрения и сопровождения

|  |  |
| --- | --- |
| Мероприятия | Сроки в днях |
| Подготовка оборудования (технического обеспечения) | 3 |
| Установка необходимой ОС | 1 |
| Установка (инсталляция ПО/ИС) | 2 |

Продолжение Таблицы 3

|  |  |
| --- | --- |
| Мероприятия | Сроки в днях |
| Настройка ПО под запросы пользователя | 3 |
| Обучение персонала работе с ИС (знакомство с руководством пользователя) | 2 |
| Подписание акта о сдачи ИС в эксплуатации | 1 |
| Мониторинг и анализ результатов | 256 |
| Обновление данных и параметров модели | 256 |
| Контроль качества и верификация | 256 |
| Оптимизация методов и алгоритмов | 256 |

# Заключение

Разработано программное обеспечение, которое автоматизирует расчёты в теории вероятностей для упрощения работы и повышения точности результатов.

Для достижения поставленной цели были решены следующие цели и задачи:

1. Были исследованы анализы существующих решений:

* Изучены существующие программные решения для расчётов в теории вероятностей;
* Определены их преимущества и недостатки;
* Выявлены потребности пользователей в данном ПО.

1. Было исследовано проектирование ПО:

* Определены функциональные возможности ПО;
* Разработал архитектуру ПО.

1. Была исследована разработка ПО:

* Реализовал функциональные возможности ПО;
* Обеспечил удобный интерфейс пользователя;
* Провёл тестирование ПО.

Разработка программного обеспечения для автоматизации расчётов в теории вероятностей является актуальной задачей, так как реализация данного проекта позволит повысить эффективность работы пользователей и расширить их возможности для решения задач в этой области.

1. Было исследовано, что ручные вычисления в теории вероятностей могут быть трудоемкими, утомительными и подверженными ошибкам, поэтому, было создано программное обеспечение для расчётов в теории вероятностей, которое:

* Автоматизирует рутинные задачи;
* Ускоряет процесс вычислений;
* Повышает точность и надежность результатов;
* Освобождает время для более сложных задач.

# Список использованных источников

1. "Проектирование автоматизированных систем" - А. А. Лебедев.
2. "Разработка автоматизированных систем" - В. В. Бойко, В. М. Партыка.
3. "Теория вероятностей и математическая статистика" - Г. М. Гнеденко, Ю. К. Беляев, А. Д. Соловьев.
4. Автоматизация [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F>
5. Задачи по теории вероятностей [Электронный ресурс] – Режим доступа: <https://www.matburo.ru/ex_subject.php?p=tv>
6. Методы повышения точности измерений [Электронный ресурс] – Режим доступа: <https://info.metrologu.ru/spravochnik/metrologiya/izmereniya/povishenie_tochnosti_izmereniy.html>
7. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург : Изд-во ОГУ, 2015. – 89 с.
8. Теория вероятностей [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D0%BE%D1%8F%D1%82%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9>

# Приложение А

**Текст программы**

Код главного окна программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

Form2 a = new Form2();

a.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

Form3 b = new Form3();

b.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

Form4 с = new Form4();

с.ShowDialog();

}

private void button4\_Click(object sender, EventArgs e)

{

Form5 d = new Form5();

d.ShowDialog();

}

private void button7\_Click(object sender, EventArgs e)

{

Form8 g = new Form8();

g.ShowDialog();

}

private void button6\_Click(object sender, EventArgs e)

{

Form7 f = new Form7();

f.ShowDialog();

}

private void button5\_Click(object sender, EventArgs e)

{

Form6 q = new Form6();

q.ShowDialog();

}

}

}

Код программы задачи №1:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

double m = double.Parse(textBox1.Text);

double n = double.Parse(textBox2.Text);

double result = Math.Round(m / n, 2);

textBox6.Text = result.ToString();

textBox3.Text = result.ToString();

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void Form2\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

}

}

Код программы задачи №2:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp

{

public partial class Form3 : Form

{

public Form3()

{

InitializeComponent();

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox10\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox7.Text;

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox8.Text;

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox10.Clear();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox7.Text;

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox8.Text;

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void Form3\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

private void button3\_Click(object sender, EventArgs e)

{

double m = double.Parse(textBox1.Text);

double n = double.Parse(textBox2.Text);

double m1 = double.Parse(textBox7.Text);

double n1 = double.Parse(textBox8.Text);

double result = Math.Round((m / n) \* (m1 / n1), 2);

textBox6.Text = result.ToString();

textBox3.Text = result.ToString();

}

}

}

Код программы задачи №3:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowsFormsApp

{

public partial class Form4 : Form

{

public Form4()

{

InitializeComponent();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox9.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox8.Text = textBox2.Text;

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox7.Text;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

textBox8.Text = textBox2.Text;

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox7.Text;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void Form4\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

private void button3\_Click(object sender, EventArgs e)

{

double R = double.Parse(textBox1.Text);

double a = double.Parse(textBox2.Text);

double b = double.Parse(textBox7.Text);

double Pi = Math.PI;

double result = Math.Round((Pi \* Math.Pow(R, 2)) / (a \* b), 3);

textBox6.Text = result.ToString();

textBox3.Text = result.ToString();

}

}

}

Код программы задачи №4:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowsFormsApp

{

public partial class Form5 : Form

{

public Form5()

{

InitializeComponent();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

textBox6.Text = textBox3.Text;

}

private void Form5\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

private void button3\_Click(object sender, EventArgs e)

{

double A = double.Parse(textBox1.Text);

double B = double.Parse(textBox2.Text);

double A1 = double.Parse(textBox7.Text);

double B1 = double.Parse(textBox8.Text);

double result = Math.Round(Math.Pow(A, A1) \* Math.Pow(B, B1), 3);

textBox6.Text = result.ToString();

textBox3.Text = result.ToString();

}

}

}

Код программы задачи №5:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp

{

public partial class Form6 : Form

{

public Form6()

{

InitializeComponent();

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox10.Clear();

textBox11.Clear();

textBox12.Clear();

textBox13.Clear();

textBox14.Clear();

textBox15.Clear();

textBox16.Clear();

textBox17.Clear();

textBox18.Clear();

textBox19.Clear();

textBox20.Clear();

textBox21.Clear();

textBox22.Clear();

textBox23.Clear();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox4.Text = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox5.Text = textBox2.Text;

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

textBox13.Text = textBox7.Text;

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox8.Text;

}

private void textBox12\_TextChanged(object sender, EventArgs e)

{

textBox16.Text = textBox12.Text;

}

private void textBox11\_TextChanged(object sender, EventArgs e)

{

textBox15.Text = textBox11.Text;

}

private void textBox16\_TextChanged(object sender, EventArgs e)

{

textBox16.Text = textBox12.Text;

}

private void textBox15\_TextChanged(object sender, EventArgs e)

{

textBox15.Text = textBox11.Text;

}

private void textBox13\_TextChanged(object sender, EventArgs e)

{

textBox13.Text = textBox7.Text;

}

private void textBox10\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox8.Text;

}

private void Form6\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

private void button3\_Click(object sender, EventArgs e)

{

double m = double.Parse(textBox1.Text);

double n = double.Parse(textBox2.Text);

double m1 = double.Parse(textBox7.Text);

double n1 = double.Parse(textBox8.Text);

double m2 = double.Parse(textBox12.Text);

double n2 = double.Parse(textBox11.Text);

double AH1 = double.Parse(textBox19.Text);

double AH2 = double.Parse(textBox17.Text);

double AH3 = double.Parse(textBox21.Text);

double resultH1 = m / n;

double resultH2 = m1 / n1;

double resultH3 = m2 / n2;

double resultAH1 = AH1 / 100;

double resultAH2 = AH2 / 100;

double resultAH3 = AH3 / 100;

double resultA = Math.Round(((resultAH1 \* resultH1) + (resultAH2 \* resultH2) + (resultAH3 \* resultH3)) \* 100, 2);

textBox6.Text = resultH1.ToString();

textBox18.Text = resultH1.ToString();

textBox9.Text = resultH2.ToString();

textBox22.Text = resultH2.ToString();

textBox14.Text = resultH3.ToString();

textBox20.Text = resultH3.ToString();

textBox23.Text = resultA.ToString();

textBox3.Text = resultA.ToString();

}

}

}

Код программы задачи №6:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowsFormsApp

{

public partial class Form7 : Form

{

public Form7()

{

InitializeComponent();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox1.Text;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox2.Text;

}

private void textBox9\_TextChanged(object sender, EventArgs e)

{

textBox9.Text = textBox1.Text;

}

private void textBox10\_TextChanged(object sender, EventArgs e)

{

textBox10.Text = textBox2.Text;

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox10.Clear();

}

private void Form7\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

private void button3\_Click(object sender, EventArgs e)

{

double A1 = double.Parse(textBox1.Text);

double A2 = double.Parse(textBox2.Text);

double \_A1 = 1 - A1;

double \_A2 = 1 - A2;

double result = Math.Round((A1 \* \_A2) + (\_A1 \* A2), 2);

textBox7.Text = \_A1.ToString();

textBox8.Text = \_A2.ToString();

textBox6.Text = result.ToString();

textBox3.Text = result.ToString();

}

}

}

Код программы задачи №7:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace WindowsFormsApp

{

public partial class Form8 : Form

{

public Form8()

{

InitializeComponent();

}

private void button5\_Click(object sender, EventArgs e)

{

Form1 form1 = new Form1();

this.Close();

}

private void button4\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

textBox3.Clear();

textBox4.Clear();

textBox5.Clear();

textBox6.Clear();

textBox7.Clear();

textBox8.Clear();

textBox9.Clear();

textBox10.Clear();

}

private void button3\_Click(object sender, EventArgs e)

{

double a1 = double.Parse(textBox1.Text);

double a2 = double.Parse(textBox2.Text);

double a3 = double.Parse(textBox4.Text);

double resultA1 = 1 - a1;

double resultA2 = 1 - a2;

double resultA3 = 1 - a3;

double result = Math.Round(resultA1 \* resultA2 \* resultA3, 3);

double result1 = 1 - result;

textBox6.Text = result.ToString();

textBox5.Text = result.ToString();

textBox10.Text = result1.ToString();

textBox3.Text = result1.ToString();

textBox9.Text = resultA1.ToString();

textBox8.Text = resultA2.ToString();

textBox7.Text = resultA3.ToString();

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(e.KeyChar >= '0' && e.KeyChar <= '9' || (int)e.KeyChar == 8 || (int)e.KeyChar == ','))

e.KeyChar = (char)0;

}

}

}