Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и технологий

Кафедра «Теплофизика и информатика в металлургии»

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

Бурдакова Владислава Юрьевича

Зав. кафедрой Н.А. Спирин

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**Разработка системы управления разрывной испытательной машиной на базе открытых систем**

ВКР 09.03.02 000 000 708 ПЗ

Руководитель

доцент, к.т.н. Е.А. Девятых

(должность, уч. ст., зван.) (подпись) (расшифровка подписи)

Нормоконтролер

доцент, к.т.н. Е.В. Киселев

(должность, уч. ст., зван.) (подпись) (расшифровка подписи)

Студент

НМТ-493907 В.Ю. Бурдаков

(группа) (подпись) (расшифровка подписи)

Екатеринбург,

2023 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и технологий

Кафедра Теплофизика и информатика в металлургии

Направление (бакалавриат) 09.03.02 «Информационные системы и технологии»

Образовательная программа 09.03.02/33.11 «Информационные системы и технологии в металлургии»

УТВЕРЖДАЮ:

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Н.А. Спирин)

(подпись)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**ЗАДАНИЕ**

на выполнение выпускной квалификационной работы

**студенту** Бурдакову Владиславу Юрьевичу **группы** НМТ-493907

(фамилия, имя, отчество)

**1. Тема ВКР** Разработка системы управления разрывной испытательной машиной на базе открытых систем

Утверждена распоряжением по институту от «11» января 2023 г. № 33.20-05/02

**2. Руководитель** Девятых Е.А., доцент, к.т.н.

(Ф.И.О., должность, ученое звание, ученая степень)

**3. Исходные данные к работе**

**4 Содержание ВКР**

**5. Перечень демонстрационных материалов**

**6. Календарный план**

|  |  |  |
| --- | --- | --- |
| Наименование этапов выполнения  работы | Срок выполнения  этапов работы | Отметка  о выполнении |
| Разработка проектной части | до 17.05.2023 г. |  |
| Оформление пояснительной записки | до 01.06.2023 г. |  |
| Создание демонстрационных материалов | до 01.06.2023 г. |  |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Е.А. Девятых)

(подпись) (расшифровка подписи)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (В.Ю. Бурдаков)

(подпись) (расшифровка подписи)

**7. Выпускная работа закончена** «01» июня 2023 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Е.А. Девятых)

(подпись) (расшифровка подписи)

**8. Допустить** Бурдакова Владислава Юрьевича к защите выпускной квалификационной работы в Государственной экзаменационной комиссии (протокол заседания кафедры № \_\_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.)

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Н.А. Спирин)

(подпись)

РЕФЕРАТ

Пояснительная записка изложена на 40 листах и содержит 35 рисунков, 1 таблицу, 3 библиографических наименования и 1 приложение.

АВТОМАТИЗАЦИЯ, РАЗРЫВНАЯ МАШИНА, ИСПЫТАНИЯ ЗАГОТОВОК, ЭЛЕКТРИЧЕСКАЯ СХЕМА, ФУНКЦИОНАЛЬНАЯ СХЕМА, ПРОГРАММИРОВАНИЕ, WINDOWS FORMS.

Выпускная квалификационная работа посвящена разработке классического оконного приложения системы управления разрывной испытательной машиной.

Рассмотрены основные этапы разработки программного обеспечения: постановка задачи, изучение объекта информатизации, поиск и формализация требований к информационной системе, проектирование принципиальной электрической схемы и архитектуры системы, проектирование принципиальной электрической схемы и архитектуры системы, реализация программного обеспечения. Представлено описание работы настольного приложения на примере работы испытательной разрывной машины.

Основными функциями программного обеспечения являются: ввод исходных данных, защита от некорректно вводимых данных, отображение результатов расчета в табличном и графическом видах; сохранение в базе данных результатов расчётов, возможность работы с базой данных через приложение (добавление, удаление, обновление), экспорт отчёт по расчёту во внешний формат офисных документов; ведение блока нормативно-справочной информации. В ходе разработки программного обеспечения были использована ресурсы удаленного репозитория хранения и совместной разработки программного кода IT-проектов GitHub.

Разработанное программное обеспечение предназначено для сотрудников предприятия, занимающегося испытанием материалов и деталей на испытательной разрывной машине.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc136385324)

[1 ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ ВКР 6](#_Toc136385325)

[1.1 Изучение объекта информатизации 6](#_Toc136385326)

[1.2 Принцип работы разрывной машины 8](#_Toc136385327)

[1.3 Описание расчетной части 9](#_Toc136385328)

[1.4 Технологии программной реализации современных информационно-моделирующих систем 10](#_Toc136385329)

[1.5 Постановка цели и задач ВКР 12](#_Toc136385330)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ 13](#_Toc136385331)

[2.1 Организация процесса разработки программного обеспечения 13](#_Toc136385332)

[2.2 Требования к функциональным характеристикам информационной системы 14](#_Toc136385333)

[2.3 Разработка базы данных 16](#_Toc136385334)

[2.4 Реализация программного обеспечения 17](#_Toc136385335)

[2.5 Формирование отчёта 29](#_Toc136385336)

[3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 31](#_Toc136385337)

[3.1 Требования к установке и функционированию системы 31](#_Toc136385338)

[3.2 Технология работы пользователя с программой 31](#_Toc136385339)

[ЗАКЛЮЧЕНИЕ 36](#_Toc136385340)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 37](#_Toc136385341)

[Приложение А Фрагменты листинга программного кода 38](#_Toc136385342)

# ВВЕДЕНИЕ

В современном мире информационные технологии играют неизмеримо важную роль во всех областях жизни. Вероятно, никто не сможет представить свой день без использования как минимум одного устройства, использующего современные технологии. Технические устройства, которые мы используем каждый день - компьютеры, смартфоны, планшеты, домашние устройства и т.д., все они оборудованы сетевыми соединениями, позволяющими людям свободно обмениваться информацией без ограничений. Это привело к тому, что информационные технологии стали неотъемлемой составляющей современной жизни.

Однако, стремительный темп развития и постоянное обновление технологий, а также изменение потребностей общества, требуют от человечества постоянного развития и совершенствования технологических процессов. В связи с этим, информационные технологии становятся все более важными, особенно в сфере промышленного производства. Использование новейших достижений науки в производственных технологиях является необходимым условием конкурентоспособности предприятия. Сегодня практически невозможно представить современную производственную компанию, которая бы не использовала современные инструменты автоматизации, информационных средств и т.д..

Мощное оборудование, автоматизированные производственные линии и системы сбора данных, контрольно-измерительное и лабораторное оборудование являются неотъемлемой частью любого производства в современном мире. Такие технические средства позволяют автоматически обрабатывать и интерпретировать информацию, исходящую из "внешнего мира", что в свою очередь позволяет повысить производительность труда и качество выпускаемой продукции, что, в свою очередь, благоприятно сказывается на производителях и потребителях.

В связи с этим, необходимо обеспечить достаточное количество информационно-модулирующих систем, которые бы своевременно подстраивались под изменяющиеся потребности рынка и позволяли руководству предприятия оптимизировать производственные процессы. Создание и внедрение таких систем является не только актуальной, но и важной задачей, поскольку в ситуации постоянного изменения технологических процессов и потребностей рынка, безопасность и конкурентоспособность предприятия зависят от эффективного использования новых информационных технологий. Управление производством может быть существенно улучшено благодаря развитию информационных технологий.

Таким образом, информационные технологии имеют несомненный вклад в развитие современной жизни. Они являются наиболее прогрессивными инструментами, которые помогают обеспечивать устойчивое развитие науки, промышленности и экономики. Важно продолжать работать над развитием новых технологий, чтобы следить за меняющимися потребностями общества и достижениями науки, и успешно адаптировать их в производственных процессах и других областях жизни. Этим и обусловлена актуальность работы и определена необходимость разработки соответствующей информационно-модулирующей системы.

# 1 ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ ВКР

## 1.1 Изучение объекта информатизации

Разрывная машина является неотъемлемой частью производства качественных материалов и изделий. Материалы, которые мы используем в повседневной жизни, чаще всего подвергаются различным видам нагрузок, таким как давление, растяжение, изгиб и другим. Поэтому перед выпуском любого изделия на рынок необходимо тщательно проверить устойчивость и прочность материалов, из которых оно состоит.

Разрывная машина применяется для проведения тестов на устойчивость материала к нагрузкам в разных направлениях. Она работает путем непрерывного наращивания силы натяжения до тех пор, пока материал не начнет ломаться. Тестирование проводится до тех пор, пока не будут получены данные о максимальной нагрузке, при которой материал выдерживает непрерывную натяжку, высокое давление или другие разновидности прочности материала. Результаты тестирования позволяют производителям изделий определить, какой уровень нагрузки можно ожидать от конкретного материала и применить его для создания конечного продукта.

Использование разрывных машин является одним из наиболее эффективных и надежных методов для тестирования качества материалов и изделий. Это позволяет производителям материалов и изделий избежать возможных неприятностей, связанных со слабостью материала, а также гарантировать более высокую надежность и долговечность конечного продукта. Для разных материалов разрывная машина может создать разные нагрузки и направления, что делает этот метод более гибким и удобным для использования в различных отраслях производства. На рисунке 1.1 изображен общий вид современной испытательной машины, которая может использоваться для проведения различных видов испытаний.

Изображение выглядит как компьютер, мебель, письменный стол, стол

Автоматически созданное описание

Рисунок 1.1 – Пример разрывной машины

Основные части машины:

* электродвигатель постоянного тока Electrocraft серии S644-3A/T через редуктор поднимает раму установки, преобразует электрическую энергию постоянного тока в механическую энергию;
* энкодер Pepperl-Fuchs RVI58N считает обороты двигателя, преобразует угловые положения или линейные перемещения в аналоговый либо цифровой сигнал;
* тензодатчик extensometer tinius olsen fbb-2,5 kn реагирует на изменение величины физического воздействия (усилия) и переводит его в электрический сигнал;
* электрический регулятор мощности / напряжения питания ACMC 60–1 создает и поддерживает фиксированное выходное напряжение независимо от изменений входного напряжения или условий нагрузки;
* контроллер Arduino Uno R3 является посредником между пользователем, работающим в программе и разрывной машиной, выполняющей испытания;
* преобразователь сигнала тензодатчика КСК1 преобразует сигнал с тензодатчика в унифицированный 0…5 (10) В.

## 1.2 Принцип работы разрывной машины

Перед проведением испытаний на разрывной машине оператор должен ввести исходные данные о размерах заготовки, такие как сечение, начальный диаметр, расчетную и рабочую длину материала для того, чтобы обеспечить точность и доскональность проводимых измерений. Затем материал закрепляется с помощью клинового захвата, а специальные измерительные приборы - как, например, датчики перемещения, а также растяжения и сжатия - позволяют измерить различные механические показатели материала.

Для определения временного сопротивления продукта материал деформируется с нарастающей силой до момента разрушения. Во время проведения испытаний на разрывной машине, на материал начинают действовать напряжения в виде передачи растягивающей или сжимающей нагрузки, которые увеличиваются постепенно до наступления момента разрушения.

При проведении испытаний вычисляются различные характеристики материала, такие как максимальная нагрузка, предел текучести, временное сопротивление, относительное сужение и удлинение после разрыва. При этом в программе формируется график нагрузки, который поможет в анализе поведения материала и определении точек, в которых происходит сильная деформация или структурное нарушение.

В конце проведения испытаний на разрывной машине создается отчет, который содержит информацию об исходных данных, измерениях и результатах испытаний. Это позволяет проверить продукцию на соответствие государственным стандартам и требованиям. Разрывная машина является важным оборудованием для определения наилучшего выбора материала для конкретной сферы применения и для повышения качества выпускаемой продукции.

На рисунке 1.2. предоставлена электрическая схема, она выполнена в программе для рисования схем «sPlan 7.0.0.9»

Принципиальная электрическая схема — это графическое представление электрической цепи, на которой изображены электрические элементы и их соединения в виде линий, символов и буквенных обозначений. Элементы схемы обычно отображаются в виде идеализированных символов, таких как резисторы, конденсаторы, индуктивности, источники питания и т. д.

Принципиальная электрическая схема позволяет визуально представить, как электрические элементы соединены друг с другом и как электрический ток будет течь вдоль цепи. Она используется для анализа и проектирования электрических систем.

Кроме того, принципиальная электрическая схема позволяет быстро и удобно диагностировать, и находить неисправности в электрических системах. Например, если электрическое устройство не работает, можно провести проверку его принципиальной электрической схемы, чтобы выяснить, где находится неисправность и как ее устранить.

В целом, принципиальная электрическая схема является важным инструментом для любого инженера или техника, который работает с электро-вычислительными системами, электрооборудованием, а также оборудованием для автоматизации процессов.

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 1.2 – Принципиальная электрическая схема

## 1.3 Описание расчетной части

Когда пользователь вносит данные и нажимает на кнопку «Рассчитать», в процессе измерений по показаниям датчика перемещений и тензодатчика строится график *P*(Δ*l*), где *P* — прилагаемое усилие, Δ*l* — удлинение. Из массива данных выделяется прямолинейный участок и строится параллельная ему прямая на расстоянии 0,2*lрасч.* (20% расчетной длины). По точке пересечения получившейся прямой с графиком *P*(Δ*l*) определяют *Р0,2* для расчета условного предела текучести (на данном примере график является эмуляцией работы разрывной машины).

Условный предел текучестиопределяется согласно ГОСТ 1497 [3] по формуле:

(1.1)

где σ0,2 — предел текучести условный;

P0,2 — усилие предела текучести условного;

F0,2 — начальная площадь поперечного сечения образца.

Для определения временного сопротивления σв образец подвергают растяжению под действием плавно возрастающего усилия до разрушения. Наибольшее усилие, предшествующее разрушению образца, принимается за усилие Рmах, соответствующее временному сопротивлению.

Временное сопротивление рассчитывается согласно ГОСТ 1497 по формуле:

σв=Pmах/F0 (1.2)

где σв — временное сопротивление;

P0,2 — усилие, предшествующее разрушению образца;

F0 — начальная площадь поперечного сечения образца.

Относительное удлинение образца после разрыва рассчитывается согласно ГОСТ 1497 по формуле:

δ=100·(lk—l0)/l0 (1.3)

где δ — относительное удлинение;

 lk — конечная расчетная длина;

 l0 — начальная расчетная длина.

Относительное сужение после разрыва рассчитывается согласно ГОСТ 1497 по формуле:

ψ=100•(F0—Fk)/Fk (1.4)

где ψ — относительное удлинение;

Fk — начальная площадь поперечного сечения образца;

 F0 — площадь поперечного сечения образца после разрыва.

После того, как все расчеты будут получены пользователь на выбор, может сохранить результаты либо распечатать отчет об испытаниях.

## 1.4 Технологии программной реализации современных информационно-моделирующих систем

Разработка информационной системы, предназначенной для проведения испытаний на разрывной машине, выполнена в среде разработки Visual Studio 2022, на языке программирования C# и на платформе .NET Framework 4.7.2, что позволяет использовать передовые технологии и инструменты, которые обеспечивают высокую скорость разработки и улучшение производительности приложения.

Архитектура приложения Windows Forms обеспечивает удобство и простоту интерфейса пользователя, что позволяет удобно взаимодействовать с программой и получать необходимые результаты. Также, событийно-ориентированная архитектура приложения позволяет эффективно контролировать и обрабатывать действия пользователя, что обеспечивает стабильность работы приложения.

Поддержка разработки приложений на платформе .NET Framework 4.7.2 и использование инструментов и библиотек от Microsoft обеспечивает высокую совместимость и стабильность работы приложения на различных операционных системах. В результате пользователи могут использовать разработанную информационную систему на разных устройствах, не опасаясь проблем с совместимостью и работой приложения.

Разработанная архитектура представлена ниже на рисунке 1.3.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 1.3 – Архитектура ПО

Реляционная СУБД Microsoft Access предоставляет широкие возможности для хранения, обработки и управления данными. Она позволяет хранить данные в таблицах, формировать запросы, создавать отчёты и связывать данные между различными таблицами.

Создание базы данных в среде разработки Microsoft Access 2023 на языке Transact-SQL обеспечивает удобный и быстрый доступ к данным, простоту в использовании и возможность быстрого анализа и обработки данных. Язык Transact-SQL является расширением языка SQL и позволяет использовать более мощные и универсальные средства для работы с данными.

Использование реляционной СУБД Microsoft Access для хранения данных, выполнения запросов и формирования отчёта позволяет создать надёжную и удобную систему хранения и обработки информации. Она обеспечивает высокую производительность и совместимость с другими приложениями Microsoft, что позволяет использовать её в различных задачах и проектах.

## 1.5 Постановка цели и задач ВКР

Целью выпускной квалификационной работы является, разработка программно-аппаратного комплекса, который должен решить задачи повышения эффективности работы сотрудников предприятия и оптимизации рабочих процессов в области испытаний материалов и деталей на испытательной разрывной машине.

Основными задачами разработанной системы являются автоматизация процессов обработки и анализа результатов испытаний, предоставление возможности формирования отчётов и их экспорта в различные форматы, обеспечение удобного и быстрого доступа к результатам испытаний и их архивации.

Данный программно-аппаратный комплекс должен предоставлять сотрудникам предприятия лёгкий и простой интерфейс с возможностью настройки параметров испытаний, масштабирования данных и проведения анализа результатов испытаний.

Результатом успешной разработки системы будет увеличение производительности сотрудников предприятия и улучшение качества и точности испытаний материалов и деталей на испытательной разрывной машине.

Для выполнения работы необходимо реализовать следующие задачи:

* изучение объекта информатизации;
* поиск и формализация требований к информационной системе;
* проектирование принципиальной электрической схемы и архитектуры системы;
* реализация базы данных, предназначенной для хранения результатов расчета данных;
* разработка приложения Windows Forms системы разрывной испытательной машины.

# ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Организация процесса разработки программного обеспечения

В процессе разработки ПО использовался сервис GitHub — является крупнейшей платформой для хостинга и совместной разработки IT-проектов. Её основой является система контроля версий Git, которая позволяет разработчикам сохранять разные версии проекта и контролировать его изменения.

В процессе разработки ПО использование системы контроля версий Git оказалось очень полезным. Благодаря возможности хранения нескольких версий проекта разработчик мог экспериментировать с кодом без риска потерять рабочую версию. Кроме того, система Git обеспечила возможность откатиться к предыдущей версии в случае неработоспособности проекта и быстро обнаружить ошибки.

Одним из ключевых преимуществ использования GitHub является удобство совместной работы над проектом. Разработчики могут совместно работать над единым проектом. Каждый разработчик может работать со своей локальной копией проекта, а изменения сливать в единую версию проекта благодаря Git.

В итоге использование GitHub в процессе разработки программного обеспечения позволило обеспечить одновременную работу разработчиков над проектом с минимизацией возможных ошибок и максимальной производительностью.

Перед началом работы над реализацией программного обеспечения был создан удаленный репозиторий на платформе GitHub под названием "Diplom". Репозиторий содержал несколько папок проекта, используемых для хранения различных компонентов программы:

* docs – предназначена для хранения документации;
* app – предназначена для программы, написанной в Microsoft Visual Studio 2022;
* packages – предназначена для хранения установленных расширений Nuget;
* database – предназначена для хранения резервной копии базы данных.

В данном репозитории на GitHub фиксировались все поэтапные изменения версии разрабатываемой программы через приложение Git Extensions. При необходимости, например, после добавления новой функциональности или исправления ошибок, разработчик «коммитил» свой код в свою ветку и затем делал «pull request», чтобы его изменения были приняты и добавлены в основную ветку проекта.

Использование Git Extensions облегчило процесс отслеживания изменений и контроля версий в проекте. Кроме того, Git Extensions предоставляет различные инструменты для решения конфликтов и слияния изменений, что позволяет быстро и эффективно управлять проектом и уменьшить вероятность возникновения ошибок.

Этапы и версии разрабатываемой программы представлены на рисунке 2.1.

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

Автоматически созданное описание

Рисунок 2.1 – Содержимое репозитория на GitHub

## Требования к функциональным характеристикам информационной системы

Требования к функциональным характеристикам информационной системы определяют те возможности и функции, которые должны быть доступны в системе для обеспечения эффективного и эффективного выполнения задач пользователей.

Эти требования обычно устанавливаются в ходе анализа и планирования системы, и они должны быть четко определены с целью создания системы, которая удовлетворит потребности пользователей.

Критерии функциональных характеристик включают в себя следующее:

* Функциональные возможности системы. В системе управления производством функциональными возможностями могут быть возможность создания и управления производственными задачами, контроля цикла производства от начала до конца, автоматического уведомления о возможных проблемах и сбоях в процессе выполнения задач и т.д.
* Производительность системы. В случае системы управления производством, производительность системы может быть измерена скоростью выполнения производственных задач, быстротой реакции системы на изменения в производственном процессе и снижением времени, затраченного на управление производственными задачами.
* Надежность и устойчивость системы. Это определяет надежность работы системы и ее способность к сохранению и защите производственных данных, а также устойчивость системы к возможным аварийным ситуациям. Например, система управления производством должна обеспечивать безопасность обработки данных, предлагать резервное хранение данных для защиты от потерь и отказов системы.
* Интерфейс системы. Интерфейс системы должен быть интуитивно понятным, удобным в использовании и удовлетворять конкретные потребности пользователей. Например, система управления производством должна иметь пользовательский интерфейс, который обеспечивает быстрое и простое выполнение производственных задач. Интерфейс должен быть настроен для полярного использования различимости наглядных факторов в связи с режимами (безопасности, выборки, отметки и т.д.).

В целом, требования к функциональным характеристикам системы направлены на обеспечение эффективного выполнения задач, удовлетворение потребностей пользователей и создание интуитивно понятного и удобного для использования интерфейса системы.

Разрабатываемое программное обеспечение должно содержать следующие функции и возможности:

* наличие справочного раздела, содержащего, руководство пользования программой, а также контакты разработчика;
* ввод исходных данных;
* защита от некорректно вводимых данных в текстовые поля, исходные данные регулируются ГОСТом 1497;
* вывод расчётов на интерфейсе приложения в численном и графическом видах;
* использование пользователем наборов исходных данных;
* возможность сохранение результатов расчета и исходных данных в базу данных;
* редактирование данных таблицы базы данных через интерфейс приложения (добавление, удаление, обновление);
* возможность формирования отчета в формате текстового документа с результатами расчета согласно ГОСТу 1497.

## Разработка базы данных

В данном случае в базе данных только одна таблица, содержащая в себе данные, необходимые для формирования отчёта, а именно исходные данные и результаты расчёта (начальный диаметр, диаметр после разрыва, начальная расчетная длина, конечная расчётная длина, относительное удлинение, временное сопротивление).

В процессе реализации базы данных учтены общие требования к ее свойствам: целостность базы данных – требование полноты и непротиворечивости данных; многократное использование данных; простота обновления данных; обеспечение безопасности и надежности данных; обеспечение целостности и согласованности данных.

Также были учтены специфические требования к свойствам базы данных для формирования отчёта, такие как: точность данных, сохранность исходных данных, возможность переиспользования результатов расчёта, а также удобный доступ к данным для формирования отчёта. Были разработаны схемы и интерфейсы для добавления, изменения и удаления данных из базы данных, а также процедуры и правила, обеспечивающие целостность, безопасность и надежность базы данных. Таким образом, разработанная база данных соответствует требованиям заказчика и обеспечивает удобный и надежный механизм для формирования отчётов на основе данных о различных параметрах испытаний. Таблица БД предоставлена ниже.

Таблица 2.1 – Характеристика сущностей базы данных

|  |  |  |
| --- | --- | --- |
| Наименование столбца | Тип данных | Примечание |
| ID | Счётчик | Первичный ключ |
| Номер\_испытания | Короткий текст |  |
| Дата\_испытания | Короткий текст |  |
| Начальный\_диаметр\_сечения,мм | Числовой |  |
| Диаметр\_сечения\_после\_разрыва,мм | Числовой |  |
| Относительное\_удлиннение,% | Числовой |  |
| Относительное\_сужение,% | Числовой |  |
| Предел\_текучести\_(условный),кгс/мм² | Числовой |  |
| Временное\_сопротивление,кгс/мм² | Числовой |  |

## Реализация программного обеспечения

Программное обеспечение для данного проекта было разработано с использованием совершенных технологий в среде Microsoft Visual Studio 2022. Это мощное инструментальное средство обеспечивает разработчику единый цикл разработки, начиная от написания и редактирования кода, до его отладки и компиляции в готовое приложение. Благодаря интеграции нескольких инструментов в одном месте Visual Studio упрощает и ускоряет процесс разработки программного обеспечения.

К языку программирования C# также предъявляются высокие требования потребителей, так как он позволяет создавать надежные и безопасные приложения. Сборка мусора и обработка исключений обеспечивают автоматическую очистку памяти и увеличивают стабильность и производительность приложения. Также C# позволяет разработчикам динамически создавать объекты, что является важным преимуществом при написании кода.

Составленное в процессе разработки программное обеспечение отвечает всем высоким требованиям потребителей и обеспечивает эффективное и безопасное использование приложения для достижения всех желаемых результатов.

Разработка программы началась с создания проекта, для этого было создано классическое оконное приложение на платформе .NET Framework 4.7.2.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.2 – Создание проекта

Кроме того, в Microsoft Visual Studio 2022 имеется возможность изменять свойства добавленных элементов, как например, цвет, размер, текст и местоположение. Редактирование всех свойств элементов происходит в режиме редактирования формы.

В «Панели элементов» можно выбрать не только различные элементы управления, такие как кнопки, текстовые поля, списки, но и графические элементы, например изображения, формы, линии и т.д. Это позволяет создавать привлекательный и функциональный пользовательский интерфейс.

Также в Microsoft Visual Studio 2022 существует возможность добавления сторонних элементов, таких как библиотеки классов, компоненты и другие расширения. Это усиливает гибкость и разнообразие возможностей, доступных приложению.

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

Автоматически созданное описание

Рисунок 2.3 – Панель элементов

Далее необходимо открыть форму в конструкторе и добавить нужные нам элементы:

* label — нужен для отображения текста, с помощью него добавили заголовки;
* textbox — элемент, который предназначен для ввода и редактирования символов, в нём записываются данные для проведения испытаний, а также выводятся результаты расчётов;
* radioButton — элемент, который позволяет выбрать какой-либо вариант из группы вариантов, он используется для выбора сечения образца: круглое или прямоугольное, от этого зависит какую формулу расчёта нужно использовать, какие исходные данные нужно заполнять, а также с помощью данного элемента можно запустить построение графика или остановить;
* menuStrip — контейнер, добавляющий список меню в верхней части программы, необходим для вызова справки о пользовании программой, а также снабжен кнопкой «выход»;
* chart — компонент, с помощью которого можно построить различного вида графики, добавлен, чтобы выводить результаты расчёта в графическом виде, наглядном для пользователя, по оси x строится удлинение, а по оси y прилагаемое усилие;
* groupBox — нужен для группировки нескольких элементов формы на функциональные области;
* numericUpDown — элемент, который содержит в себе одно числовое значение, которое можно увеличить или уменьшить на определенное значение, путём нажатия кнопок в нём, используется для эмуляции работы разрывной машины, увеличивая или уменьшая его значение на графике меняется прилагаемое усилие;
* dataGridView — элемент в виде таблицы, отображающий необходимый набор данных для просмотра и редактирования, используется, чтобы построить таблицу исходных данных и результатов расчёта, по которым в дальнейшем можно построить отчёт;
* button — элемент, нажав на который происходит событие, которое можно запрограммировать по желанию разработчика, кнопка «Рассчитать» считывает исходные данный, проверяет их на соответствие размеров, считает результат; кнопка «Значение по умолчанию» устанавливает исходные данные для расчёта; кнопка «Сохранить результаты» переносит текстовые поля в таблицу, из которой впоследствии строится отчёт; кнопка «Добавить» нужна, чтобы добавить данные из dataGridView в базу данных; кнопка «Загрузить» загружает данные из базы в таблицу dataGridView; кнопка «Обновить» обновляет данные в базе данных, считывая значения полей таблицы dataGridView; кнопка «Удалить» удаляет строку из dataGridView и из базы данных; кнопка «Открыть базу данных» открывает из программы базу данных в Microsoft Access; кнопка «Сформировать отчёт» формирует отчёт, сделанный через программу Fast Report;
* panel — элемент управления, содержащий другие элементы управления, использован для эмуляции лампочки, зеленый цвет, когда работает, красный-нет;
* tabControl — элемент, который позволяет добавить несколько вкладок на форму, с помощью него сделано две вкладки: «Расчёты» и «Отчёт»;
* timer — элемент, с помощью которого можно выполнять событие через заданный интервал времени, использован при построении графика.

Дизайн приложения предоставлен на рисунке ниже.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 2.4 – Интерфейс разработанного программного продукта, вкладка «Расчёты»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 2.5 – Интерфейс разработанного программного продукта, вкладка «Отчёт»

После добавления на форму нужных элементов необходимо их настроить. Первым делом настроен элемент dataGridView, для него сделано 9 столбцов с нужными параметрами, также прописано свойство AutoSizeMode на каждый столбец, чтобы ширина столбца принимала нужное значение автоматически.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 2.6 – Настройка dataGridView

Также при запуске приложения и загрузке формы она будет появляться посреди экрана, используется свойство положения формы.

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Автоматически созданное описание

Рисунок 2.7 – Положение формы на экране

Далее был построен график, сначала проверяется значение положение radioButton, если оно принимает одно положение, то график строится, если другое- то нет, через свойство Series добавлено 2 значения, первое для x – время сейчас, для y – значение элемента numericUpDown, также добавлено условие, которое проверяет сколько секунд, прошло с начала отсчёта по оси x, и если значение достигает 60 секунд, то график обновляется, интервал с которым добавляется точка на график, по которой строится линия равен 6 миллисекундам.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 2.8 – Построение графика через Chart

В программном обеспечении, разработанном на Microsoft Visual Studio 2022 на языке C#, важным элементом является обработка исключений. Обработка исключений позволяет программе перехватывать и обрабатывать возможные ошибки, которые могут возникнуть в процессе работы программы.

В контексте данного проекта перед тем, как производить расчеты на основе введенных пользователем данных, необходимо выполнить проверку того, что все необходимые данные были введены корректно. Если какой-либо из входных параметров не задан или задан некорректно, программа должна выдать соответствующую ошибку. Для этого можно использовать соответствующие конструкции языка C#, такие как условные операторы if-else и switch-case, а также библиотеки классов, предназначенные для работы с исключениями.

Такая проверка является необходимой для обеспечения корректности и стабильности работы программы. Если пользователям будет позволено вводить некорректные данные, то это может привести к непредсказуемым результатам и негативно сказаться на работе программы в целом. Поэтому проверка входных данных является важным элементом проектирования и разработки программного обеспечения.

Чтобы проверить какие данные ввел пользователь перед расчётом программа проверяет в каком диапазоне лежат исходные данные, если они не удостоверяют условиям, то появляется табличка об ошибке.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2.9 – Ошибка

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.10 – Проверка вводимых данных

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2.11 – Проверка исходных данных

Далее необходимо считать вводимые пользователем в текстовые поля данные

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.12 – Считывание исходных данных

После того как были получены необходимые данные можно перейти к расчёту, для этого пропишем формулы, также учтём, что в зависимости от сечения образца будут разные формулы. И после расчёта сразу выводим данные в текстбоксы.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2.13 – Формулы расчёта

Для установки значений по умолчанию используется кнопка, нажав на которую исходные данные наполнят текстовые поля

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 2.14 – Исходные данные

Выбор типа заготовки – круглой или прямоугольной – является важным этапом перед вводом исходных данных в программу. Это позволяет пользователю сформировать конкретные параметры расчета, соответствующие выбранному типу заготовки.

При выборе круглой заготовки ширина и толщина заготовки не учитываются, вместо этого необходимо задать диаметр сечения заготовки. Это связано с тем, что форма круглой заготовки представляет собой окружность, а значит для расчета необходимо знать ее диаметр.

Если выбрана прямоугольная заготовка, то для расчетов необходимо задать ее ширину и толщину, а параметр диаметра сечения заготовки не учитывается. Это обусловлено формой прямоугольной заготовки, которая имеет прямоугольную форму, а значит для расчетов необходимо знать ее ширину и толщину.

Выбор типа заготовки является важным моментом при вводе исходных данных, и он влияет на параметры, которые пользователь должен указать для получения корректных результатов расчета. Для удобства пользователей программное обеспечение должно быть спроектировано таким образом, чтобы выбор типа заготовки был ясным, понятным и доступным.

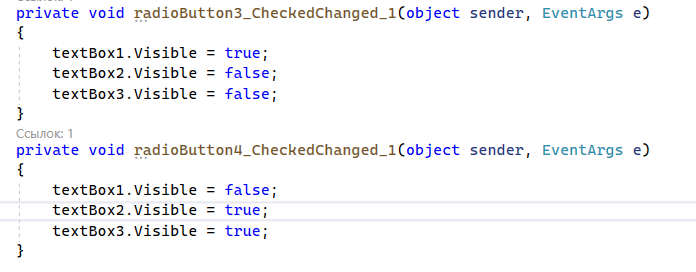


Рисунок 2.15 – Выбор типа сечения

Руководство пользователя – важный компонент программного обеспечения, который позволяет пользователям ознакомиться с функциями программы и получить необходимую информацию о ее работе. В разработанной программе, также добавлено руководство пользователя, которое можно открыть по кнопке "О программе", расположенной в меню сверху.

Руководство пользователя должно содержать информацию обо всех частях программы, включая входные данные, результаты расчетов, возможные ошибки и способы их устранения, а также другую важную информацию о работе программы.

Кроме того, в руководстве пользователя должны быть описаны все функции программы и объяснены основные концепции, лежащие в ее основе. Это поможет пользователям понять, как правильно использовать программу и как получать максимальную пользу от ее использования.

Руководство пользователя может быть представлено в различных форматах, таких как документация в формате pdf или html, видеоуроки или интерактивные инструкции. Важно, чтобы оно было доступно пользователям в удобном и понятном формате, так что они могли легко найти и использовать нужную им информацию.

Изображение выглядит как текст, линия, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2.16 – Руководство пользователя

Далее было сделано, чтобы по нажатию кнопки переносились нужные значения из текстбоксов в dataGridView.

Для этого были написан код, который при нажатии на кнопку считывает значения из текстбоксов и переносит их в соответствующие ячейки dataGridView.

Функция переноса значений из текстбоксов в dataGridView позволяет пользователям легко сохранять результаты расчетов и просматривать их в удобном формате. Она также позволяет легко изменять значения, если это необходимо, и обеспечивает более наглядное представление данных.

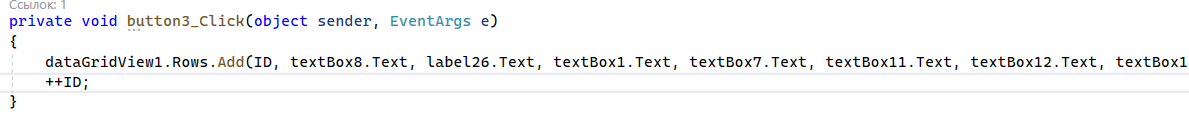


Рисунок 2.17 – Перенос в таблицу

Для установки соединения с БД используется специальный класс connectionString, который позволяет установить соединение с БД, используя строку подключения. После установки соединения происходит выполнение запроса к БД, для этого используется объект класса, куда передается запрос к БД в виде строки с помощью метода.

После выполнения запроса данные из базы данных сохраняются в объекте класса, который предоставляет доступ к строкам, возвращаемым запросом. Затем данные могут быть использованы для отображения или дальнейшей обработки.

После выполнения запроса и получения данных из БД соединение закрывается с помощью метода Close.

Для работы dataGridView и базы данных между собой были написаны 4 запроса:

* Запрос на загрузку данных из БД – устанавливается соединение с базой данных, после чего происходит запрос к БД и после выполнения запроса подключение закрывается;

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.18 – Запрос «SELECT»

* Запрос на добавление данных в БД из столбцов dataGridView предоставлен на рисунке 2.19;

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 2.19 – Запрос «INSERT»

* По аналогии добавлен запрос на обновление данных, с помощью которых происходит обновление данных в базе данных данными из таблицы dataGridView;

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 2.20 – Запрос «UPDATE»

* Последний запрос удаляет данные из базы данных и dataGridView.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.21 – Запрос «DELETE»

## Формирование отчёта

Отчет является неотъемлемой частью любого приложения, на котором работают с данными. Он позволяет наглядно отобразить полученные результаты расчетов, анализа и пр.

Чтобы создать отчет в программе на Microsoft Visual Studio 2022 на языке C#, необходимо скачать и установить конструктор отчетов FastReport. После установки необходимо добавить библиотеку FastReport.dll в проект, чтобы сделать доступным элемент report для создания отчетов.

Для открытия конструктора отчетов необходимо добавить элемент report в проект и открыть его. В конструкторе отчетов можно создавать новые отчеты, добавлять поля, вычисляемые колонки, таблицы и пр.

Для заполнения отчета данными из базы данных необходимо создать соответствующий запрос к БД и передать полученные данные в отчет. Для этого необходимо использовать объект DataSet, который предназначен для хранения данных из БД.

После заполнения отчета данными, его можно сохранить в различных форматах, например, PDF, Excel или HTML.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 2.22 – Интерфейс программы «FastReport»

Чтобы добавить на страницу отчёта данные из базы данных необходимо установить с ней подключение.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 2.23 – Подключенная база данных

Теперь, когда соединение с БД установлено можно настроить отчёт, добавлен заголовок отчёта и таблица с нужными данными. Отчёт добавлен в проект и при нажатии кнопки он отображается.

# ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 3.1 Требования к установке и функционированию системы

Для пользования программой необходим компьютер с операционной системой, поддерживающей .NET Framework 4.7.2. Чтобы запустить программу необходимо запустить файл с расширением .exe.

## Технология работы пользователя с программой

Важной частью технологии работы пользователя с программой является обучение пользователя её использованию и доступность соответствующей документации.

При запуске программы пользователь попадает на главную форму. Далее у пользователя есть выбор, либо ознакомиться с функциями программы и нажать кнопку «О программе», тогда он сможет изучить функции программы, либо начать заполнять исходные данные вручную или по кнопке «Значения по умолчанию».

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

Автоматически созданное описание

Рисунок 3.1 – Ввод исходных данных

После ввода исходных данных пользователь должен запустить график, чтобы пройти эмуляцию работы разрывной машины.

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 3.2 – Построение графика

Далее по кнопке «Рассчитать» по формуле определяются искомые значения. Закончив расчёты, пользователь может ввести номер испытания для протокола, дата формируется автоматически, в таблице ее можно сменить на другую.

Изображение выглядит как текст, Шрифт, число, линия

Автоматически созданное описание

Рисунок 3.3 – Расчёт данных

Далее по кнопке сохраняются результаты.

Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описание

Рисунок 3.4 – Заполнение таблицы

Также можно загрузить данные о предыдущих испытаниях, нажав на кнопку «Загрузить».

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 3.5 – Загруженные из БД данные

Если с данными что-то не так, то их можно отредактировать и нажать кнопку «Обновить», тогда существующие в БД данные обновятся.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описаниеРисунок 3.6 – Успешное обновление данных

По кнопке «Добавить» данные добавляются в базу данных.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 3.7 – Успешное добавление данных

Нажав кнопку «Открыть базу данных» откроется база данных в Microsoft Access. Кнопка «Удалить» удалит строку из таблицы и из БД. После расчётов и сохранения данных в базу данных можно сформировать отчёт.

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

Автоматически созданное описание

Рисунок 3.8 – Сформированный отчёт

Данный отчёт можно сохранить в текстовый документ, например Word.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 3.9 – Отчёт в текстовом редакторе

Таким образом, разработанная информационная система для проведения испытаний на разрывной машине позволяет пользователям получать достоверные и точные результаты испытаний и быстро обрабатывать их, а также хранить данные в базе данных, что способствует их безопасности и удобству доступа. Кроме этого, добавление функций «добавить», «удалить», «обновить» и «загрузить» из программы добавляет удобства в работе с базой данных. Генерация отчета позволяет быстро и эффективно обработать данные и получить выводы и рекомендации по их дальнейшему использованию. Кнопка «Сохранить как» позволяет сохранить отчёт в различных форматах, что способствует удобству работы с данными.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы спроектирована и реализована система, использующаяся для проведения испытаний на разрывной машине. Были реализованы следующие этапы:

* изучение объекта информатизации, изучение особенностей работы и процесса проведения испытаний на разрывной машине, а также определение потребностей и требований к разрабатываемому программному обеспечению;
* поиск и формализация требований к информационной системе, составление списка требований к функциональным и нефункциональным характеристикам системы, а также к ее интерфейсу и взаимодействию с пользователем;
* проектирование принципиальной электрической схемы и архитектуры системы, проектирование основных компонентов системы, а также определение аппаратного обеспечения, необходимого для работы разработанного программного обеспечения;
* реализация базы данных, предназначенной для хранения результатов расчета данных, создание программного модуля для взаимодействия с базой данных;
* формирование отчета о результатах проведенных испытаний и их анализ. В рамках данного этапа происходит сбор и анализ полученных результатов испытаний на разрывной машине, формирование отчёта, содержащего данные о соответствии испытуемых материалов и деталей требованиям, а также выводы и рекомендации по их дальнейшему использованию. Сформированный отчёт анализируется и используется заказчиком в дальнейшей работе с материалами и деталями, а также соответствует государственному стандарту;
* разработка приложения Windows Forms системы разрывной испытательной машины - включает в себя написание программного кода приложения, позволяющего управлять процессом испытаний на разрывной машине, а также обрабатывать, анализировать и хранить данные.

Полученное программное обеспечение соответствует требованиям, предъявляемым к его функциональным характеристикам, и предназначено для использования в сфере испытания материалов и деталей на разрывной машине. Оно позволяет существенно упростить и ускорить процесс проведения испытаний и обеспечивает высокую точность и достоверность результатов.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сексенбаев, Курманбек. Информационные технологии в развитии современного информационного общества / Курманбек Сексенбаев, Б. К. Султанова, М. К. Кисина. — Текст : непосредственный // Молодой ученый. — 2015. — № 24 (104). — С. 191-194. — URL: <https://moluch.ru/archive/104/24209>
2. Испытательная техника: справочник: в 2-х кн. /Г.С. Батуев, А.С. Больших, В.С. Голубков, Д.А. Гречинский, В.Н. Гудцов и др.; под ред. В.В. Клюева. – 528 с.
3. ГОСТ 28840-90. Машины для испытаний материалов на растяжения, сжатие и изгиб. Общие технические требования. – 8 с.
4. ГОСТ 1497-84 (ИСО 6892-84, СТ СЭВ 471-88) МЕТОДЫ ИСПЫТАНИЙ НА РАСТЯЖЕНИЕ. – 26 с. –  URL:  <https://files.stroyinf.ru/Data2/1/4294852/4294852801.pdf>.
5. Кислов А.Н. Сопротивление материалов. Простые виды деформации/
6. <https://elar.urfu.ru/bitstream/10995/105753/1/978-5-7996-3392-9_2021.pdf>

# Приложение А Фрагменты листинга программного кода

public Form1()

{

InitializeComponent();

StartPosition = FormStartPosition.CenterScreen;

label26.Text = DateTime.Now.ToString("dd.MM.yyyy");

dataGridView1.ColumnCount = 9;

dataGridView1.Columns[0].Name = "ID";

dataGridView1.Columns[1].Name = "Номер\_испытания";

dataGridView1.Columns[2].Name = "Дата\_испытания";

dataGridView1.Columns[3].Name = "Начальный\_диаметр\_сечения,мм";

dataGridView1.Columns[4].Name = "Диаметр\_сечения\_после\_разрыва,мм";

dataGridView1.Columns[5].Name = "Относительное\_удлиннение,%";

dataGridView1.Columns[6].Name = "Относительное\_сужение,%";

dataGridView1.Columns[7].Name = "Предел\_текучести\_(условный),кгс/мм²";

dataGridView1.Columns[8].Name = "Временное\_сопротивление,кгс/мм²";

dataGridView1.Columns[0].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[2].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[3].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[4].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[5].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[6].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[7].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

dataGridView1.Columns[8].AutoSizeMode = DataGridViewAutoSizeColumnMode.Fill;

}

private int \_countSecond = 0;

int limitNagruz = 1000;

private void Form1\_Load(object sender, EventArgs e)

{

timer1.Enabled = true;

numericUpDown1.Maximum = 1000;

numericUpDown1.Minimum = 0;

chart1.ChartAreas[0].AxisY.Maximum = 1000;

chart1.ChartAreas[0].AxisY.Minimum = 0;

chart1.ChartAreas[0].AxisX.LabelStyle.Format = "H:mm:ss";

chart1.Series[0].XValueType = ChartValueType.DateTime;

chart1.ChartAreas[0].AxisX.Minimum = DateTime.Now.ToOADate();

chart1.ChartAreas[0].AxisX.Maximum = DateTime.Now.AddMinutes(1).ToOADate();

chart1.ChartAreas[0].AxisX.IntervalType = DateTimeIntervalType.Seconds;

chart1.ChartAreas[0].AxisX.Interval = 6;

}

private void timer1\_Tick(object sender, EventArgs e)

{

if (radioButton1.Checked)

{

panel1.BackColor = Color.Green;

Random random = new Random(DateTime.Now.Millisecond);

double ranNagruzka = random.Next(0, 1000);

label19.Text = ranNagruzka.ToString();

DateTime timeNow = DateTime.Now;

int value = Convert.ToInt32(numericUpDown1.Value);

chart1.Series[0].Points.AddXY(timeNow, value);

chart1.Series[1].Points.AddXY(timeNow, ranNagruzka);

int max = 0;

int secMax = 0;

if (Convert.ToInt16(numericUpDown1.Value) > max)

{

max = Convert.ToInt16(numericUpDown1.Value);

}

secMax = max;

textBox9.Text = secMax.ToString();

\_countSecond++;

if (\_countSecond == 60)

{

\_countSecond = 0;

chart1.ChartAreas[0].AxisX.Minimum = DateTime.Now.ToOADate();

chart1.ChartAreas[0].AxisX.Maximum = DateTime.Now.AddMinutes(1).ToOADate();

chart1.ChartAreas[0].AxisX.IntervalType = DateTimeIntervalType.Seconds;

chart1.ChartAreas[0].AxisX.Interval = 6;

}

}

else

{ panel1.BackColor = Color.Red; }

\_countSecond++;

if (\_countSecond == 60)

{

\_countSecond = 0;

chart1.ChartAreas[0].AxisX.Minimum = DateTime.Now.ToOADate();

chart1.ChartAreas[0].AxisX.Maximum = DateTime.Now.AddMinutes(1).ToOADate();

chart1.ChartAreas[0].AxisX.IntervalType = DateTimeIntervalType.Seconds;

chart1.ChartAreas[0].AxisX.Interval = 6;

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

// Удаляем данные с ячейки

textBox10.Text = "";

textBox11.Text = "";

textBox12.Text = "";

textBox13.Text = "";

textBox14.Text = "";

// Сверяем значения исходных данных

if (textBox1.Text == "" ||

textBox2.Text == "" ||

textBox3.Text == "" ||

textBox4.Text == "" ||

textBox5.Text == "" ||

textBox6.Text == "" ||

textBox7.Text == "")

{

MessageBox.Show("Заполните исходные данные", "Внимание!");

return;

}