Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и технологий

Кафедра «Теплофизика и информатика в металлургии»

Оценка работы\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель от УрФУ Е.А. Девятых

Тема задания на практику

«Проектирование и реализация функций информационной системы разрывной испытательной машины»

ОТЧЕТ

Вид практики Производственная практика

Тип практики Преддипломная практика

Студент: Бурдаков Владислав Юрьевич

Код, направление подготовки: 09.03.02 – «Информационные системы и технологии»

Учебная группа: НМТ-493907

Екатеринбург 2023

Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н. Ельцина»

|  |  |
| --- | --- |
|  | СОГЛАСОВАНО:  УрФУ  «10» апреля 2023 г. |
|  | Зав. кафедрой «Теплофизика и информатика в металлургии» ИНМТ  \_\_\_\_\_\_\_\_\_\_ Н.А. Спирин  Подпись расшифровка подписи |

Институт ИНМТ. Группа НМТ-493907. Кафедра «Теплофизика и информатика в металлургии»

Код, наименование направления 09.03.02 – Информационные системы и технологии (СУОС)

Наименование образовательной программы 09.03.02/33.11 – Информационные системы и технологии в металлургии

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

на преддипломную практику студента

Бурдаков Владислав Юрьевич

(фамилия, имя, отчество)

1. Тема задания на практику «Проектирование и реализация функций информационной системы разрывной испытательной машины»

2. Срок практики с 10.04.2023 г. по 21.05.2023 г. Срок сдачи студентом отчета c 22.05.2023 г. по 28.05.2023 г.

3. Место прохождения практики Кафедра «Теплофизика и информатика в металлургии».

4. Вид практики: Производственная практика.

5. Тип практики: Преддипломная практика

6. Содержание отчета: (описание принципов работы, объектов испытаний разрывной испытательной машины; формализация требований к разрабатываемой информационной системе; описание аппаратной конфигурации, описание функций программы управления, описание принципов построения информационной системы на базе открытых систем. Объем отчета - не менее 20 страниц).

**Рабочий график (план) проведения практики**

|  |  |  |  |
| --- | --- | --- | --- |
| **Этапы практики** | **Наименование работ студента** | **Срок** | **Примечание** |
| организационный | Ознакомление с рабочей программой практики; изучение методических рекомендаций по практике; согласование индивидуального задания с РП от УрФУ; усвоения правил техники безопасности и охраны труда | 10.04.2023 г. –  17.04.2023 г. |  |
| основной | Изучение объекта информатизации, поиск и формализация требований к информационной системе | 18.04.2023 г. –  07.05.2023 г. |  |
| заключительный | Подведение итогов и составление отчета: систематизация, анализ, обработка собранного в ходе практики материала, предоставление отчета | 08.05.2023 г. –  21.05.2023 г. |  |

Содержание практики и планируемые результаты практики согласованы с руководителем:

Руководитель от УрФУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.А. Девятых

Подпись расшифровка подписи

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

Подпись расшифровка подписи

РЕФЕРАТ

Пояснительная записка изложена на 33 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 ПО 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 5](#_Toc135698800)

[1 ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ ВКР 6](#_Toc135698801)

[1.1 Изучение объекта информатизации 6](#_Toc135698802)

[1.2 Принцип работы разрывной машины 7](#_Toc135698803)

[1.3 Описание расчетной части 8](#_Toc135698804)

[1.4 Технологии программной реализации современных информационно-моделирующих систем 9](#_Toc135698805)

[1.5 Постановка цели и задач ВКР 10](#_Toc135698806)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ 11](#_Toc135698807)

[2.1 Организация процесса разработки программного обеспечения 11](#_Toc135698808)

[2.2 Требования к функциональным характеристикам информационной системы 12](#_Toc135698809)

[2.3 Разработка базы данных 12](#_Toc135698810)

[2.4 Реализация программного обеспечения 13](#_Toc135698811)

[2.5 Формирование отчёта 24](#_Toc135698812)

[3 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 25](#_Toc135698813)

[3.1 Требования к установке и функционированию системы 25](#_Toc135698814)

[3.2 Технология работы пользователя с системой 25](#_Toc135698815)

[ЗАКЛЮЧЕНИЕ 29](#_Toc135698816)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 30](#_Toc135698817)

[Приложение А Фрагменты листинга программного кода 31](#_Toc135698818)

# ВВЕДЕНИЕ

В современном мире просто невозможно представить жизнь без информационных технологий несмотря на то, что в самом недалеком прошлом человек и понятия не имел о них. В нашу жизнь они вошли прочно, применяются информационные технологии во всех сферах жизни человечества, выполняя особо значимую двойственную роль. Информационные технологии представляют весь накопленный опыт человечества в форматизированном виде, пригодном для прикладного использования. И в нем сконцентрированы научные знания и материалистический опыт для осуществления общественных процессов, при этом экономятся затраты труда, времени, энергии, вещественных средств.

Стремительные темпы компьютеризации всех сторон человеческой деятельности привели к тому, что сегодня компьютеры, и, прежде всего, персональные ЭВМ, стали непременным атрибутом самых различных технических комплексов. Это касается и современных систем управления и сбора данных, контрольно-измерительного и лабораторного оборудования, т. е. любых комплексов, основной задачей которых является обработка и интерпретация информации, поступающей из “внешнего мира”.

Не остались в стороне и предприятия: сложно представить сегодняшний завод или компанию, которые не используют в своей работе современные технологии.

# 1 ПОСТАНОВКА ЦЕЛИ И ОСНОВНЫХ ЗАДАЧ ВКР

## 1.1 Изучение объекта информатизации

Все материалы имеют свойство рваться при воздействии на них разнонаправленной нагрузки. При запуске массового производства, определении брака и других задачах производства важно определить степень устойчивости образца к нагрузкам. Для тестирования в лабораториях используются разрывные машины. Они создают сильное натяжение и подходят для тестирования разных типов материалов и деталей [1].

Разрывная машина — это аппарат для статических испытаний и определения физических свойств материалов на осевое растяжение, сжатие и изгиб, и другие испытания в рамках технических возможностей машины.

Изображение выглядит как компьютер, мебель, письменный стол, стол

Автоматически созданное описание

Рисунок 1.1.1 – 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 тем, как провести испытания, оператор вводит исходные данные о размерах заготовки: сечение, диаметр начальный, расчетная и рабочая длина. После этого материал закрепляется клиновым захватом. Для определения механических показателей заготовки используется измеритель силы и измеритель перемещений, также измерители растяжения и сжатия. Если вам необходимо узнать временное сопротивление продукта, его деформируют с нарастающей силой до тех пор, пока продукт не разрушится [2]. При этом рассчитываются значения максимальной нагрузки, предела текучести, временного сопротивления, относительного сужения и удлинения после разрыва. Во время процесса деформации программой строится график нагрузок. В конце проверки создается требуемый отчет. Для любого завода или предприятия обязательно наличие разрывной машины, специальных дополнительных устройств и программного обеспечения. Как раз с помощью этого уникального устройства и осуществляется проверка продукции на соответствие государственным стандартам. На рисунке 1.2.1 предоставлена электрическая схема, она выполнена в программе для рисования схем «sPlan 7.0.0.9»

Изображение выглядит как диаграмма

Автоматически созданное описание

Рисунок 1.2.1 – 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).

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

σ0,2=P0,2/F0,

где σ0,2 — предел текучести условный, P0,2 — усилие предела текучести условного, F0 — 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,

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,

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,

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 испытания на разрывной машине, будет вестись в среде разработке Visual Studio 2022, язык программирования C#, на платформе .NET Framework 4.7.2. Архитектура приложения Windows Forms представляет собой событийно-ориентированное (выполнение программы определяется событиями — действиями пользователя: клавиатура, мышь, сенсорный экран, сообщениями других программ и потоков, событиями операционной системы) приложение, поддерживаемое Microsoft. Разработанная архитектура представлена ниже на рисунке 1.4.1.

Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание

Рисунок 1.4.1 – Архитектура ПО

Для хранения данных расчетов, выполнения запросов и формирования отчёта будет использоваться реляционная СУБД Microsoft Access. Создание базы данных будет осуществляться в среде разработки Microsoft Access 2023 на языке Transact-SQL.

## 1.5 Постановка цели и задач ВКР

Целью ВКР является разработка программно-аппаратного комплекса, предназначенного для сотрудников предприятия, занимающегося испытанием материалов и деталей на испытательной разрывной машине. Разработанное программное обеспечение предназначено для сотрудников предприятия, занимающегося испытанием материалов и деталей на испытательной разрывной машине. Таким образом, данный программно-аппаратный комплекс должен существенно облегчить работу сотрудникам предприятия и в целом оптимизировать рабочие процессы.

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.

# 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 2.1 Организация процесса разработки программного обеспечения

В процессе разработки ПО использовался GitHub— крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git, позволяющей разработчикам в процессе работы над проектом сохранять различные его версии. Ключевыми практическими преимуществами использования системы контроля версий Git являются возможность откатиться до предыдущей версии в случае наступления неработоспособности IT-проекта вместо долгих поисков ошибок и удобство совместной работы разработчиков над единым проектом.

Перед началом работы над реализацией программного обеспечения был создан удаленный репозиторий «Diplom», содержащий несколько папок проекта:

* 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 фиксировались все поэтапные изменения версии разрабатываемой программы через приложение Git Extensions.

Этапы и версии предоставлены на рисунке 2.1.1.

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

Автоматически созданное описание

Рисунок 2.1.1 – Содержимое репозитория на GitHub

## 2.2 Требования к функциональным характеристикам информационной 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.

## 2.3 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.

Таблица 2.3.1 – 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 |  |

## 2.4 Реализация программного обеспечения

Программное обеспечение разрабатывалось в среде разработки Microsoft Visual Studio 2022 на языке C#.

Microsoft Visual Studio — это мощное средство разработчика, которое можно использовать для единого завершения всего цикла разработки. Это комплексная интегрированная среда разработки (IDE), которую можно использовать для написания, редактирования, отладки и сборки кода, а затем для развертывания приложения. Помимо редактирования и отладки кода, Visual Studio включает компиляторы, средства завершения кода, систему управления версиями, расширения и многие другие функции для улучшения каждого этапа процесса разработки программного обеспечения.

C# — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. Его функции, которые позволяют создавать надежные и устойчивые приложения:

* Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами;
* Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты;
* Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них;
* Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем;
* C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке.

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 2.4.1 – Создание проекта

Добавление элементов на форму происходит через «Панель элементов», для этого необходимо перенести интересующий элемент на дизайн формы.

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

Автоматически созданное описание

Рисунок 2.4.2 – 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 2.4.3 – Интерфейс разработанного программного продукта, вкладка «Расчёты»

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

Автоматически созданное описание

Рисунок 2.4.4 – 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 2.4.5 – 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 2.4.6 – 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 2.4.7 – Построение графика через Chart

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 2.4.8 – Ошибка

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.4.9 – Проверка вводимых данных

Также перед расчётом необходимо проверить ввел ли пользователь исходные данные или нет, если нет, то он получит соответствующую ошибку.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 2.4.10 – Проверка исходных данных

Далее необходимо считать вводимые пользователем в текстовые поля данные

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.4.11 – 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 2.4.12 – 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 2.4.13 – Исходные данные

Перед вводом исходных данных пользователю нужно выбрать тип заготовки, если круглая, то ширина и толщина заготовки не будут учитываться, если прямоугольная, то не учитывается диаметр сечения.

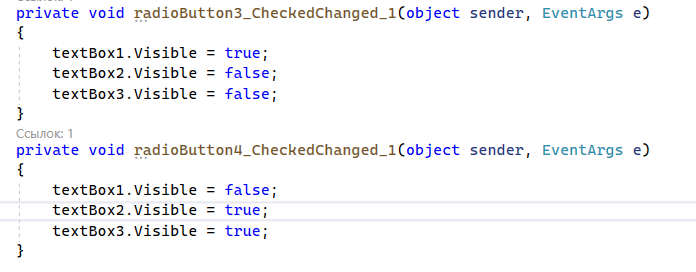


Рисунок 2.4.14 – Выбор типа сечения

Чтобы пользователь мог ознакомиться с функциями программы добавлено руководство пользователя, оно открывается по кнопке «О программе», расположенной в меню сверху.

Изображение выглядит как текст, линия, Шрифт, снимок экрана

Автоматически созданное описание

Рисунок 2.4.15 – Руководство пользователя

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

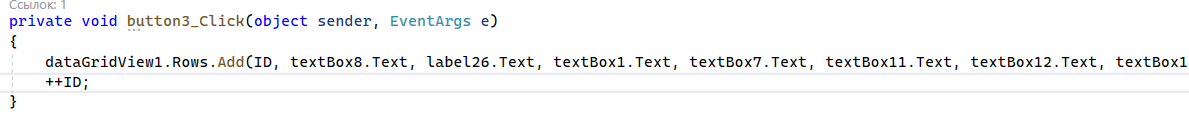


Рисунок 2.4.16 – Перенос в таблицу

Далее 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 2.4.17 – Запрос «SELECT»

* Запрос на добавление данных в БД из столбцов dataGridView предоставлен на рисунке 2.5.18.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 2.4.18 – 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 2.4.19 – Запрос «UPDATE»

* Последний запрос удаляет данные из базы данных и dataGridView.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 2.4.20 – Запрос «DELETE»

## 2.5 Формирование отчёта

Чтобы сделать отчёт первым делом необходимо скачать программу FastReport (конструктор отчётов), после чего необходимо добавить библиотеку FastReport.dll, тогда станет доступен элемент report. Через добавленный элемент открываем конструктор отчётов.

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

Автоматически созданное описание

Рисунок 2.5.1 – 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 2.5.2 – 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.

# 3 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.

## 3.2 Технология работы пользователя с системой

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 3.2.1 – 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 3.2.2 – 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 3.2.3 – Расчёт данных

Далее по кнопке сохраняются результаты.

Изображение выглядит как текст, линия, Шрифт, число

Автоматически созданное описание

Рисунок 3.2.4 – 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 3.2.5 – 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 3.2.6 – 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 3.2.7 – 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 3.2.8 – Сформированный отчёт

Данный отчёт можно сохранить в текстовый документ, например Word.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 3.2.9 – Отчёт в текстовом редакторе

# ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была спроектирована и разработана информационная система, для проведения испытаний на разрывной машине. Работа состояла из следующих этапов:

* 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.

Разработанное программное обеспечение предназначено для сотрудников предприятия, занимающегося испытанием материалов и деталей на испытательной разрывной машине, и соответствует требованиям, предъявляемым к ее функциональным характеристикам. Таким образом, данный программно-аппаратный комплекс должен существенно облегчить работу сотрудникам предприятия и в целом оптимизировать рабочие процессы.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. 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.
2. ГОСТ 28840-90. Машины для испытаний материалов на растяжения, сжатие и изгиб. Общие технические требования. – 8 с.
3. 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>.

# 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;

}