Департамент образования и науки Курганской области

Государственное бюджетное профессиональное образовательное учреждение

«Курганский технологический колледж

имени Героя Советского Союза Н.Я. Анфиногенова»

(ГБПОУ «КТК»)

Защищен с оценкой:

2022 г.

ОТЧЕТ

по *производственной (по профилю специальности)*  практике

вид практики

*ПМ.01Разработка модулей программного обеспечения для компьютерных систем*

наименование профессионального модуля

в

наименование организации

Выполнил обучающийся группа

Ф.И.О. обучающегося № группы

Специальность *09.02.07 Информационные системы и программирование*

шифр и наименование специальности

Руководитель практики от организации:

должность подпись расшифровка подписи

МП

Руководитель практики от колледжа:

преподаватель *А.В.Афанасьев*

должность подпись расшифровка подписи

Курган 2022

Содержание

[Введение 3](#_Toc106110577)

[1. Аналитическая часть 4](#_Toc106110578)

[1.1. Характеристика программных средств для создания приложения 4](#_Toc106110579)

[1.2 Функции и параметры программных средств, используемых для решения задач 5](#_Toc106110580)

[2. Проектная часть 10](#_Toc106110581)

[2.1 Структура базы данных 10](#_Toc106110582)

[2.2 Структура визуального интерфейса 11](#_Toc106110583)

[Заключение 15](#_Toc106110584)

[Литература 16](#_Toc106110585)

Введение

Было получено задание от руководителя практики на производстве ООО «Механический завод» по автоматизации учета данных. Автоматизация заключалась в хранение данных о продукции, количество которой осталось на складе в конце каждого месяца. Данные в программу сотрудник вводит каждый месяц. Был получен наглядный пример хранение данных в Microsoft Excel.

Цель: Написание программы для хранения данных о продукции

Для удовлетворения поставленной цели, необходимо выполнить ряд задач:

* создать базу данных для хранения информации;
* построить макет приложение;
* написать код для ввода и вывода информации о продукции при использовании базы данных.

1. Аналитическая часть

1.1. Характеристика программных средств для создания приложения

Для реализации цели были использованы следующие программы:

- Microsoft Visual Studio - это панель запуска для написания, отладки и создания кода для приложений, а затем их публикации. В дополнение к стандартному редактору и отладчику, которые есть в большинстве IDE, Visual Studio включает компиляторы, надстройки кода, графические дизайнеры и многие другие функции, расширяющие возможности разработки. Visual Studio включает редактор исходного кода, который поддерживает технологию IntelliSense и позволяет легко перерабатывать код. Встроенный отладчик может действовать как отладчик уровня исходного кода или уровня машины. Другие встраиваемые инструменты включают редактор форм, упрощающий создание графического пользовательского интерфейса приложения, веб-редактор, конструктор классов и конструктор схемы базы данных. Visual Studio позволяет создавать и подключать сторонние надстройки (расширения) для расширения функциональности практически на каждом уровне, включая поддержку систем контроля версий исходного кода (таких как Subversion и Visual SourceSafe), добавление новых наборов инструментов (таких как domain- определенные языки программирования), редактирование кода и визуальный дизайн) или другие аспекты процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

- Material Design In XAML - дополнение (плагин), разработанное компанией Microsoft, для удобного создания графического интерфейса в приложение WPF

- Database Designer - онлайн-инструмент проектирования и моделирования баз данных, графическое представление таблиц и связей. Помимо создания SQL-скрипта для всех основных движков баз данных, также можно экспортировать свою базу данных в формате image (PNG) или pdf. SQL скрипт создается автоматически после визуального построения модели и поддерживается следющие движки баз данных: MySQL, PostgreSQL, MS SQL, Oracle и SQLite.

- Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, созданный совместно Microsoft и Sybase. Transact-SQL — это реализация стандартного языка структурированных запросов ANSI/ISO с расширениями. Он используется для работы с базами данных, начиная от персональных и заканчивая крупномасштабными базами данных

- Microsoft Excel - программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA (Visual Basic for Application). Microsoft Excel входит в состав Microsoft Office.

1.2 Функции и параметры программных средств, используемых для решения задач

Для реализации кода в Microsoft Visual Studio был использован язык программирования С#. Для реализации дизайна, язык разметки XAML.

C# — это объектно - и компонентно-ориентированный язык программирования. C# предоставляет языковые конструкции прямая поддержка такой концепции работы. Это делает C# подходящим для создания и использования программных компонентов.

С момента своего появления C# был обогащен функциями, поддерживающими новые рабочие нагрузки и современные передовые методы разработки программного обеспечения. Программы C# выполняются в .NET, виртуальной системе выполнения, которая вызывает Common Language Runtime (CLR) и библиотеки классов. Common Language Runtime (CLR) — это международная стандартная реализация Microsoft Common Language Infrastructure (CLI). CLI является основой для создания сред выполнения и разработки, в которых языки и библиотеки прозрачно взаимодействуют друг с другом.

Исходный код, написанный на C#, переводится на промежуточный язык (IL), соответствующий спецификациям CLI. Код и ресурсы IL, включая растровые изображения и строки, хранятся в одной сборке, обычно с расширением .dll. Компиляция содержит манифест, содержащий информацию о типах, версии и культуре компиляции.

Когда программа C# выполняется, сборка загружается в среду CLR. CLR JIT преобразует код IL в инструкции машинного языка. Среда CLR также выполняет другие операции, такие как автоматическая сборка мусора, обработка исключений и управление ресурсами. Код, выполняемый в общеязыковой среде выполнения, иногда называют «управляемым кодом», чтобы отличить этот подход от «неуправляемого кода» который немедленно переводится на машинный язык для данной платформы. Обеспечение взаимодействия между языками — ключевая особенность .NET. IL-код, сгенерированный компилятором C#, соответствует стандарту Common Type Specification (CTS). Код IL, сгенерированный из кода C#, может работать с кодом, сгенерированным из языковых версий .NET F#, Visual Basic, C++ и более чем 20 CTS-совместимых языков. Сборка может содержать несколько модулей, написанных на разных языках. NET, и все типы могут ссылаться друг на друга, как если бы они были написаны на одном языке.

В дополнение к службам среды выполнения .NET включает расширенные библиотеки. Эти библиотеки поддерживают широкий спектр рабочих нагрузок. Они организованы в пространства имен, которые предоставляют полезные опции от операций ввода и вывода файлов до манипуляций со строками и анализа XML, от фреймворков веб-приложений до элементов управления Windows Forms. Приложения C# обычно широко используют библиотеку классов .NET для решения общих задач.

Работа над C# началась в декабре 1998 года. Проект получил кодовое название COOL (объектно-ориентированный язык в стиле C).

Первая бета-версия C# 1.0 была выпущена летом 2000 года, а в феврале 2002 года это была финальная версия языка в дополнение к Microsoft Visual Studio. Поскольку C# сочетает в себе лучшие черты популярных языков программирования, которые предшествовали ему в C, Java и C++, программисты могут легко перейти на C# на основе своих знаний любого из этих языков.

Основным отличием от предшественников была возможность прописывать ингредиенты. C # представил такие функции, как события, методы и свойства.

Финальный релиз C# 2.0 вышел в 2005 году. Это укрепило ее позиции на рынке. Новые функции, такие как анонимные методы, частичные и параметризованные типы, значительно расширили возможности использования C#.

Во второй версии добавлены 64-битные вычисления, что открыло возможность увеличения адресного пространства. Триггеры, хранимые процедуры и типы данных также были созданы на языках .NET.

Версия 3.0, выпущенная в 2008 году, позволила C# занять лидирующие позиции в «гонке вооружений» языков. Новым в C# 3.0 является интегрированный язык запросов (LINQ); неявно введенные переменные и методы расширения; лямбда-выражения, которые назначают реализацию кода делегатам с использованием нового, более простого синтаксиса.

В частности, лямбда-выражения можно «отличить» вместе с языком интегрированных запросов.

Переменные анонимного типа позволили избежать неудобств и прямого неудобства описания переменных, позволяя объявлять новый тип непосредственно при создании. Также новым в C# 3.0 являются так называемые «ленивые вычисления», которые выполняют необходимые вычисления только в том случае, если запрашиваются соответствующие релевантные данные.

C# 4.0 был выпущен в 2010 году. Основное дополнение к предыдущим версиям с именем и необязательными аргументами. Первый позволяет связать аргумент и параметр по имени, а второй позволяет указать аргумент по умолчанию для каждого параметра. Не менее важным нововведением является динамический тип. Позволяет проверять соответствие типов объектов не на этапе компиляции, а непосредственно во время выполнения программы.

В то же время в .NET Framework 4.0 были внедрены нововведения — параллельная версия библиотеки распараллеливания задач (TPL) и интегрированный язык запросов (PLNQ). Их поддержка позволяет выполнять код параллельно на компьютерах с многоядерными или несколькими одноядерными процессорами.

XAML - основанный на XML язык разметки для декларативного программирования приложений, разработанный Microsoft. XAML широко используется в .NET Framework 3.0, особенно в Windows Presentation Foundation (WPF), Xamarin.Forms, Windows Workflow Foundation (WWF) и Silverlight. В WPF XAML — это язык разметки пользовательского интерфейса для определения элементов пользовательского интерфейса, привязки данных, поддержки событий и других свойств. В WWF XAML можно использовать для определения рабочих процессов.

Файлы XAML можно создавать и редактировать с помощью инструментов визуального дизайна, таких как Microsoft Expression Blend, Microsoft Visual Studio, WPF Visual Designer. Их также можно создавать с помощью стандартного текстового редактора, редактора кода, такого как XAMLPad, или графического редактора, такого как Vectropy.

Все, что создано или реализовано на XAML, может быть выражено на традиционных языках .NET, таких как C# или Visual Basic.NET. Однако ключевым аспектом технологии является снижение сложности инструментов, используемых для обработки XAML, поскольку XAML основан на XML. В результате существует множество продуктов, которые создают приложения на основе XAML. Поскольку XAML основан на XML, разработчики и дизайнеры могут работать с содержимым одновременно без необходимости перевода.

WPF (Windows Presentation Foundation) - это .NET платформа для создания клиентских приложений для настольных систем. Данный видео курс предназначен для тех, кто владеет языком C# на базовом уровне и хочет развиваться в сфере разработки настольного программного обеспечения.

2. Проектная часть

2.1 Структура базы данных

Диаграмма структуры базы данных (Рисунок 2.1.1) хранит в себе информацию о продукте: название, номер, количество, единица измерения, примечание и изображение.

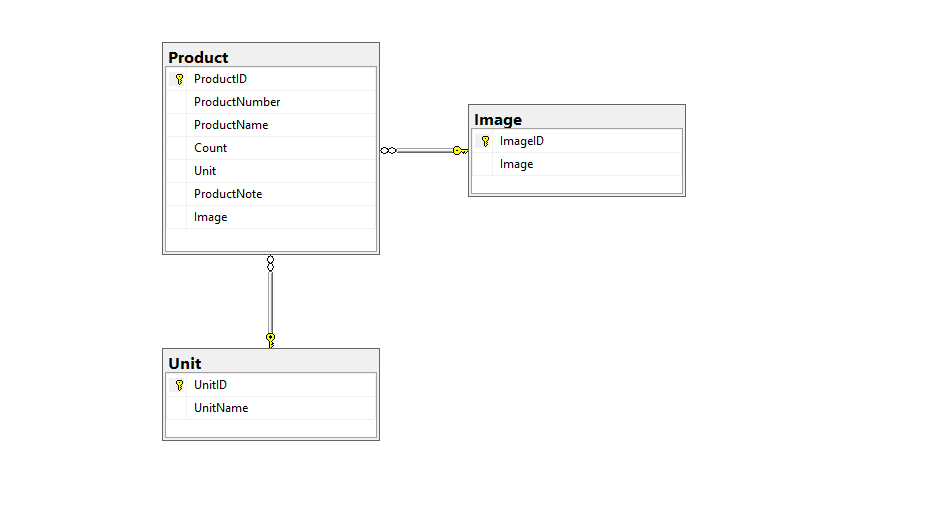


Рисунок 2.1.1 – Диаграмма структуры базы данных

Схема базы данных описывает содержимое, структуру и ограничения целостности, используемые для создания и обслуживания базы данных.

Постоянные данные в среде базы данных содержат схему и базу данных. Система управления базами данных (СУБД) использует определения данных в схеме для управления доступом и доступом к данным в базе данных.

Схема БД — описана на формальном языке, поддерживаемом СУБД. В реляционных базах данных схема определяет таблицы, поля в каждой таблице (обычно с их именем, типом, обязательным) и ограничениями целостности.

Схемы обычно хранятся в словаре данных. Хотя схема определяется как текст на языке базы данных, этот термин часто используется для графического представления структуры базы данных.

Основными объектами в графическом представлении схемы являются таблицы и отношения, определяемые внешними ключами.

На диаграмме наглядно видно, за что отвечает таблицы. Таблица «Product» отвечает за хранение общей информации. Таблица «Unit» хранит в себе информацию о единице измерения. Таблица «Image» хранит в себе изображение продукта.

2.2 Структура визуального интерфейса

Интерфейс ввода данных (Рисунок 2.2.1) отвечает за ввод информации в базу данных.

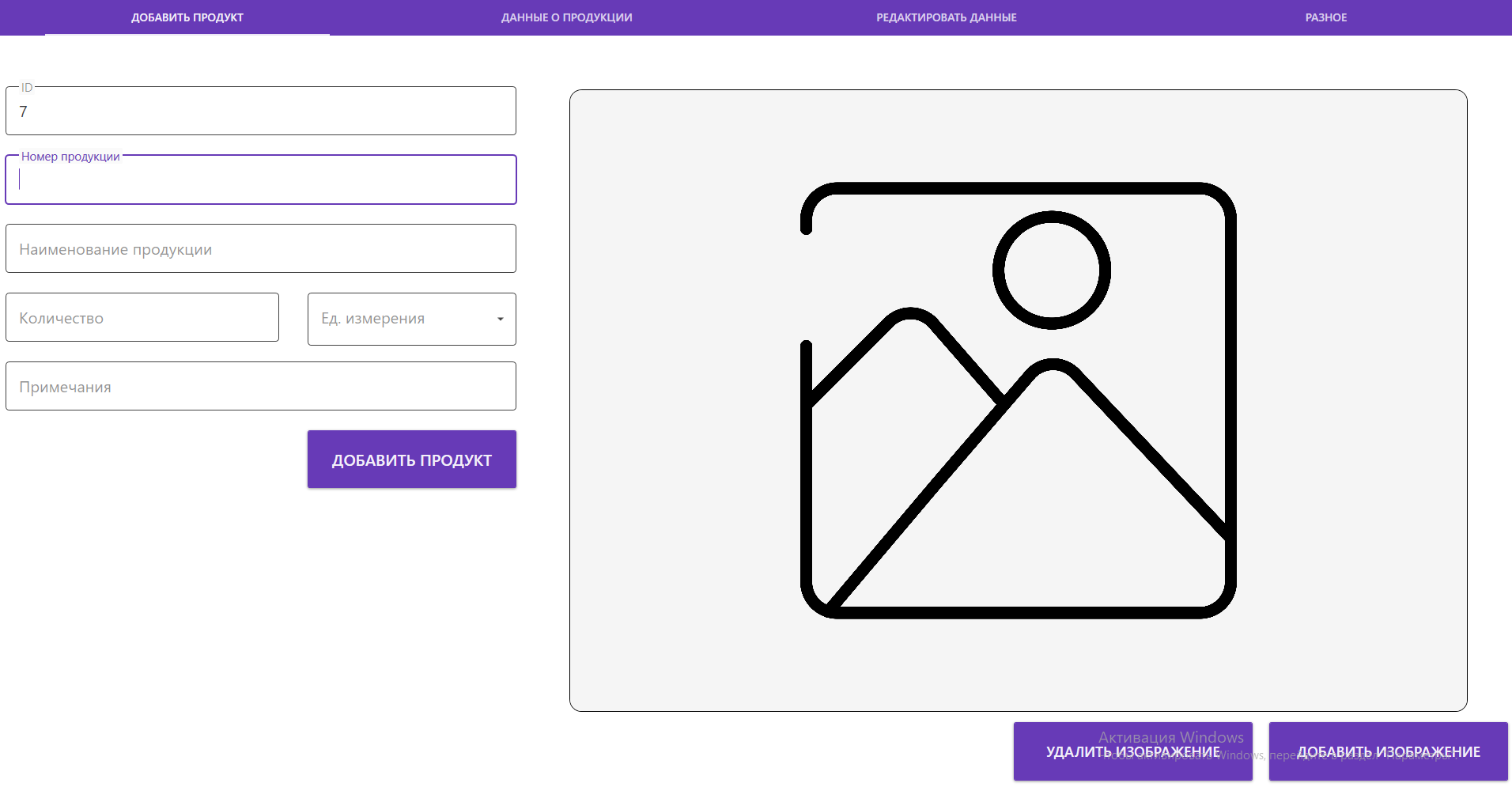


Рисунок 2.2.1 – Интерфейс ввода данных

Пользователю необходимо обязательно заполнить поля: «Номер продукции», «Количество». По предоставленным данным от организации некоторых данных может не быть. Есть возможность добавления фотографии для товара, для этого необходимо нажать на кнопку «Добавить изображение».

Интерфейс редактирования данных (Рисунок 2.2.2) отвечает за изменение информации о продукции в базе данных.

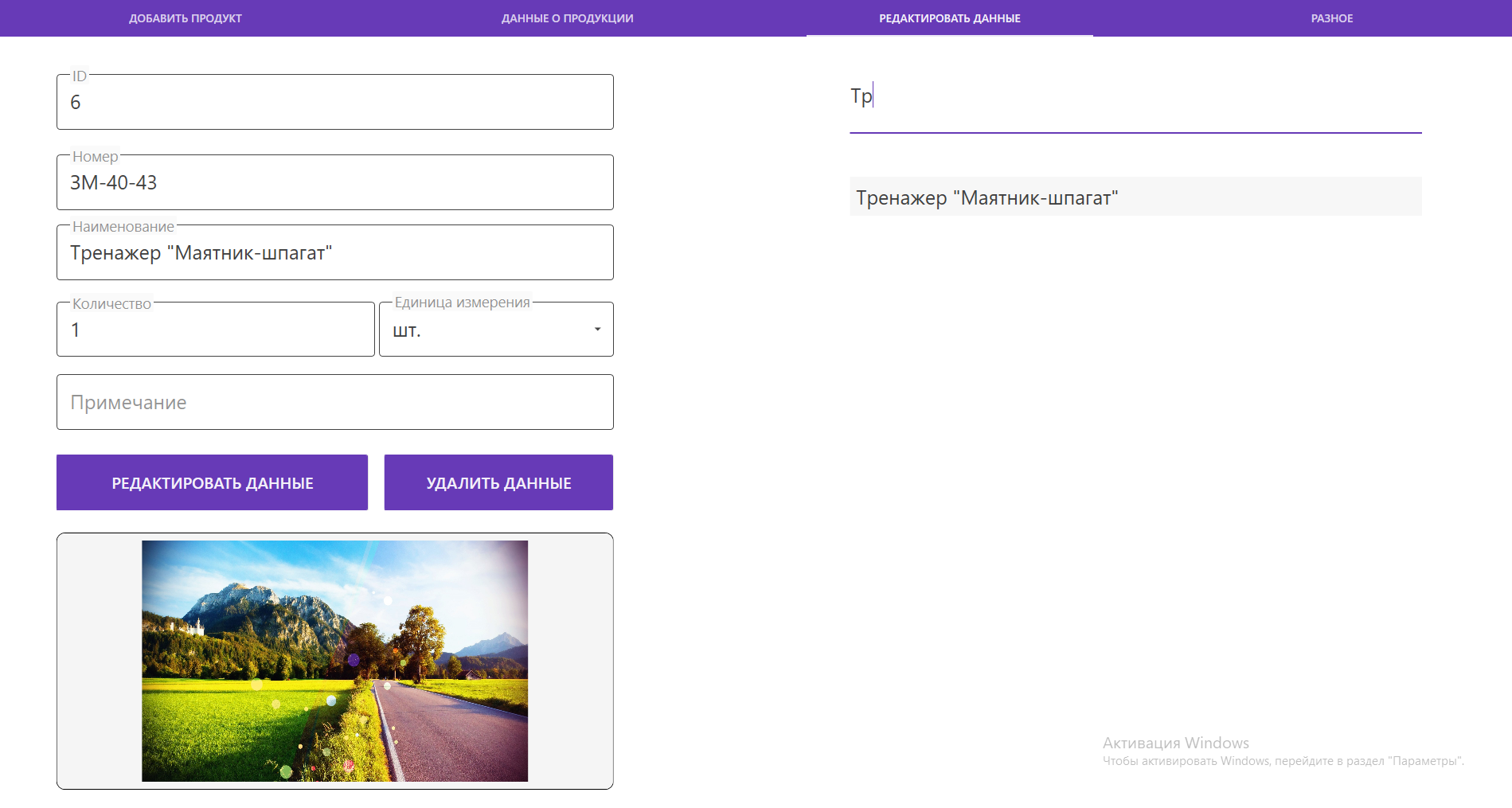


Рисунок 2.2.2 – Интерфейс редактирования данных.

Для того, чтоб отредактировать данные о продукции пользователю необходимо нажать на раздел «Редактировать данные. Далее необходимо выбрать продукт, нажатием на него. Не предусмотрена возможно смены ID продукта, так как он является первичным ключом в таблице данных, другие данные отредактировать можно. На странице реализован поиск продукции, притом условие если продукции в базе будет много и визуальный поиск не удобен.

Если пользователь перейдет во вкладку «Разное», у него появится возможность добавления наименования единицы измерения. Данный раздел был создан не только для добавления единицы измерения, но и для дальнейших обновлений, если потребуется место для реализации чего-либо.

Во вкладке «Данные о продукции» пользователь наглядно сможет увидеть все продукты, также воспользоваться поиском, если визуальный поиск затруднен (Рисунок 2.2.3).

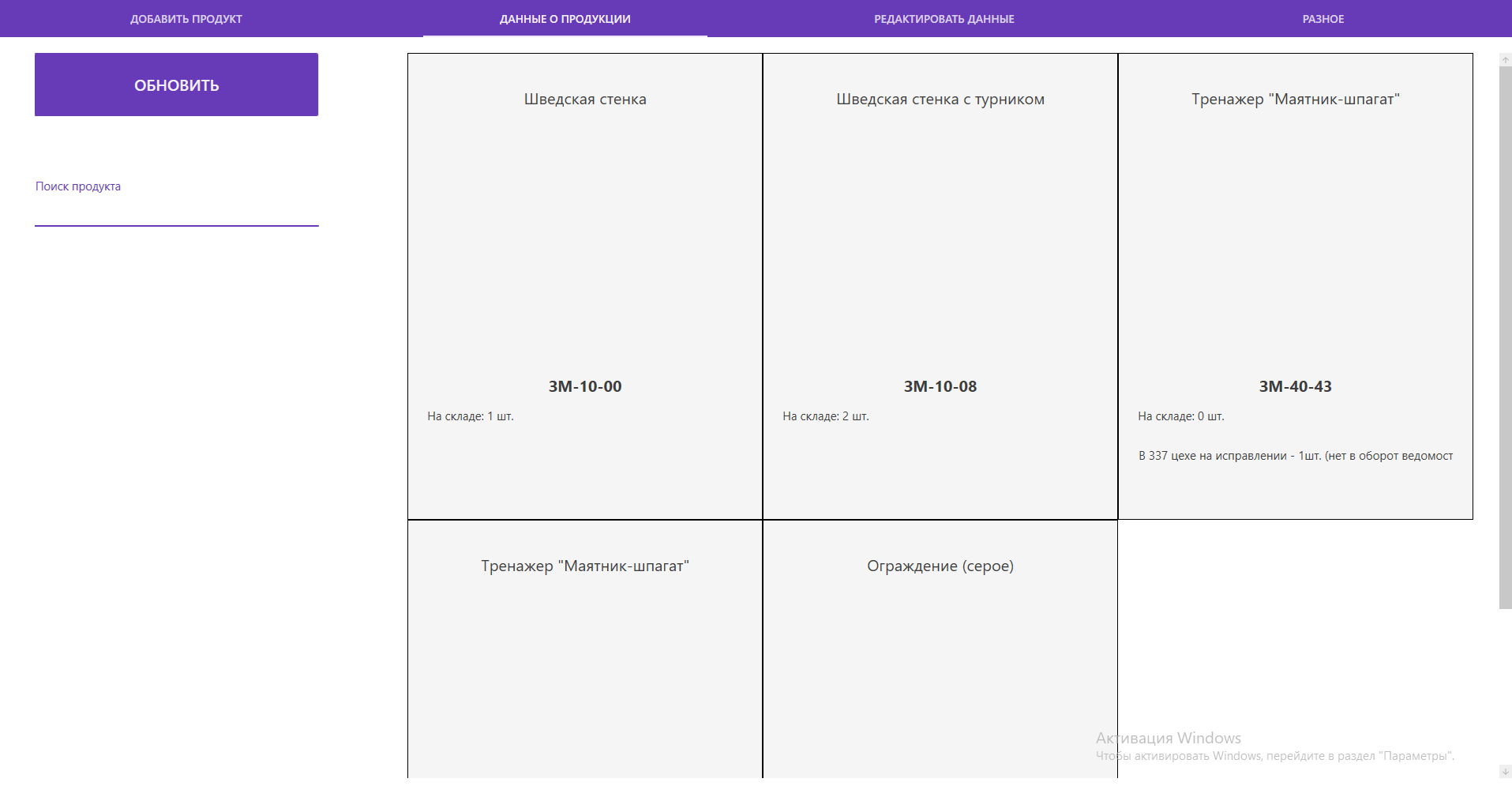


Рисунок 2.2.3 – Интерфейс данных о продукции

Продукция отображается согласно тем данным, которые ввел пользователь на странице «Добавить продукт». В случае, если пользователь отредактировал данные на странице «Редактировать данные», будет отображена актуальная информация.

Информация на странице «Данные о продукции» выводится с помощью элемента ListView, который унаследован от класса ListBox, поэтому может вести себя как простой список. Несмотря на то, что ListView, унаследован от класса ListBox, в нем можно хранить сразу несколько разных элементов (Рисунок 2.2.4).

Наглядно видно, что в ListView хранятся элементы: TextBlock, Image. Указанные элементы выводятся в одной «плитке», данные отображаются автоматически из базы данных, в этом помогла привязка Binding.

В WPF привязка (binding) является мощным инструментом программирования, без которого не обходится ни одно серьезное приложение.

Привязка подразумевает взаимодействие двух объектов: источника и приемника. Объект-приемник создает привязку к определенному свойству объекта-источника. В случае модификации объекта-источника, объект-приемник также будет модифицирован.

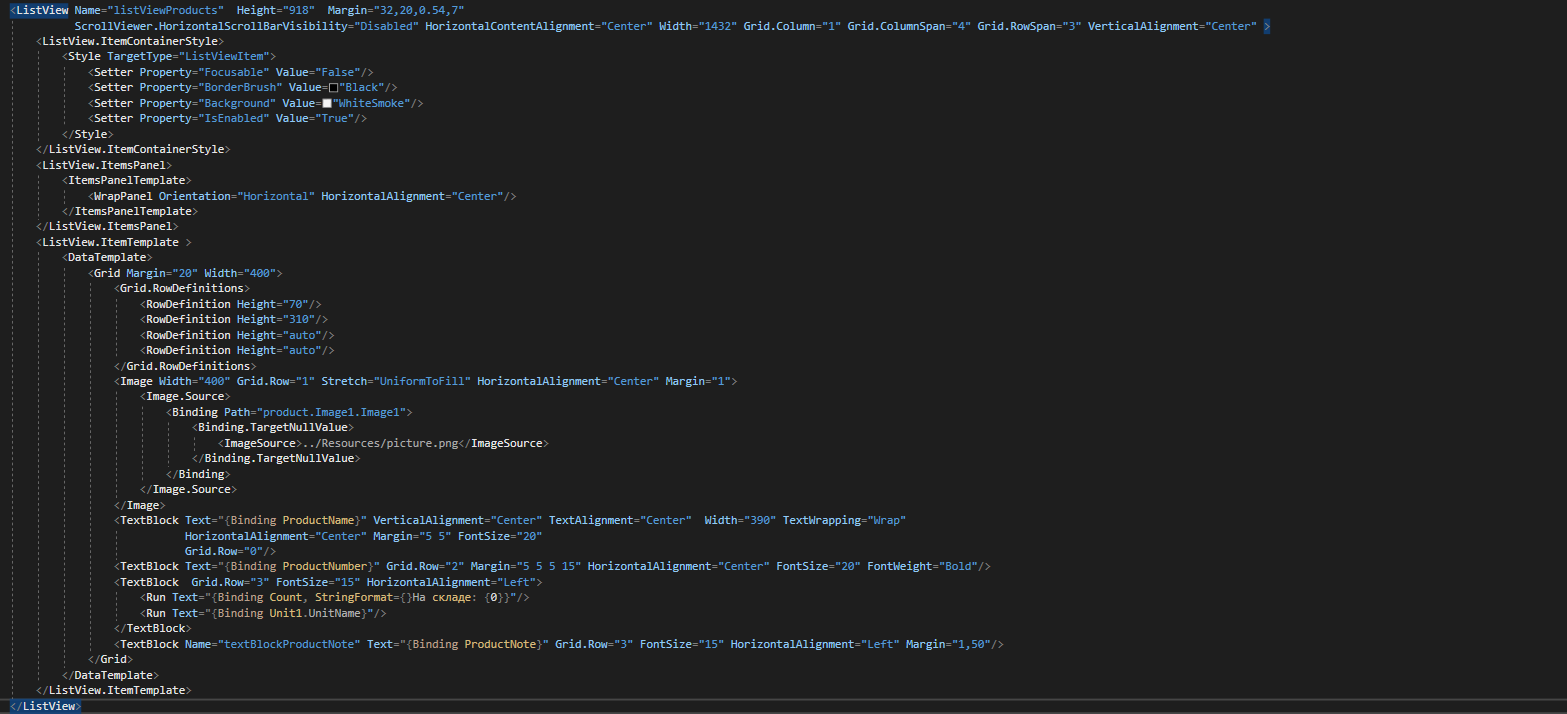


Рисунок 2.2.4 – Использование ListView

То есть в данном случае у нас поле из базы данных является источником, а TextBlock - приемником привязки. Свойство Text элемента TextBlock привязывается к свойству Text элемента из базы данных. В итоге если будут внесены изменения в базу данных, они будут отображены в текстовом блоке, за которым полем осуществлена привязка.

Заключение

На производственной практике была рассмотрена работа по автоматизации учета данных о продукции, на производстве ООО «Завод механический». Разработано программное обеспечение по поставленной цели. Можно наглядно увидеть, как происходит и функционирует это приложение.

В ходе написания проекта была освоена и закреплена работа со средой разработки Microsoft Visual Studio и языком программирования C#, а также Microsoft SQL Server. Цели по закреплению теоретических знаний была достигнута, полученных при изучении курса «», путем создания приложения, представляющего собой программную реализацию работы складского помещения и хранение данных в программе.

В программном обеспечение реализованы функции:

* Хранение данных
* Ввод данных
* Изменение данных
* Вывод данных

По итогу мы получили готовый продукт, отлично подходящий для учета данных склада.

Литература

1. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джефри. - М.: Питер, 2020. - 656 c.
2. Нейгел, К. C# 2005 для профессионалов / К. Нейгел. - М.: Вильямс, 2020. – 927 c.
3. Милов, А. В. Основы программирования в задачах и примерах / А.В. Милов. - М.: Фолио, 2018. - 400 c.
4. Основы языка программирования C#: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/C\_Sharp
5. Microsoft Visual Studio [Электронный ресурс]. URL <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>
6. Microsoft SQL Server [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/Microsoft_SQL_Server>
7. Microsoft Excel [Электронный ресурс] URL <https://ru.wikipedia.org/wiki/Microsoft_Excel>
8. XAML [Электронный ресурс] URL https://ru.wikipedia.org/wiki/XAML