Частное учреждение образования

«Колледж бизнеса и права»

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

«Программа автоматизации работы магазина по продаже холодного оружия»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту по дисциплине

«Конструирование программ и языки программирования»

КП Т.593005.401

Руководитель курсового проекта (Н.В. Ржеутская)

Учащийся (Д. А. Зеленко)

2018

Содержание

Изм.

Лист

№ докум.

Подпись

Датаа

Лист

3

кп Т.593005.401 ПЗ

Разраб.

Д.А.Зеленко.

Провер.

Н.В.Ржеутская

Реценз.

Н. Контр.

Утверд.

*«Программа автоматизации работы магазина по продаже холодного оружия»*

Лит.

Листов

КБП

50

у

[Введение 3](#_Toc514758975)

[1 Постановка задачи 4](#_Toc514758976)

[1.1 Организационная сущность задачи 4](#_Toc514758977)

[1.2 Информационная модель 4](#_Toc514758978)

[2 Вычислительная система 7](#_Toc514758979)

[2.1 Используемые технические средства 7](#_Toc514758980)

[2.2 Инструменты разработки 7](#_Toc514758981)

[3 Проектирование задачи 10](#_Toc514758982)

[3.1 Требования к приложению 10](#_Toc514758983)

[3.2 Концептуальный прототип 10](#_Toc514758984)

[3.3 Организация данных 11](#_Toc514758985)

[3.4 Функции и элементы управления 16](#_Toc514758986)

[3.5 Проектирование справочной системы 18](#_Toc514758987)

[4 Описание программного средства 19](#_Toc514758988)

[4.1 Общие сведения 19](#_Toc514758989)

[4.2 Функциональное назначение 19](#_Toc514758990)

[4.3 Входные и выходные данные 19](#_Toc514758991)

[5 Методика испытаний 20](#_Toc514758992)

[5.1 Технические требования 20](#_Toc514758993)

[5.2 Функциональное тестирование 20](#_Toc514758994)

[6 Применение 26](#_Toc514758995)

[6.1 Назначение программы 26](#_Toc514758996)

[6.2 Условия применения 26](#_Toc514758997)

[6.3 Справочная система 26](#_Toc514758998)

[Заключение 27](#_Toc514758999)

[Список используемых источников 28](#_Toc514759000)

[Приложение А 29](#_Toc514759001)

[Приложение Б 45](#_Toc514759002)

# Введение

На сегодняшний день в магазинах существуют "программы автоматизации работы магазина", которые помогают продавцам в сортировке, поиске товара, ведении продаж,

составлении отчётов.

Целью разработки проекта на тему «Программа автоматизации работы магазина по продаже холодного оружия» является создание программы, которая позволит покупать различные виды холодного оружия.

Пояснительная записка к курсовому проекту состоит из шести разделов, содержащих необходимую информацию по организации эксплуатации программного приложения.

В первом разделе «Постановка задачи» раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть автоматизированы. Описывается задача, перечисляются основные функции программы. Строится информационная модель, отражающая сущности задачи, их свойства и взаимосвязи.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения. Описываются новые возможности программы, а также ее отличия от предыдущих версий.

В третьем разделе «Проектирование задачи» проводится объектно-ориентированный анализ задачи, строится концептуальный прототип системы меню, диалоговых окон и элементов управления.

В четвертом разделе «Описание программного средства» представлены общие сведения о программном средстве и его функциональном назначении.

В пятом разделе «Методика испытаний» описываются требования к техническим средствам для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации, требования к информационной и программной совместимости. Представляются результаты функционального и полного тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства и области его применения. В этом разделе приводится структура справочной системы, а также методика ее использования.

В заключении будет проанализировано созданное программное приложение, определена степень соответствия поставленной задачи и выполненной работы.

Приложение будет содержать текст программы.

В графической части будут представлены диаграммы вариантов использования, классов, деятельности, последовательности и компонентов.

# 1 Постановка задачи

## 1.1 Организационная сущность задачи

Рассмотрим предметную область задачи. Магазин холодного оружия зависит от таких показателей, как: база данных холодного и огнестрельного оружия.

Задачей данного курсового проекта является автоматизация работы магазина, который занимается продажей оружия. Программное приложение должно корректно выполнять выбранное действие пользователем.

Исходя из исследования предметной области, основными задачами, подлежащими автоматизации, будут являться:

- выбор действия над списком оружия;

- возможность добавления выбранного оружия в корзину;

- возможность покупать выбранное оружие;

- возможность составление отчётов о поступлении и продаже оружия;

- предоставление справочной информации о поступление / убытие оружия.

Данные необходимы, сохранять в виде базы данных Access.

В настоящее время существуют аналогичные программы: «Каталог Онлайнер», «1к.by», «ttn.by» и другие.

Однако разработка программы, решающий подобный круг задач, только расширит спектр предлагаемых товаров на рынке программного обеспечения.

## 1.2 Информационная модель

Цель моделирования данных состоит в обеспечении разработчика информационной системы концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных.

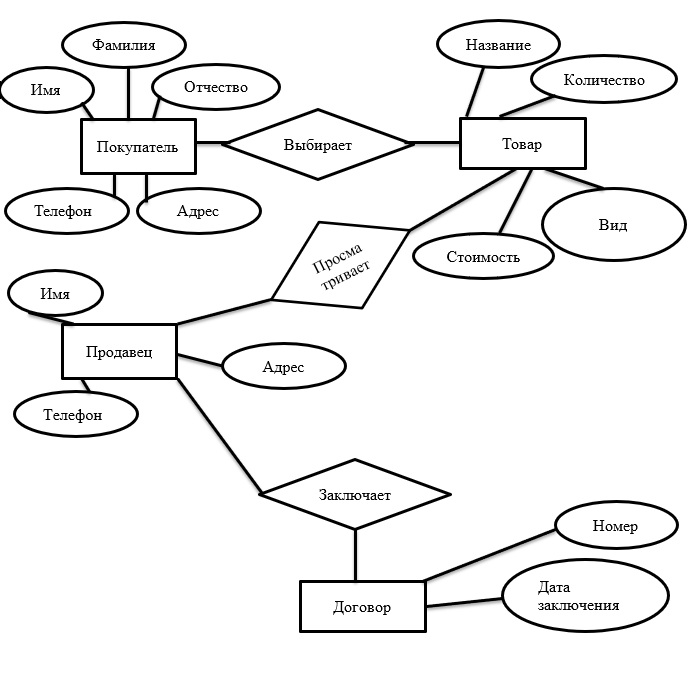
Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных. Нотация ERD была впервые введена П. Ченном и получила дальнейшее развитие в работах Баркера. Диаграмма «сущность-связь» представлена на рисунке 1.

Рисунок 1 – Диаграмма «сущность-связь»

Исходя из исследования предметной области, можно выделить следующие сущности разработки: покупатель, товар, договор, продавец.

Для сущности «Покупатель» можно выделить следующие атрибуты:

- имя;

- фамилия;

- отчество;

- телефон;

- адрес.

Для сущности «Товар» атрибутами будут являться:

- название;

- количество;

- стоимость;

- критерий сортировки.

Для сущности «Продавец» атрибутами будут являться:

- имя;

- телефон;

- адрес.

Для сущности «Договор» атрибутами будут являться:

- номер;

- дата заключения

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей, взаимодействующих с системой с помощью вариантов использования.

Варианты использования описывают взаимодействия между продавцом и сущностью.

Продавец представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения целей.

Данный программный продукт имеет следующие основные (Include) функции:

- осуществить работу с метательным оружием;

- сортировать список метательного оружия;

- выбрать критерий сортировки метательного оружия;

- осуществить поиск метательного оружия;

- выбрать критерий поиска;

- добавить в корзину;

- осуществить работу с огнестрельным оружием;

- сортировать список огнестрельного оружия;

- выбрать критерий сортировки огнестрельного оружия;

- осуществить поиск огнестрельного оружия;

- выбрать критерий поиска;

- прочитать данные из БД;

- вывести статистику;

- вывести список поступивших / купленных оружий.

К вспомогательным (Exten) функциям, расширяющим возможности системы относятся следующие функции:

- предоставить справочную систему;

- предоставить формирование списка холодного оружия;

- предоставить формирование списка огнестрельного оружия.

Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма классов для проектируемой системы представлена в графической части на листе 2.

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Диаграмма деятельности для функции поиска информации представлена в графической части на листе 3.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы.

Диаграмма последовательности для проектируемой системы представлена в графической части на листе 4.

Рассмотренные ранее диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления. Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют самостоятельного материального воплощения. Другими словами, различные элементы логического представления, такие как классы, ассоциации, состояния, сообщения, не существуют материально или физически. Они лишь отражают наше понимание структуры физической системы или аспекты ее поведения.

Основное назначение логического представления состоит в анализе структурных и функциональных отношений между элементами модели системы. Однако для создания конкретной физической системы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели.

Диаграмма компонентов описывает объекты реального мира - компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Вид диаграммы компонентов для данной проектируемой системы представлен в графической части на листе 5.

# 2 Вычислительная система

## 2.1 Используемые технические средства

Конфигурация компьютера, на котором будет разрабатываться программное приложение:

- процессор Intel Core i5 7600 3,50Ггц;

- оперативная память DDR4 8ГБ;

- жёсткий диск HDD Toshiba 2ТБ;

- видеокарта AMD Radeon R7 370 4ГБ;

- материнская плата Asus;

- монитор Viewsonic 24”;

- клавиатура Logitech k280e;

- мышь Ozon Neon Red.

## 2.2 Инструменты разработки

Инструментами разработки будут являться:

- операционная система Windows 10 Pro;

- среда Microsoft Visual Studio 2017;

- язык программирования С#;

- программа для создание базы данныхMicrosoft Access 2016;

- программа для создание справки: Help & Manual;

- программа для создание диаграмм: Umlet;

- программа для создание чеков: MS Word;

- программа для создания инстоллятора: Smart Install Maker;

- программа для создание отчётов: MS Excel.

Windows 10 — [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) для [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [рабочих станций](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в рамках семейства [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT). После [Windows 8.1](https://ru.wikipedia.org/wiki/Windows_8.1) система получила номер 10, минуя 9. Серверный аналог Windows 10 — [Windows Server 2016](https://ru.wikipedia.org/wiki/Windows_Server_2016). Система призвана стать единой для разных устройств, таких как [персональные компьютеры](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), [планшеты](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD%D1%88%D0%B5%D1%82%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), [смартфоны](https://ru.wikipedia.org/wiki/%D0%A1%D0%BC%D0%B0%D1%80%D1%82%D1%84%D0%BE%D0%BD), консоли [Xbox One](https://ru.wikipedia.org/wiki/Xbox_One) и пр. Доступна единая платформа разработки и единый [магазин](https://ru.wikipedia.org/wiki/Windows_Store) [универсальных приложений](https://ru.wikipedia.org/wiki/%D0%A3%D0%BD%D0%B8%D0%B2%D0%B5%D1%80%D1%81%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0_Windows), совместимых со всеми поддерживаемыми устройствами. Windows 10 [поставляется в качестве услуги](https://ru.wikipedia.org/wiki/SaaS) с выпуском обновлений на протяжении всего цикла поддержки. В течение первого года после выхода системы пользователи могли бесплатно обновиться до Windows 10 на устройствах под управлением лицензионных копий [Windows 7](https://ru.wikipedia.org/wiki/Windows_7), [Windows 8.1](https://ru.wikipedia.org/wiki/Windows_8.1) и [Windows Phone 8.1](https://ru.wikipedia.org/wiki/Windows_Phone_8.1). Среди значимых нововведений — [голосовая помощница](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%BB%D0%BB%D0%B5%D0%BA%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B0%D0%B3%D0%B5%D0%BD%D1%82) [Кортана](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D1%82%D0%B0%D0%BD%D0%B0_(%D0%B3%D0%BE%D0%BB%D0%BE%D1%81%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%BC%D0%BE%D1%89%D0%BD%D0%B8%D1%86%D0%B0)), возможность создания и переключения нескольких рабочих столов и др. Windows 10 — последняя «коробочная» версия [Windows](https://ru.wikipedia.org/wiki/Windows), все последующие версии будут распространяться исключительно в цифровом виде.[9]

Microsoft Visual Studio — линейка продуктов компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft), включающих [интегрированную среду разработки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как [консольные](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) [приложения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), так и приложения с [графическим интерфейсом](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F), в том числе с поддержкой технологии [Windows Forms](https://ru.wikipedia.org/wiki/Windows_Forms), а также [веб-сайты](https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82), [веб-приложения](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), [веб-службы](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0) как в [родном](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4), так и в [управляемом](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%8F%D0%B5%D0%BC%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) кодах для всех платформ, поддерживаемых [Windows](https://ru.wikipedia.org/wiki/Windows), [Windows Mobile](https://ru.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://ru.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), [Xbox](https://ru.wikipedia.org/wiki/Xbox), [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone) [.NET Compact Framework](https://ru.wikipedia.org/wiki/.NET_Compact_Framework) и [Silverlight](https://ru.wikipedia.org/wiki/Silverlight).

Visual Studio включает в себя [редактор исходного кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) с поддержкой технологии [IntelliSense](https://ru.wikipedia.org/wiki/IntelliSense) и возможностью простейшего [рефакторинга кода](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3). Встроенный [отладчик](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и дизайнер [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Visual Studio позволяет создавать и подключать сторонние дополнения ([плагины](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD)) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем [контроля версий исходного кода](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8) (как, например, [Subversion](https://ru.wikipedia.org/wiki/Subversion) и [Visual SourceSafe](https://ru.wikipedia.org/wiki/Microsoft_Visual_SourceSafe)), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на [предметно-ориентированных языках программирования](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)) или инструментов для прочих аспектов [процесса разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F) (например, клиент Team Explorer для работы с [Team Foundation Server](https://ru.wikipedia.org/wiki/Team_Foundation_Server)). [13]

C# (произносится си шарп) — [объектно-ориентированный](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Разработан в [1998](https://ru.wikipedia.org/wiki/1998)—[2001 годах](https://ru.wikipedia.org/wiki/2001_%D0%B3%D0%BE%D0%B4) группой инженеров компании [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) под руководством [Андерса Хейлсберга](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%BB%D1%81%D0%B1%D0%B5%D1%80%D0%B3,_%D0%90%D0%BD%D0%B4%D0%B5%D1%80%D1%81) и Скотта Вильтаумота как язык разработки приложений для платформы [Microsoft .NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework). Впоследствии был стандартизирован как [ECMA](https://ru.wikipedia.org/wiki/ECMA)-334 и [ISO](https://ru.wikipedia.org/wiki/ISO)/[IEC](https://ru.wikipedia.org/wiki/IEC) 23270.C# относится к семье языков с С-подобным синтаксическом, из них его синтаксис наиболее близок к C++ и Java.

Язык имеет [статическую типизацию](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), поддерживает [полиморфизм](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%BE%D1%80%D1%84%D0%B8%D0%B7%D0%BC_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [перегрузку операторов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%B3%D1%80%D1%83%D0%B7%D0%BA%D0%B0_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BE%D0%B2) (в том числе операторов явного и неявного приведения типа), [делегаты](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BB%D0%B5%D0%B3%D0%B0%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), атрибуты, [события](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B1%D1%8B%D1%82%D0%B8%D0%B9%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), [свойства](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [обобщённые](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BE%D0%B1%D1%89%D1%91%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) типы и методы, [итераторы](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80), [анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) с поддержкой [замыканий](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [LINQ](https://ru.wikipedia.org/wiki/Language_Integrated_Query), [исключения](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D1%81%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%BD%D0%B8%D0%B9), [комментарии](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%B8%D0%B8_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) в формате [XML](https://ru.wikipedia.org/wiki/XML).

Переняв многое от своих предшественников — языков [C++](https://ru.wikipedia.org/wiki/C%2B%2B), [Pascal](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%81%D0%BA%D0%B0%D0%BB%D1%8C_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), [Модула](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D0%B0-2), [Smalltalk](https://ru.wikipedia.org/wiki/Smalltalk) и, в особенности, [Java](https://ru.wikipedia.org/wiki/Java) — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает [множественное наследование](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) классов (между тем допускается [множественное наследование интерфейсов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BD%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B8_%D1%80%D0%B5%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%8)). [14]

Microsoft Office Access или просто Microsoft Access — [реляционная система управления базами данных (СУБД)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) корпорации [Microsoft](https://ru.wikipedia.org/wiki/Microsoft). Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и [базами данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Благодаря встроенному языку [VBA](https://ru.wikipedia.org/wiki/VBA), в самом Access можно писать приложения, работающие с базами данных. [Microsoft Jet Database Engine](https://ru.wikipedia.org/w/index.php?title=Microsoft_Jet_Database_Engine&action=edit&redlink=1) ([англ.](https://en.wikipedia.org/wiki/Microsoft_Jet_Database_Engine)), которая используется в качестве [движка базы данных](https://ru.wikipedia.org/wiki/Database_engine) MS Access, является [файл-серверной](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) и потому применима лишь к приложениям, работающим с небольшими объёмами данных и при небольшом числе пользователей, одновременно работающих с этим данными. Непосредственно в Access отсутствует ряд механизмов, необходимых в многопользовательских базах данных, таких, например, как [триггеры](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B8%D0%B3%D0%B3%D0%B5%D1%80_(%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)). [12]

Help & Manual - это инструмент для создания справки, основанный на Windows, опубликованный компанией EC Software, компанией. Как и многие инструменты для создания справки, Help & Manual позволяет писателю создавать один исходный текст, который затем преобразует в несколько целевых форматов. В этом случае автор создает исходный текст, используя редактор, встроенный в программу «Справка и руководство». Текст, наряду с настройками пользователя для проекта, хранится в файлах XML. [15]

UMLet - это [инструмент UML](https://en.wikipedia.org/wiki/UML_tool) с [открытым исходным кодом на](https://en.wikipedia.org/wiki/Open-source) основе [Java](https://en.wikipedia.org/wiki/Java_(programming_language))[,](https://en.wikipedia.org/wiki/UML_tool)предназначенный для обучения [унифицированному языку моделирования](https://en.wikipedia.org/wiki/Unified_Modeling_Language)и для быстрого создания диаграмм UML. Это инструмент рисования, а не инструмент моделирования, поскольку нет базового словаря или каталога объектов многократного использования. UMLet распространяется по [лицензии GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) . UMLet имеет простой пользовательский интерфейс, который использует коды форматирования текста для изменения основных фигур с помощью украшений и аннотаций, поэтому нет леса икон или диалогов списка параметров по-своему. Это требует от пользователя изучения еще одного языка разметки текста, но усилия небольшие, а разметка очевидна для опытного дизайнера UML. [16]

Microsoft Word — [текстовый процессор](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81%D0%BE%D1%80), предназначенный для создания, просмотра и редактирования [текстовых документов](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), с локальным применением простейших форм [таблично](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0)-[матричных](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) алгоритмов. Выпускается [корпорацией Microsoft](https://ru.wikipedia.org/wiki/Microsoft) в составе [пакета](https://ru.wikipedia.org/wiki/%D0%9E%D1%84%D0%B8%D1%81%D0%BD%D1%8B%D0%B9_%D0%BF%D0%B0%D0%BA%D0%B5%D1%82) [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office). Первая версия была написана [Ричардом Броди](https://ru.wikipedia.org/w/index.php?title=%D0%91%D1%80%D0%BE%D0%B4%D0%B8,_%D0%A0%D0%B8%D1%87%D0%B0%D1%80%D0%B4&action=edit&redlink=1) ([Richard Brodie](https://en.wikipedia.org/wiki/Richard_Brodie)) для [IBM PC](https://ru.wikipedia.org/wiki/IBM_PC), использующих [DOS](https://ru.wikipedia.org/wiki/DOS), в 1983 году. Позднее выпускались версии для [Apple Macintosh](https://ru.wikipedia.org/wiki/Macintosh) (1984), [SCO UNIX](https://ru.wikipedia.org/wiki/SCO_UNIX) и [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows) (1989). Текущей версией является Microsoft Office Word 2016 для [Windows](https://ru.wikipedia.org/wiki/Windows) и [macOS](https://ru.wikipedia.org/wiki/MacOS). [10]

Microsoft Excel  — [программа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0) для работы с [электронными таблицами](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%82%D0%B0%D0%B1%D0%BB%D0%B8%D1%86%D0%B0), созданная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft) для [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT) и [Mac OS](https://ru.wikipedia.org/wiki/Mac_OS), а также [Android](https://ru.wikipedia.org/wiki/Android), [iOS](https://ru.wikipedia.org/wiki/IOS) и [Windows Phone](https://ru.wikipedia.org/wiki/Windows_Phone). Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под [Mac OS X](https://ru.wikipedia.org/wiki/Mac_OS_X), язык макропрограммирования [VBA](https://ru.wikipedia.org/wiki/VBA) (Visual Basic for Application). Microsoft Excel входит в состав [Microsoft Office](https://ru.wikipedia.org/wiki/Microsoft_Office) и на сегодняшний день Excel является одним из наиболее популярных приложений в мире. [11]

Microsoft Office Access или просто Microsoft Access — [реляционная система управления базами данных (СУБД)](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) корпорации [Microsoft](https://ru.wikipedia.org/wiki/Microsoft). Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и [базами данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Благодаря встроенному языку [VBA](https://ru.wikipedia.org/wiki/VBA), в самом Access можно писать приложения, работающие с базами данных. [12]

Утилита предоставляет алгоритм Cabinet для сжатия и создания компактных инсталляторов, поддерживает создание многоязычного инсталлятора (до 25 языков), а также предоставляет средства для управления внешним видом программы-установки.

Среди возможностей Smart Install Maker, можно выделить создание деинсталлятора, ярлыков в указанных каталогах системы, регистрацию новых шрифтов или расширений в операционной системе, поддержку ActiveX и переменных, проверку установленного .NET Framework, разделение инсталлятора на установочные диски, детальную настройку информации об установочном файле, системных требований (например, запретить установку программы в Windows 95) а также его внешнего вида и многое другое. [17]

# 3 Проектирование задачи

## 3.1 Требования к приложению

Программное средство должно выполнять все основные функции, определённые на диаграмме вариантов использования:

- осуществить работу с метательным оружием;

- сортировать список метательного оружия;

- выбрать критерий сортировки метательного оружия;

- осуществить поиск метательного оружия;

- выбрать критерий поиска;

- добавить в корзину;

- осуществить работу с огнестрельным оружием;

- сортировать список огнестрельного оружия;

- выбрать критерий сортировки огнестрельного оружия;

- осуществить поиск огнестрельного оружия;

- выбрать критерий поиска;

- прочитать данные из БД;

- вывести статистику;

- вывести список поступивших / купленных оружий.

В данном программном средстве в качестве средств защиты будет выступать личные данные покупателя: фамилия, имя, отчество, год рождения, место проживания, номер телефона.

Основные ограничения:

- в полях «фамилия» , «имя», «отчество» нельзя вводить цифры;

- в полях «год рождения», «номер телефона» нельзя вводить буквы.

## 3.2 Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным диалоговым средством работы пользователя. Разрабатываемое приложение будет содержать несколько форм: две основные формы и пять дополнительных. Такая структура интерфейса позволит классифицировать основные функции программы по группам.

При проектировании концептуального прототипа предполагается, что при запуске программы первой будет загружаться форма «startform». На ней будет отображаться меню с пунктами: «Купить», «О программе», «Выход». При нажатии на пункт «Купить» будет вызвана соответствующая форма.

На форме «startform» будет находиться:

- кнопка «Купить» для перехода на форму «catalog»;

- кнопка «О программе» для перехода в справочную систему;

- кнопка «Выход» для выхода из приложения.

На форме «catalog» будет находиться:

- компонент Menu Strip для быстрой навигации по приложению;

- компонент Picture Box 1 для перехода на форму «metatel»;

- компонент Picture Box 2 для перехода на форму «ognestrel»;

- компонент Picture Box 3 для перехода на форму «korzina»;

- кнопка «в главное меню» для перехода в главное меню.

На форме «metatel» будет находиться:

- компонент DataGrigView для отображение таблицы из базы данных;

- компонент TextBox для ввода данных для поиска товара;

- компонент GroupBox для объединения параметров поиска;

- компонент RadioButton для сортировки товара в таблице;

- компонент ChekBox для сортировки товара в таблице;

- компонент Timer для бегущей строки;

- кнопка «Поиск» для осуществления поиска товара в базе данных;

- кнопка «?» для открытия файла с существующими товарами;

- кнопка «В корзину» для добавление товара в корзину;

- кнопка «Сортировать» для сортировки товара;

- кнопка «Сбросить» для сброса сортировки товара;

- кнопка «Назад» для перехода на форму «catalog».

На форме «ognestrel» будет находиться:

- кнопка «Пистолеты» для перехода на форму «pistols»;

- кнопка «Пистолеты-пулемёты» для перехода на форму «micro pp»;

- кнопка «Штурмовые винтовки» для перехода на форму «shtyrm»;

- кнопка «Снайперские винтовки» для перехода на форму «snapa»;

- кнопка «Дробовики» для перехода на форму «pompa»;

- кнопка «Гранаты» для перехода на форму «granata».

Формы «pistols», «micro pp», «shtyrm», «snapa», «pompa», «granata» имеют одинаковую структуру. Каждая из них содержит:

- компонент DataGrigView для отображение таблицы из базы данных;

- компонент TextBox 1 для ввода данных для поиска товара;

- компонент GroupBox для объединения параметров поиская;

- компонент RadioButton для сортировки товара в таблице;

- компонент ChekBox для сортировки товара в таблице;

- компонент Timer для бегущей строки;

- кнопка «Поиск» для осуществления поиска товара в базе данных;

- кнопка «?» для открытия файла с существующими товарами;

- кнопка «В корзину» для добавление товара в корзину;

- кнопка «Сортировать» для сортировки товара;

- кнопка «Сбросить» для сброса сортировки товара;

- кнопка «Назад» для перехода на форму «catalog».

На форме «korzina» будет находиться:

- компонент DataGrigView для отображение таблицы из базы данных;

- компонент TextBox для заполнения личных данных покупателя;

- кнопка «Купить» для создание чека о покупке;

- кнопка «Очистить » для очищение корзины;

- кнопка «Назад» для перехода на форму «catalog».

Практически на всех формах используется компонент Label для осуществления соответствующих надписей на них.

## 3.3 Организация данных

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель основана на математическом понятии отношения, представлением которого является таблица. В реляционной модели отношения используются для хранения информации об объектах, представленных в базе данных. Отношение имеет вид двухмерной таблицы, в которой строки соответствуют записям, а столбцы - атрибутам. Каждая запись должна однозначно характеризоваться в таблице. Для этого используют первичные и вторичные ключи. Достоинством реляционной модели является простота и удобство физической реализации.

Реляционная модель базы данных подразумевает нормализацию всех таблиц данных. Нормализация - это формальный метод анализа отношений на основе их первичного ключа и функциональных зависимостей, существующих между их атрибутами.

В разделе 1.2 на рисунке 1 представлена схема Баркера-Чена, на которой отображены главные сущности задачи: «Товар», «Поставщик», «Договор». Каждой сущности ставится в соответствие таблица базы данных. Для сущности «Покупатель» - таблица «Покупатели». Для сущности «Товар» - таблицы «Ножи» «Пистолеты», «Пистолеты-пулемёты», «Штурмовые винтовки», «Снайперские винтовки», «Дробовики», «Гранаты». Для сущности «Договор» - таблица «Корзина».

Структура базы данных разрабатываемого программного средства включает девять таблиц.

Структура данных таблиц, и их краткое описание приводится в таблицах 1-9.

Таблица «Ножи» хранит информацию о ножах. Структура приведена в таблице 1.

Таблица 1- Структура таблицы «Ножи»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Вес | int | 5 | Вес товара |
| Длина\_лезвия (мм) | int | 10 | Длина лезвия товара |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Пистолеты» хранит информацию о пистолетах. Структура приведена в таблице 2.

Таблица 2- Структура таблицы «Пистолеты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Патронов\_в\_магазине | int | 5 | Количество патронов в магазине |
| Калибр | int | 6 | Калибр патронов |
| Вес\_без\_патронов | int | 5 | Вес товара без патронов |

Продолжение таблицы 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |
| Вес\_с\_патронами | int | 5 | Вес товара с патронами |

Таблица «Пистолеты-пулемёты» хранит информацию о пистолетах-пулемётах. Структура приведена в таблице 3.

Таблица 3- Структура таблицы «Пистолеты- пулемёты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Патронов\_в\_магазине | int | 5 | Количество патронов в магазине |
| Калибр | int | 6 | Калибр патронов |
| Вес\_без\_патронов | int | 5 | Вес товара без патронов |
| Вес\_с\_патронами | int | 5 | Вес товара с патронами |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Штурмовые винтовки» хранит информацию о штурмовых винтовках. Структура приведена в таблице 4.

Таблица 4- Структура таблицы «Штурмовые винтовки»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Патронов\_в\_магазине | int | 5 | Количество патронов в магазине |
| Калибр | int | 6 | Калибр патронов |
| Вес\_без\_патронов | int | 5 | Вес товара без патронов |
| Вес\_с\_патронами | int | 5 | Вес товара с патронами |

Продолжение таблицы 4.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Снайперские винтовки» хранит информацию о снайперских винтовках. Структура приведена в таблице 5.

Таблица 5- Структура таблицы «Снайперские винтовки»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Патронов\_в\_магазине | int | 5 | Количество патронов в магазине |
| Калибр | int | 6 | Калибр патронов |
| Вес\_без\_патронов | int | 5 | Вес товара без патронов |
| Вес\_с\_патронами | int | 5 | Вес товара с патронами |
| Вид прицела | string | 30 | Вид прицела снайперской винтовки |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Дробовики» хранит информацию о дробовиках. Структура приведена в таблице 6.

Таблица 6- Структура таблицы «Дробовики»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Патронов\_в\_магазине | int | 5 | Количество патронов в магазине |
| Калибр | int | 6 | Калибр патронов |
| Вес\_без\_патронов | int | 5 | Вес товара без патронов |
| Вес\_с\_патронами | int | 5 | Вес товара с патронами |

Продолжение таблицы 6.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Гранаты» хранит информацию о гранатах. Структура приведена в таблице 7.

Таблица 7- Структура таблицы «Гранаты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Тип\_гранаты | string | 30 | Тип гранаты |
| Масса | int | 5 | Масса гранаты |
| Механизм детонации(сек) | int | 5 | Механизм детонации (сек) |
| Радиус\_действия (м) | int | 6 | Радиус действия (м) |
| Количество\_на\_складе | int | 5 | Количество товара на складе |
| Цена | int | 5 | Цена товара |

Таблица «Корзина» хранит информацию о товаров добавленных в корзину . Структура приведена в таблице 8.

Таблица 8- Структура таблицы «Корзина»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| Производитель | string | 30 | Производитель |
| Имя\_товара | string | 30 | Наименование товара |
| Поставщик | string | 30 | Поставщик товара |
| Количество | int | 5 | Количество купленного товара |
| Цена | int | 5 | Цена товара |
| Количество\_на\_складе | int | 5 | Количество товара на складе |

Таблица «Покупатель» хранит информацию о покупателях. Структура приведена в таблице 9.

Таблица 9- Структура таблицы «Покупатель»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип  поля | Размер поля / маска ввода | Описание поля |
| ФИО | string | 50 | ФИО покупателя |
| Год\_Рождения | date | ##.##.#### | Дата рождение покупателя |
| Адрес | string | 30 | Адрес проживания покупателя |
| Телефон | int | 15 | Номер телефона покупателя |

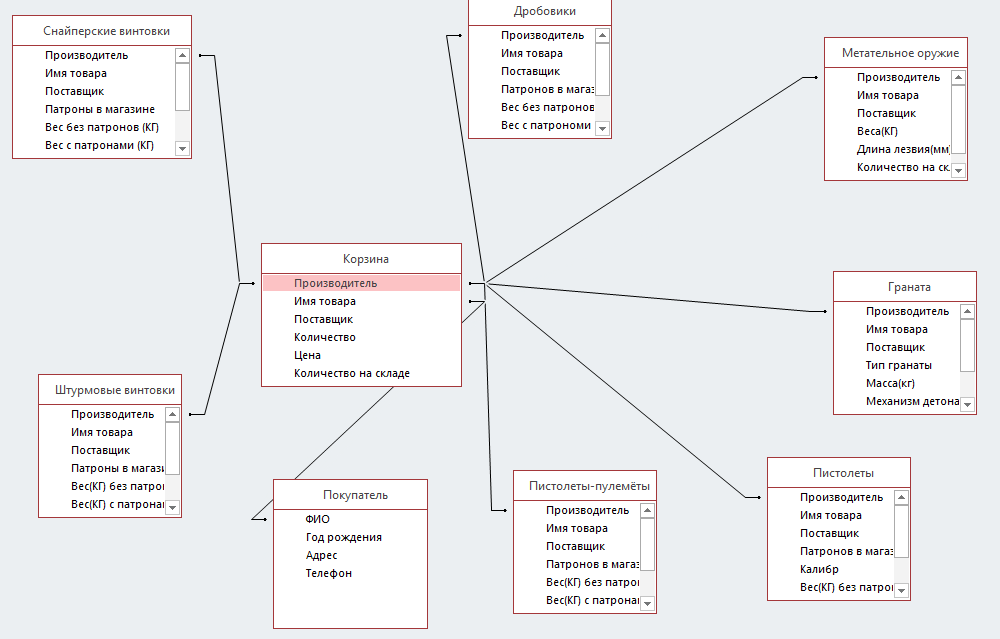
Структура базы данных представлена на схеме на рисунке 2.

Рисунок 2 – Схема данных

## 3.4 Функции и элементы управления

Рассмотрим основные функции программы.

При загрузке формы «metatel» будет вызван метод, позволяющий подключить базу данных к проекту и запускающий метод бегущей строки. Программный код реализации данного метода представлен ниже.

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

{

this.метательное\_оружиеTableAdapter.Fill(this.оружейкаDataSet.Метательное\_оружие);

бегущая\_строка.Start();

}

При нажатии на кнопку «?» на форме «metatel» буден вызван метод, позволяющий открыть текстовый файл для просмотра существующих наименований товара. Программный код реализации данного метода представлен ниже.

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

{try{ System.Diagnostics.Process.Start(@"СписокТоваров\СписокНожей.txt");}

catch (Exception exp){ MessageBox.Show(exp.Message);}}

При нажатии на кнопу «Поиск» на форме «metatel» буден вызван метод, позволяющий осуществить поиск товара по наименованию. Программный код реализации данного метода представлен ниже.

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

{ метательноеОружиеBindingSource.Filter = "[Имя товара]LIKE'" + texst\_poicka.Text + "%'"; }

При нажатии на кнопку «Сортировать» буден вызван метод, позволяющий сортировать список товаров по различным критериям. Часть программного кода реализации данного метода представлен ниже.

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

{

if (ZelGuns.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу ПОСТАВЩИК

if (Горный\_рыцарь.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Великобритания.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Великобритания" + "%'"; } // сортировка по столбцу ПРОИЗВОДИТЕЛЬ

if (Россия.Checked== true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }

if (США.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (Непал.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Непал" + "%'"; }

if (Германия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'"; }

При нажатии на кнопку «Сбросить» буден вызван метод, позволяющий сбросить критерий сортировки товара. Программный код реализации данного метода представлен ниже.

private void Сброс\_Click(object sender, EventArgs e) // сброс сортировки

{

ZelGuns.Checked = false;Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Великобритания.Checked = false; Россия.Checked = false; США.Checked = false; Непал.Checked = false; Германия.Checked = false;

метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";

}

Формы «pistols», «micro pp», «shtyrm», «snapa», «pompa», «granata» содержат аналогичные методы.

Полный текст программы представлен в приложении А.

## 3.5 Проектирование справочной системы

Для корректной работы с приложением требуется обеспечить пользователя справочной системой, в которой будут приведены приемы работы с приложением, включающие данные о том, что произойдёт после нажатия на определенную кнопку или при выборе пункта меню.

Справочная система необходима для ознакомления с программой. В ней должна присутствовать информация, которая поможет в решении проблемы с приложением, а также может напомнить, как пользоваться программным средством.

Система справки данного приложения будет содержать следующие разделы:

- введение;

- о авторе;

- о программе.

Справочная система будет создана с помощью программного средства Help & Manual.

# 4 Описание программного средства

## 4.1 Общие сведения

Приложение называется «Oryzheka.exe».

Основными минимальными требованиями, выдвигаемыми к аппаратному обеспечению персонального компьютера, являются:

- процессор 1.5 МГц и выше;

- оперативная память 200 Мбайт и более;

- монитор;

- клавиатура;

-мышь.

Компьютер должен работать под управлением операционной системы, начиная с Windows 7 и выше.

Объём занимаемой памяти приложения составляет примерно 90-100 Мбайт, объём памяти для файлов примерно составляет 20 Мбайт. Инсталляция производится с установочного диска.

## 4.2 Функциональное назначение

Назначение программного средства на тему «Программа автоматизации работы магазина по продаже холодного оружия» является автоматизация работы магазина, который занимается продажей оружия.

Решаемые задачи курсового проекта:

- возможность сортировать товар по различным критериям сортировки;

- возможность добавлять выбранный товар в корзину;

- возможность составление чеков о покупке товара.

В данном программном средстве в качестве средств защиты будет выступать личные данные покупателя: фамилия, имя, отчество, год рождения, место проживания, номер телефона.

Сетевая поддержка не подсмотрена.

## 4.3 Входные и выходные данные

К входной информации относятся личные данные покупателя, такие как: ФИО, адрес проживания, дата рождения, телефон.

Постоянной информацией являются: наименование поставщика, оружия.

Выходная информация будет представлена в виде сформированного чека о покупке оружия. Шаблон документа представлен в приложении Б. В чеке сначала указывается дата покупки. Далее такие реквизиты как ФИО покупателя, его место проживания, телефон. Далее указывается от какого поставщика покупатель купил оружие и какое оружие он купил. Далее пишется цена оружия и количество. В конце документа указывается место для печати и подписи продавца и покупателя.

# 5 Методика испытаний

## 5.1 Технические требования

Минимальные системные требования к приложению:

* 20 Мб свободного места на винчестере;
* 100 Мб свободной оперативной памяти;
* процессор Intel Pentium 1.5 МГц или совместимый аналог;
* видеокарта с объемом памяти не менее 20 Мб.
* операционная система семейства Microsoft Windows NT;
* клавиатура, мышь.

## 5.2 Функциональное тестирование

В процессе написания программного продукта необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

При запуске программы «Oryzheka.exe» появится главное меню программы. Главное меню представлено на рисунке 5.1

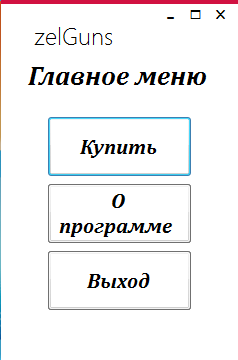


Рисунок 5.1 – Главное меню

При нажатии на кнопку «Купить» осуществляется переход на форму каталог. Каталог представлен на рисунке 5.2



Рисунок 5.2 – Каталог

При нажатии на кнопку «Ножи» будет показана форма ножи. Форма с ножами представлена на рисунке 5.3

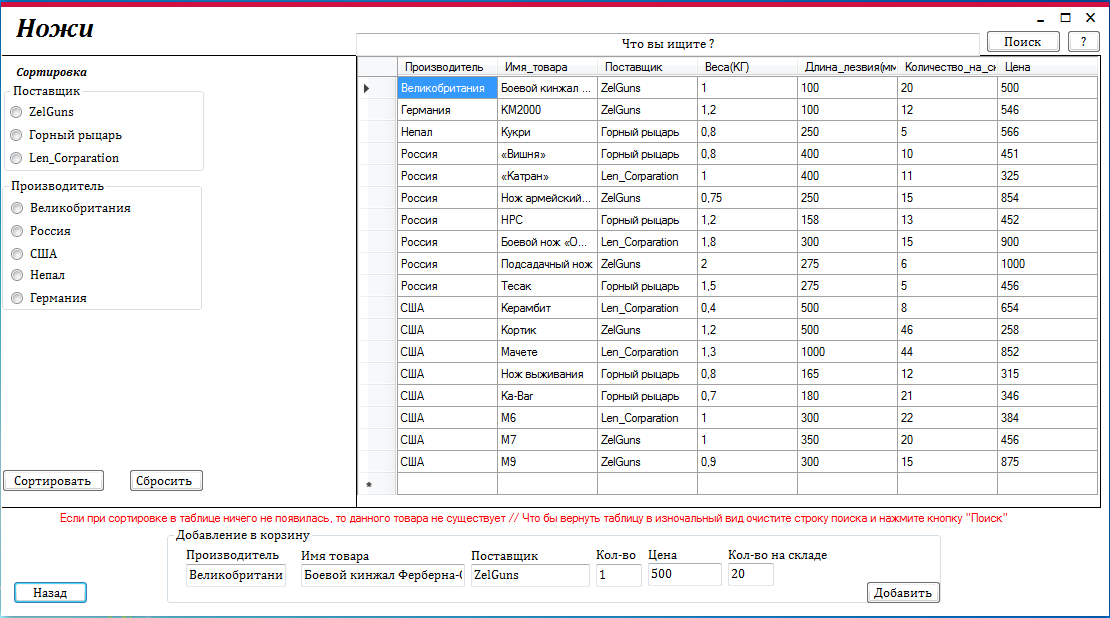


Рисунок 5.3 – Форма ножи

При заполнении поля «Что вы ищите ?» и при нажатии на кнопку «Поиск», осуществляется поиск по базе данных по имени товара. Поиск представлен на рисунке 5.4

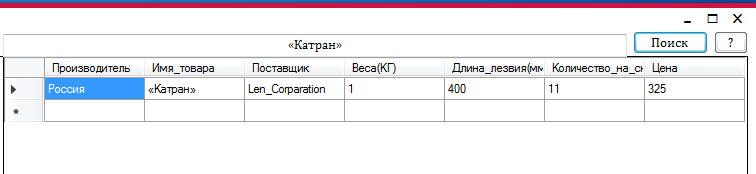


Рисунок 5.4 - Поиск

При нажатии на кнопку «?» будет открыт текстовый файл со списком существующего товара. Список существующего товара представлен на рисунке 5.5

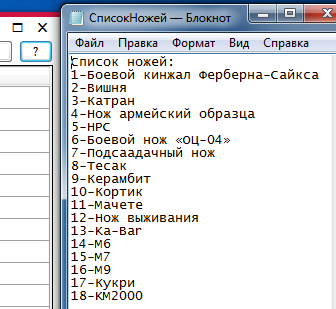


Рисунок 5.5 – Список товаров

При выборе критерия сортировки и при нажатии на кнопку «Сортировать», осуществляется сортировка ножей по базе данных. Сортировка представлена на рисунке 5.6

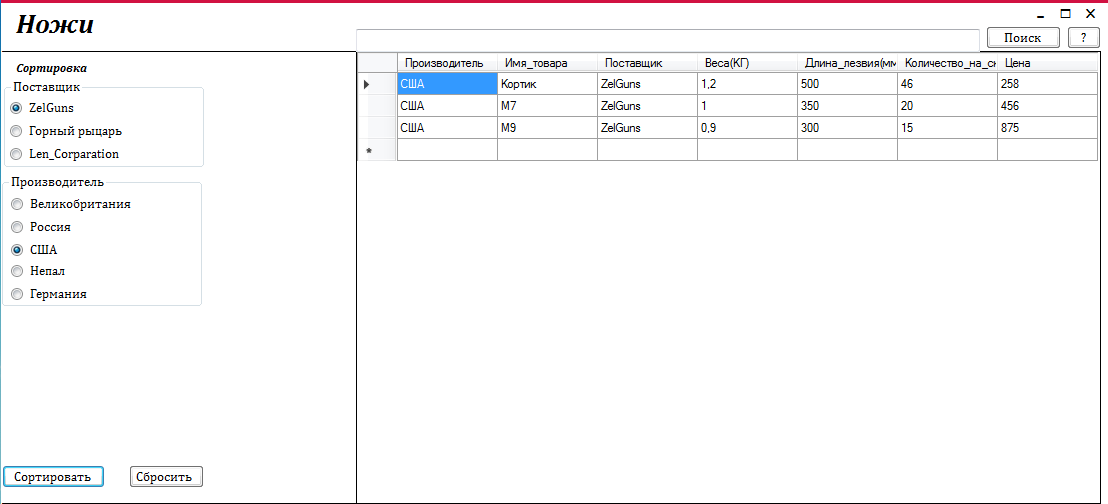


Рисунок 5.6 – Сортировка по критериям

При нажатии на кнопку «Сброс» осуществляется сброс сортировки. Сброс сортировки представлен на рисунке 5.7

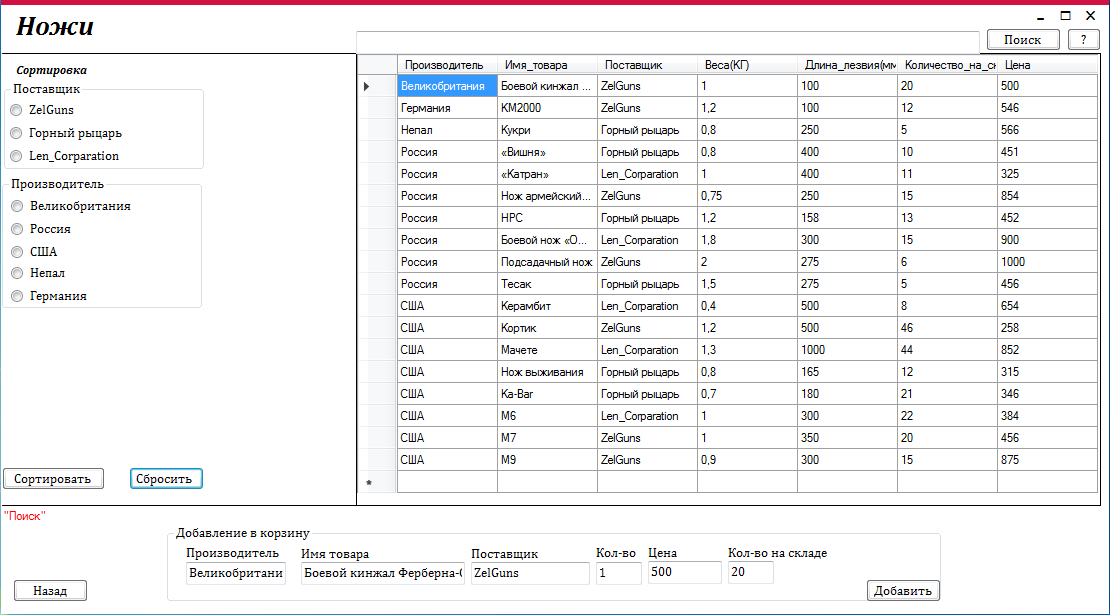


Рисунок 5.7 – Сброс сортировки

При заполнении полей: Производитель, Имя товара, Поставщик, Кол-во, Цена, Кол-во на складе и при нажатии на кнопку «Добавить», осуществляется добавление выбранного товара в корзину. Добавление товара в корзину представлено на рисунке 5.8

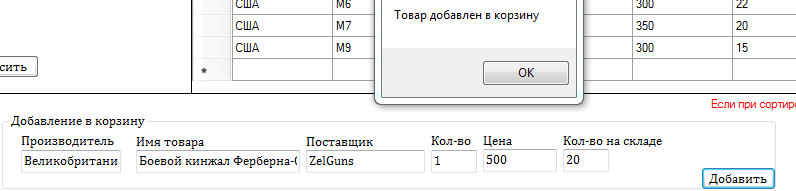


Рисунок 5.8 – Добавление в корзину

При нажатии на кнопку «В корзину» на форме ножи, осуществляется переход на форму корзина. Форма корзина представлена на рисунке 5.9

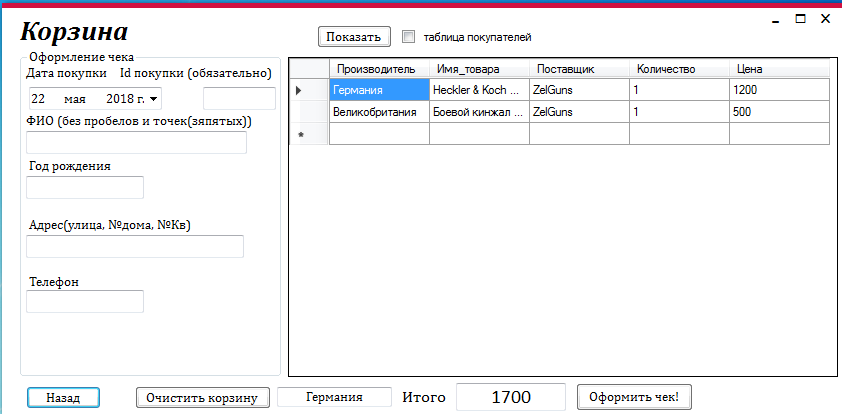


Рисунок 5.9 – Форма корзина

При заполнении полей: Дата покупки, id покупки, ФИО, Год рождения, Адрес, Телефон и при нажатии на кнопку «Оформить чек», осуществляется оформление чека в программе MS Word. Чек представлен на рисунке 5.10

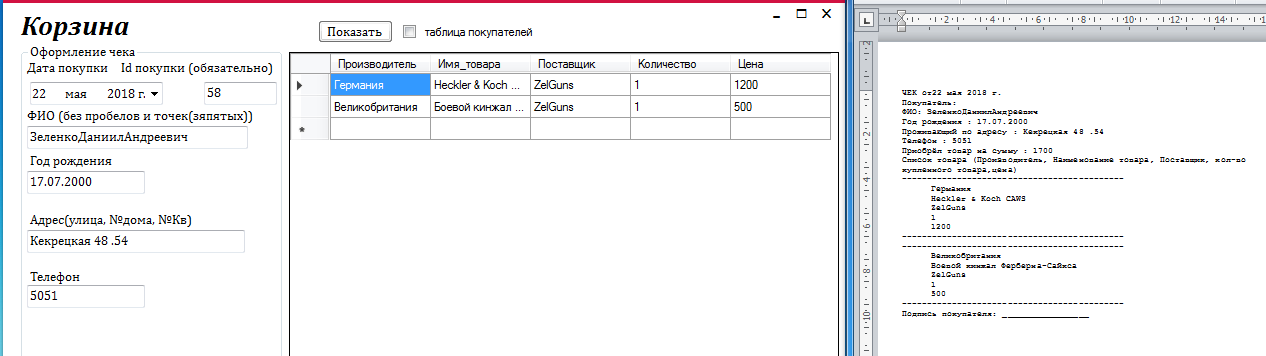


Рисунок 5.10 – Чек

При нажатии на кнопку «Отчёт о продажах», осуществляется оформление отчёта о продажах в программе MS Excel. Отчёт представлен на рисунке 5.11

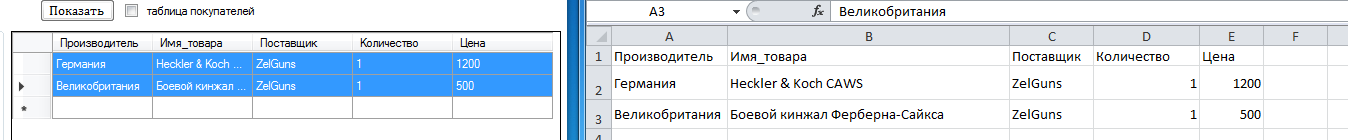


Рисунок 5.11 – Отчёт

При нажатии на кнопку «Очистить корзину», осуществляется удаление выбранного товара из корзины. Удаление товара представлено на рисунке 5.12

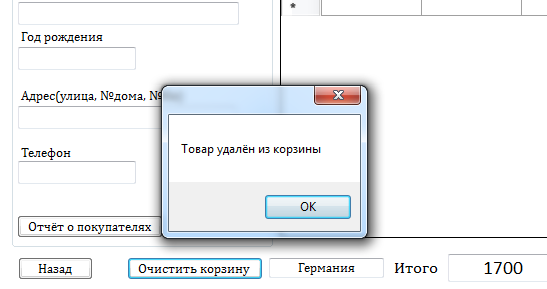


Рисунок 5.12 – Удаление товара

При нажатии на галку «таблица покупателей» и на кнопку «Показать», осуществляется переход на таблицу покупателей. Таблица покупателей представлена на рисунке 5.13

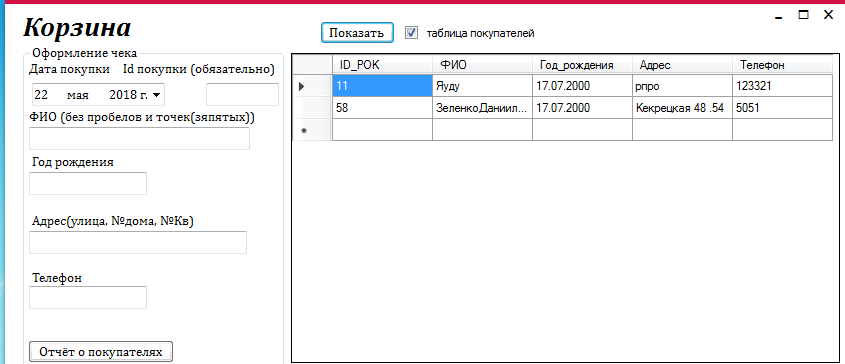


Рисунок 5.13 – Таблица покупателей

При нажатии на кнопку «Отчёт о покупателях», осуществляется оформление отчёта о покупателях в программе MS Excel. Отчёт представлен на рисунке 5.14

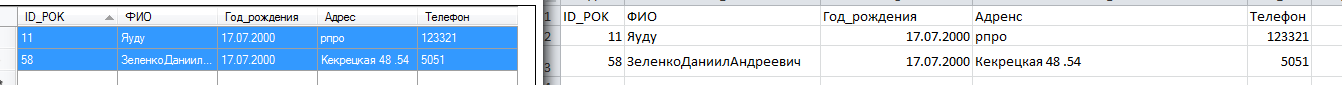


Рисунок 5.14 – Отчёт о покупателях

На формах: Пистолеты, Пистолеты-пулемёты, Штурмовые винтовки, Снайперские винтовки, Дробовики, Гранаты всё аналогично.

# 6 Применение

## 6.1 Назначение программы

Программа создана для продавца.

Продавец может в этой программе осуществлять добавление товара в корзину и осуществлять договор с покупателем. Так же продавец может осуществлять сортировку товара по различным критериям и осуществлять поиск товара по его имени. Продавец может составлять отчёт по проданному товару.

## 6.2 Условия применения

Для применения данного программного средства необходимы следующие технически требования:

* процессор Intel Core 2 Duo или выше;
* минимальный объем оперативной памяти — 400 Мбайт;
* операционная система Windows 7 и выше;
* Framework v4.5 и выше
* устройство для чтения дисков;
* клавиатура;
* мышь.

## 6.3 Справочная система

Справочная система программного средства представляет собой отдельный файл «Spravka.chm» с полным описанием основных функций программы в формате \*.chm. В справочной системе даны ответы на типичные вопросы, возникающие при работе с приложением, что, несомненно, должно помочь при освоении программного средства.

Справка имеет следующие разделы:

- введение;

- об авторе;

- о программе.

Структура справочной системы представлена на рисунке 6.1.

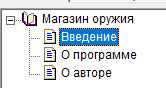


Рисунок 6.1 – Структура справочной системы

Заключение

За время курсового проектирования по созданию и сопровождению программного обеспечения был изучены процессы создания, внедрения и сопровождения программных приложений в различных средах программирования. Во время курсового проектирования удалось накопить неоценимый опыт в разработке приложений в интегрированной среде визуального программирования Windows Forms C#, развить умения находить оптимальный метод решения задачи. Кроме того, изучить принципы организации и технологии реализации программного обеспечения, развить умения разработки программ в соответствии с требованиями технического задания, обеспечивая высокий уровень качества программного обеспечения и экономической эффективности. Получить навыки оформления комплекта документации на созданное программное обеспечение.

При создании программных приложений входящих в обязательную программу практики были отработаны навыки программирования.

# Список используемых источников

1. Орлов С.А. Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. / С. А. Орлов, Б. Я. Цилькер – СПБ. : Питер, 2012.- 608 с.
2. Тепляков С. Паттерны проектирования на платформе .NET. / С. Тепляков – СПБ. : Питер, 2015.-320 с.
3. ГОСТ 19701-90 Схемы алгоритмов, программ, данных. – М. : Из-во стандартов,1990
4. ГОСТ 19301-2000 Программа и методика испытаний. – М. : Из-во стандартов,2000
5. ГОСТ 19401-2000 Текст программы. – М. : Из-во стандартов,2000
6. ГОСТ 19402-2000 Описание программы. – М. : Из-во стандартов,2000
7. ГОСТ 2.105-95 Общие требования к текстовым документам. . – М. : Из-во стандартов,1995
8. Руководство по программированию [Электронный ресурс].- Режим доступа: <https://msdn.microsoft.com/ruru/library/system.data.oledb.oledbconnection(v=vs.110).aspx>
9. Среда разработки Windows 10 [Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/Windows\_10
10. Пакет офисных программ и текстовых редакторов Microsoft Office [Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Word
11. Пакет офисных программ и текстовых редакторов Microsoft Office [Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Excel
12. СУБД [Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Access
13. Среда разработки Microsoft Visual Studio 2017[Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio
14. Язык программирования [Электронный ресурс].-Режим доступа: https://ru.wikipedia.org/wiki/C\_Sharp
15. Среда разработки Help and manual [Электронный ресурс].-Режим доступа: https://en.wikipedia.org/wiki/Help\_%26\_Manual
16. Среда разработки UMLet [Электронный ресурс].-Режим доступа: <https://en.wikipedia.org/wiki/UMLet>
17. Среда разработки Smart install maker [Электронный ресурс].- режим доступа: https://ru.wikipedia.org/wiki/Smart\_Install\_Maker

# Приложение А

(Обязательное)

Текст программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class catalog : MetroForm

{

public catalog(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){StartForm f = new StartForm();f.Show();Close();}

private void pictureBox1\_Click(object sender, EventArgs e)

{metatel m = new metatel();m.Show();Close();}

private void label2\_Click(object sender, EventArgs e){metatel m = new metatel();m.Show();Close();}

private void pictureBox2\_Click(object sender, EventArgs e){ognestrel o = new ognestrel();o.Show();Close();}

private void label3\_Click(object sender, EventArgs e){ognestrel o = new ognestrel();o.Show();Close();}

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

{metatel m = new metatel();m.Show();Close();}

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

{ognestrel p = new ognestrel();p.Show();Close();}

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

{pistols p = new pistols();p.Show();Close();}

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

{micro\_pp p = new micro\_pp();p.Show();Close();}

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

{shtyrm s = new shtyrm();s.Show();Close();}

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

{snapa s = new snapa();s.Show();Close();}

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

{pompa p = new pompa();p.Show();Close();}

private void гранатыToolStripMenuItem\_Click(object sender, EventArgs e){granata g = new granata();g.Show();Close();}

private void корзинаToolStripMenuItem\_Click(object sender, EventArgs e){Korzina l = new Korzina();l.Show();Close();}

private void выходToolStripMenuItem\_Click(object sender, EventArgs e){Application.Exit();}

private void справкаToolStripMenuItem\_Click(object sender, EventArgs e) {System.Diagnostics.Process.Start(@"Spravka.chm");}

private void catalog\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) {System.Diagnostics.Process.Start(@"Spravka.chm");}}

private void прайслистToolStripMenuItem\_Click(object sender, EventArgs e) {System.Diagnostics.Process.Start(@"Прайс\_лист.xlsx");}}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class StartForm : MetroForm

{

public StartForm(){this.KeyPreview = true;InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e){System.Diagnostics.Process.Start(@"Spravka.chm");}

private void button1\_Click(object sender, EventArgs e){catalog f = new catalog();f.Show();Hide();}

private void button3\_Click(object sender, EventArgs e){Application.Exit();}

private void StartForm\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class snapa : MetroForm

{

itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public snapa(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){ognestrel c = new ognestrel();c.Show();Close();}

private void snapa\_Load(object sender, EventArgs e){

// TODO: данная строка кода позволяет загрузить данные в таблицу "оружейкаDataSet.Снайперские\_винтовки". При необходимости она может быть перемещена или удалена.

this.снайперские\_винтовкиTableAdapter.Fill(this.оружейкаDataSet.Снайперские\_винтовки); бегущая\_строка.Start();}

private void бегущая\_строка\_Tick(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; } }//бегущая строка

private void poisk\_Click(object sender, EventArgs e) { снайперскиеВинтовкиBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'"; }

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.shtyrm(); }

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

{

if (ZelGuns.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (австралия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Австралия" + "%'"; }

if (Германия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'"; }

if (Франция.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Франция" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && австралия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Австралия" + "%'"; }

if (ZelGuns.Checked == true && Германия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (ZelGuns.Checked == true && Франция.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && австралия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Австралия" + "%'"; }

if (Горный\_рыцарь.Checked == true && Германия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (Горный\_рыцарь.Checked == true && Франция.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }

// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && австралия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Австралия" + "%'"; }

if (Len\_Corparation.Checked == true && Германия.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (Len\_Corparation.Checked == true && Франция.Checked == true) { снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }}

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

{ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; австралия.Checked = false; Германия.Checked = false; Франция.Checked = false; texst\_poicka.Clear();

снайперскиеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{if (kol\_vo.Text != "0"){f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (f.Kolichestvo <= количество\_на\_складе)

{

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение

MessageBox.Show("Товар добавлен в корзину");

в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }

}else { MessageBox.Show("Вы некорректно заполнили поле(я) "); }}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e)

{

DialogResult DR = MessageBox.Show("Запись будет удалена!\nПродолжить?", "Удаление записи", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

if (DR == DialogResult.Cancel)e.Cancel = true;}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e)

{if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}

if (Char.IsControl(e.KeyChar))

{if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e){ Korzina k = new Korzina(); k.Show(); Close(); }

private void snapa\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class shtyrm : MetroForm

{

itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public shtyrm(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){ognestrel c = new ognestrel();c.Show();Close();}

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

{this.штурмовые\_винтовкиTableAdapter.Fill(this.оружейкаDataSet.Штурмовые\_винтовки); бегущая\_строка.Start();}

private void бегущая\_строка\_Tick(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; } }//бегущая строка

private void poisk\_Click(object sender, EventArgs e) { штурмовыеВинтовкиBindingSource.Filter = "[Имя\_товара]LIKE'"+ texst\_poicka.Text + "%'";}

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.shtyrm(); }

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

{

if (ZelGuns.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (австрия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Австрия" + "%'"; }

if (Германия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && австрия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }

if (ZelGuns.Checked == true && Германия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && австрия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }

if (Горный\_рыцарь.Checked == true && Германия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && австрия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }

if (Len\_Corparation.Checked == true && Германия.Checked == true) { штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }}

private void Сброс\_Click(object sender, EventArgs e){

ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; австрия.Checked = false; Германия.Checked = false; texst\_poicka.Clear();штурмовыеВинтовкиBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

private void добавить\_Click(object sender, EventArgs e){if (kol\_vo.Text != "0")

{ f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (f.Kolichestvo <= количество\_на\_складе)

{con.Open(); // открываем соединение OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение

MessageBox.Show("Товар добавлен в корзину");

в\_корзину.Visible = true;

}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }}else { MessageBox.Show("Вы некорректно заполнили поле(я) "); }}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e){

DialogResult DR = MessageBox.Show("Запись будет удалена!\nПродолжить?", "Удаление записи", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

if (DR == DialogResult.Cancel)e.Cancel = true;}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e){

if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}

if (Char.IsControl(e.KeyChar)){if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e) { Korzina k = new Korzina(); k.Show(); Close(); }

private void shtyrm\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class pompa : MetroForm

{itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public pompa(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){ognestrel v = new ognestrel();v.Show();Close();}

private void pompa\_Load(object sender, EventArgs e) {this.дробовикиTableAdapter.Fill(this.оружейкаDataSet.Дробовики); бегущая\_строка.Start();}

private void бегущая\_строка\_Tick\_1(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; } }//бегущая строка

private void poisk\_Click\_1(object sender, EventArgs e) {дробовикиBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'"; }

private void help\_Click\_1(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.pompa(); }

private void Сортировка\_Click(object sender, EventArgs e){

if (ZelGuns.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { дробовикиBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { дробовикиBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (Германия.Checked == true) { дробовикиBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && Германия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && Германия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && Германия.Checked == true) { дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

}

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

{ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; Германия.Checked = false; texst\_poicka.Clear();

дробовикиBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{if (kol\_vo.Text != "0"){

f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (f.Kolichestvo <= количество\_на\_складе)

{

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение

MessageBox.Show("Товар добавлен в корзину");

в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }}else { MessageBox.Show("Вы некорректно заполнили поле(я) "); }}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e)

{

DialogResult DR = MessageBox.Show("Запись будет удалена!\nПродолжить?", "Удаление записи", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

if (DR == DialogResult.Cancel)e.Cancel = true;}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e)

{if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}if (Char.IsControl(e.KeyChar)){

if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e){ Korzina k = new Korzina(); k.Show(); Close(); }

private void pompa\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class pistols : MetroForm

{

itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public pistols(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e) { ognestrel c = new ognestrel(); c.Show(); Close();}

private void pistols\_Load(object sender, EventArgs e) {this.пистолетыTableAdapter.Fill(this.оружейкаDataSet.Пистолеты); бегущая\_строка.Start();}

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.pistols(); }// экземпляр класса + вызов метода класса

private void poisk\_Click(object sender, EventArgs e) { пистолетыBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'";}

private void бегущая\_строка\_Tick(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; }}//бегущая строка

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

{

if (ZelGuns.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { пистолетыBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { пистолетыBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (австрия.Checked == true) { пистолетыBindingSource.Filter = "[Производитель]LIKE'" + "Австрия" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && австрия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && австрия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }

// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && австрия.Checked == true) { пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Австрия" + "%'"; }}

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

{ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; австрия.Checked = false; texst\_poicka.Clear();

пистолетыBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{if (kol\_vo.Text != "0"){

f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (f.Kolichestvo <= количество\_на\_складе){

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение

MessageBox.Show("Товар добавлен в корзину");в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }}else { MessageBox.Show("Вы некорректно заполнили поле(я) "); } }

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e){

DialogResult DR = MessageBox.Show("Запись будет удалена!\nПродолжить?", "Удаление записи", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

if (DR == DialogResult.Cancel)e.Cancel = true;}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e){if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}

if (Char.IsControl(e.KeyChar)){if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e) { Korzina k = new Korzina(); k.Show(); Close(); }

private void pistols\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class ognestrel : MetroForm

{

public ognestrel(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){pistols p = new pistols();p.Show();Close();}

private void button6\_Click(object sender, EventArgs e){shtyrm s = new shtyrm();s.Show();Close();}

private void button5\_Click(object sender, EventArgs e){granata p = new granata();p.Show();Close();}

private void button4\_Click(object sender, EventArgs e){pompa p = new pompa();p.Show();Close();}

private void button3\_Click(object sender, EventArgs e){snapa s = new snapa();s.Show();Close();}

private void button2\_Click(object sender, EventArgs e){micro\_pp p = new micro\_pp();p.Show();Close();}

private void button7\_Click(object sender, EventArgs e){catalog c = new catalog();c.Show();Close();}

private void ognestrel\_KeyDown(object sender, KeyEventArgs e){ if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class micro\_pp : MetroForm

{

itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public micro\_pp()

{this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){ognestrel c = new ognestrel();c.Show();Close();}

private void micro\_pp\_Load(object sender, EventArgs e)

{this.пистолеты\_пулемётыTableAdapter.Fill(this.оружейкаDataSet.\_Пистолеты\_пулемёты);бегущая\_строка.Start();}

private void бегущая\_строка\_Tick(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; } }//бегущая строка

private void poisk\_Click(object sender, EventArgs e){ пистолетыпулемётыBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'"; }

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.microPP(); }// экземпляр класса + вызов метода класса

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

{

if (ZelGuns.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (италия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Производитель]LIKE'" + "Италия" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && италия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && италия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }

// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && италия.Checked == true) { пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }}

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

{ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; италия.Checked = false; texst\_poicka.Clear();

пистолетыпулемётыBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{if (kol\_vo.Text != "0"){

f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (f.Kolichestvo <= количество\_на\_складе)

{con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();con.Close(); // закрываем соединениеMessageBox.Show("Товар добавлен в корзину");в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }}

else { MessageBox.Show("Вы некорректно заполнили поле(я) "); }}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e)

{if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}if (Char.IsControl(e.KeyChar)){if (e.KeyChar == (char)Keys.Enter)

button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e) { Korzina k = new Korzina(); k.Show(); Close(); }

private void micro\_pp\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Data.OleDb;

using System.Threading.Tasks;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{public partial class metatel : MetroForm

{itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту public metatel(){this.KeyPreview = true;InitializeComponent();}

private void button1\_Click(object sender, EventArgs e){catalog c = new catalog();c.Show();Close();}

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

{ // TODO: данная строка кода позволяет загрузить данные в таблицу "оружейкаDataSet.Метательное\_оружие". При необходимости она может быть перемещена или удалена.

this.метательное\_оружиеTableAdapter.Fill(this.оружейкаDataSet.Метательное\_оружие);

бегущая\_строка.Start();}

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.nozi(); }//экземпляр класса + вызов метода класса

private void poisk\_Click(object sender, EventArgs e) { метательноеОружиеBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'"; }

private void бегущая\_строка\_Tick(object sender, EventArgs e) // бегущая строка

{ if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; }}

private void Сортировка\_Click(object sender, EventArgs e){

if (ZelGuns.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу ПОСТАВЩИК

if (Горный\_рыцарь.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Великобритания.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Великобритания" + "%'"; } // сортировка по столбцу ПРОИЗВОДИТЕЛЬ

if (Россия.Checked== true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }

if (США.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (Непал.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Непал" + "%'"; }

if (Германия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'"; }

/// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Великобритания.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'"+ "ZelGuns"+"'and [Производитель]LIKE'"+"Великобритания" +"%'"; }

if (ZelGuns.Checked == true && Россия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && Непал.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Непал" + "%'"; }

if (ZelGuns.Checked == true && Германия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

/// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Великобритания.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Великобритания" + "%'"; }

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && Непал.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Непал" + "%'"; }

if (Горный\_рыцарь.Checked == true && Германия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

/// сортировка по двум критериям : Поставщик(Len\_Corparation) и производители

if (Len\_Corparation.Checked == true && Великобритания.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Великобритания" + "%'"; }

if (Len\_Corparation.Checked == true && Россия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && Непал.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Непал" + "%'"; }

if (Len\_Corparation.Checked == true && Германия.Checked == true) { метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }}

private void Сброс\_Click(object sender, EventArgs e) // сброс сортировки

{ZelGuns.Checked = false;Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false; texst\_poicka.Clear();

Великобритания.Checked = false; Россия.Checked = false; США.Checked = false; Непал.Checked = false; Германия.Checked = false;метательноеОружиеBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{

f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (kol\_vo.Text != "0"){if (f.Kolichestvo <= количество\_на\_складе){

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение

MessageBox.Show("Товар добавлен в корзину");

в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }

}else { MessageBox.Show("Вы некорректно заполнили поле(я) "); }}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e)

{if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}if (Char.IsControl(e.KeyChar)){

if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e) {Korzina k = new Korzina(); k.Show(); Close();}

private void metatel\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.OleDb;

using System.IO;

using MetroFramework.Components;

using MetroFramework.Forms;

using Word = Microsoft.Office.Interop.Word;

using Excel = Microsoft.Office.Interop.Excel;

using System.Reflection;

namespace Oryzheka

{

public partial class Korzina : MetroForm

{

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public Korzina()

{this.KeyPreview = true;InitializeComponent();stroka\_dataPokypki.Text = DateTime.Now.ToShortDateString();}

private void button1\_Click(object sender, EventArgs e){catalog c = new catalog();c.Show();Close();}

public void ITOGO(){ double balans = 0;

foreach (DataGridViewRow row in dataGridView1.Rows)

{double incom;

double.TryParse((row.Cells[4].Value ?? "0").ToString().Replace(".", ","), out incom);balans += incom;}

stroka\_itogo.Text = balans.ToString();}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "оружейкаDataSet.Покупатель". При необходимости она может быть перемещена или удалена.

this.покупательTableAdapter.Fill(this.оружейкаDataSet.Покупатель);

this.корзинаTableAdapter.Fill(this.оружейкаDataSet.Корзина);ITOGO();}

private void stroka\_FIO\_KeyPress(object sender, KeyPressEventArgs e)

{

if ((e.KeyChar >= 'а' || e.KeyChar >= 'a' || e.KeyChar >= 'А' || e.KeyChar >= 'A') && (e.KeyChar <= 'я' || e.KeyChar <= 'z' || e.KeyChar <= 'Я' || e.KeyChar <= 'Z' ))

{return;}if (Char.IsControl(e.KeyChar)){if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}

e.Handled = true;}private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}if (Char.IsControl(e.KeyChar)){

if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void Очистить\_корзину\_Click(object sender, EventArgs e)

{con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "delete from Корзина where Производитель='" + производятел.Text + "'";

cmd.ExecuteNonQuery();

MessageBox.Show("Товар удалён из корзины");

///

cmd.CommandType = CommandType.Text;cmd.CommandText = "select \* from Корзина ";

cmd.ExecuteNonQuery();DataTable dt = new DataTable();

OleDbDataAdapter da = new OleDbDataAdapter(cmd);

da.Fill(dt);dataGridView1.DataSource = dt;con.Close(); // закрываем соединениеITOGO();}

private void отчёт\_о\_продажах\_Click(object sender, EventArgs e){ SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Excel Documents(\*.xls)|\*.xls";if (sfd.ShowDialog() == DialogResult.OK)

{Export\_Data\_To\_Excel(dataGridView1, sfd.FileName);}}

public void ReplaceStub(string stubToReplace, string text, Word.Document worldDocument)

{var wordApp = new Word.Application();var range = worldDocument.Content;range.Find.ClearFormatting();

object wdReplaceAll = Word.WdReplace.wdReplaceAll;

range.Find.Execute(FindText: stubToReplace, ReplaceWith: text, Replace: wdReplaceAll);}

private void офрмить\_чек\_Click(object sender, EventArgs e){

var dataPokypki = stroka\_dataPokypki.Text;var FIO = stroka\_FIO.Text;var god = stroka\_god\_rozd.Text;

var adres = stroka\_adresa.Text;var telefon = stroka\_mobila.Text;

var itogo = stroka\_itogo.Text;

//

if (stroka\_dataPokypki.Text != ""){if (stroka\_FIO.Text != ""){

if (stroka\_adresa.Text != "")

{if (stroka\_mobila.Text != "")

{StreamWriter sw = new StreamWriter("Чек.doc");

sw.WriteLine("ЧЕК от " + dataPokypki);

sw.WriteLine("Покупатель: ");

sw.WriteLine("ФИО: " + FIO);

sw.WriteLine("Год рождения : " + god);

sw.WriteLine("Проживающий по адресу : " + adres);

sw.WriteLine("Телефон : " + telefon);

sw.WriteLine("Приобрёл товар на сумму : " + itogo);

sw.WriteLine("Список товара (Производитель, Наименование товара, Поставщик, кол-во купленного товара,цена) ");

for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

sw.WriteLine("--------------------------------------------");

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

sw.Write("\t" + dataGridView1.Rows[i].Cells[j].Value.ToString() + "\n");

}

sw.WriteLine("--------------------------------------------");

}

sw.WriteLine("\n");

sw.WriteLine("Подпись покупателя: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

sw.Close();

MessageBox.Show("Чек создан, он откроется после закрытия этого окна");

System.Diagnostics.Process.Start(@"Чек.doc");

отчёт\_о\_продажах.Visible = true;

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Покупатель values('" + KOd.Text +"','"+ FIO + "','" + god + "','" + adres + "','" + telefon + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединение}else { MessageBox.Show("Вы не заполнили поле(я)"); }}

else { MessageBox.Show("Вы не заполнили поле(я)"); }}else { MessageBox.Show("Вы не заполнили поле(я)"); }

}else { MessageBox.Show("Вы не заполнили поле(я)"); }}

public void Export\_Data\_To\_Excel(DataGridView gridviewID, string excelFilename)

{// Скопировать результаты DataGridView в буфер обмена

copyAlltoClipboard(gridviewID);

object misValue = System.Reflection.Missing.Value;

Excel.Application xlexcel = new Excel.Application();

xlexcel.DisplayAlerts = false; // что бы не было два подтверждения перезаписи

Excel.Workbook xlWorkBook = xlexcel.Workbooks.Add(misValue);

Excel.Worksheet xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get\_Item(1);

Excel.Range rng = xlWorkSheet.get\_Range("D:D").Cells;rng.NumberFormat = "@";

// Вставка результатов буфера обмена в диапазон рабочих листов

Excel.Range CR = (Excel.Range)xlWorkSheet.Cells[1, 1];

CR.Select();CR.EntireColumn.AutoFit();

xlWorkSheet.PasteSpecial(CR, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, true);

// Удаление пустого столбца A и выберите ячейку A1

Excel.Range delRng = xlWorkSheet.get\_Range("A:A").Cells;

delRng.EntireColumn.AutoFit();delRng.Delete(Type.Missing);xlWorkSheet.get\_Range("A1").Select();

// Сохраним файл excel под захваченным местоположением из SaveFileDialog

xlWorkBook.SaveAs(excelFilename, Excel.XlFileFormat.xlWorkbookNormal, misValue, misValue, misValue, misValue, Excel.XlSaveAsAccessMode.xlExclusive, misValue, misValue, misValue, misValue, misValue);

xlexcel.DisplayAlerts = true;

xlWorkBook.Close(true, misValue, misValue);xlexcel.Quit();releaseObject(xlWorkSheet);

releaseObject(xlWorkBook);releaseObject(xlexcel);

// Очистка выбора буфера обмена и DataGridViewClipboard.Clear();gridviewID.ClearSelection();

// Открытие вновь сохраненного файла excel

if (File.Exists(excelFilename))System.Diagnostics.Process.Start(excelFilename);}

private void copyAlltoClipboard(DataGridView gridviewID){gridviewID.SelectAll();

DataObject dataObj = gridviewID.GetClipboardContent();if (dataObj != null)Clipboard.SetDataObject(dataObj);}

private void releaseObject(object obj){try{System.Runtime.InteropServices.Marshal.ReleaseComObject(obj);obj = null;}

catch (Exception ex){obj = null;MessageBox.Show("Exception Occurred while releasing object " + ex.ToString());}

finally{GC.Collect();}}

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

{if (таблица\_покупателей.Checked == true) { dataGridView2.Visible = true; dataGridView1.Visible = false; report.Visible = true; } else { dataGridView2.Visible = false; dataGridView1.Visible = true; report.Visible = false; }}

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

{SaveFileDialog sfd = new SaveFileDialog();sfd.Filter = "Excel Documents(\*.xls)|\*.xls";

if (sfd.ShowDialog() == DialogResult.OK)

{Export\_Data\_To\_Excel(dataGridView2, sfd.FileName);}}

private void Korzina\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Oryzheka

{

class itog

{

private double kolichestvo, tena;public double itogo;

public itog(double kolichestvo, double tena)

{this.kolichestvo = kolichestvo;this.tena = tena; }

public double itogi(){return kolichestvo \* tena;}

public double Kolichestvo

{get { return this.kolichestvo; }set{if (value > 0) { this.kolichestvo = value; }else { this.kolichestvo = 0; }}}

public double Tena{get { return this.tena; }set{if (value > 0) { this.tena = value; }else { this.tena = 0; }}}}}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Windows.Forms;

using MetroFramework.Components;

using MetroFramework.Forms;

namespace Oryzheka

{

public partial class granata : MetroForm

{

itog f = new itog(1, 2);

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;Data Source =Оружейка.mdb");// подключаем бд к проекту

public granata() { InitializeComponent(); this.KeyPreview = true; }

private void button1\_Click(object sender, EventArgs e) { ognestrel k = new ognestrel(); k.Show(); Close(); }

private void granata\_Load(object sender, EventArgs e) {this.гранатаTableAdapter.Fill(this.оружейкаDataSet.Граната); бегущая\_строка.Start(); }

private void бегущая\_строка\_Tick(object sender, EventArgs e) { if (label5.Left < this.Width) { label5.Left += 1; } else { label5.Left = -label5.Width; } }//бегущая строка

private void poisk\_Click(object sender, EventArgs e){ гранатаBindingSource.Filter = "[Имя\_товара]LIKE'" + texst\_poicka.Text + "%'"; }

private void help\_Click(object sender, EventArgs e) { SpisokGuns g = new SpisokGuns(); g.grema(); }

private void Сортировка\_Click(object sender, EventArgs e){

if (ZelGuns.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "%'"; } // сортировка по столбцу поставщик

if (Горный\_рыцарь.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "%'"; }

if (Len\_Corparation.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "%'"; }

if (Россия.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Россия" + "%'"; }// сортировка по производителю

if (США.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "США" + "%'"; }

if (Аргентина.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Аргентина" + "%'"; }

if (Великобритания.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Великобритания" + "%'"; }

if (Италия.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Италия" + "%'"; }

if (Франция.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Франция" + "%'"; }

if (Германия.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Германия" + "%'";}

if (Япония.Checked == true) { гранатаBindingSource.Filter = "[Производитель]LIKE'" + "Япония" + "%'"; }

// сортировка по двум критериям : Поставщик(ZelGuns) и производители

if (ZelGuns.Checked == true && Россия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (ZelGuns.Checked == true && США.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (ZelGuns.Checked == true && Аргентина.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Аргентина" + "%'"; }

if (ZelGuns.Checked == true && Великобритания.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Великобритания" + "%'"; }

if (ZelGuns.Checked == true && Италия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }

if (ZelGuns.Checked == true && Франция.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }

if (ZelGuns.Checked == true && Германия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (ZelGuns.Checked == true && Япония.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "ZelGuns" + "'and [Производитель]LIKE'" + "Япония" + "%'"; }

// сортировка по двум критериям : Поставщик(Горный рыцарь) и производители

if (Горный\_рыцарь.Checked == true && Россия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Горный\_рыцарь.Checked == true && США.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Горный\_рыцарь.Checked == true && Аргентина.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Аргентина" + "%'"; }

if (Горный\_рыцарь.Checked == true && Великобритания.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Великобритания" + "%'"; }

if (Горный\_рыцарь.Checked == true && Италия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }

if (Горный\_рыцарь.Checked == true && Франция.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }

if (Горный\_рыцарь.Checked == true && Германия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (Горный\_рыцарь.Checked == true && Япония.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Горный рыцарь" + "'and [Производитель]LIKE'" + "Япония" + "%'"; }

// сортировка по двум критериям : Поставщик(лен корп) и производители

if (Len\_Corparation.Checked == true && Россия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Россия" + "%'"; }

if (Len\_Corparation.Checked == true && США.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "США" + "%'"; }

if (Len\_Corparation.Checked == true && Аргентина.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Аргентина" + "%'"; }

if (Len\_Corparation.Checked == true && Великобритания.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Великобритания" + "%'"; }

if (Len\_Corparation.Checked == true && Италия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Италия" + "%'"; }

if (Len\_Corparation.Checked == true && Франция.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Франция" + "%'"; }

if (Len\_Corparation.Checked == true && Германия.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Германия" + "%'"; }

if (Len\_Corparation.Checked == true && Япония.Checked == true) { гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "Len\_Corparation" + "'and [Производитель]LIKE'" + "Япония" + "%'"; }}

private void Сброс\_Click(object sender, EventArgs e){

ZelGuns.Checked = false; Горный\_рыцарь.Checked = false; Len\_Corparation.Checked = false;

Россия.Checked = false; США.Checked = false; Германия.Checked = false; Италия.Checked = false; Франция.Checked = false; Япония.Checked = false; Аргентина.Checked = false; Великобритания.Checked = false; texst\_poicka.Clear();

гранатаBindingSource.Filter = "[Поставщик]LIKE'" + "" + "%'";}

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

{f.Kolichestvo = Convert.ToDouble(kol\_vo.Text); // расчётная часть для цены

f.Tena = Convert.ToDouble(tena.Text);

double количество\_на\_складе = Convert.ToDouble(kol\_vo\_na\_sklade.Text);

if (kol\_vo.Text != "0"){if (f.Kolichestvo <= количество\_на\_складе){

con.Open(); // открываем соединение

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into Корзина values('" + proizvoditel.Text + "','" + nameTovara.Text + "','" + postavshik.Text + "','" + kol\_vo.Text + "','" + f.itogi() + "','" + количество\_на\_складе + "')"; // записываем данные

cmd.ExecuteNonQuery();

con.Close(); // закрываем соединениеMessageBox.Show("Товар добавлен в корзину");в\_корзину.Visible = true;}

else { MessageBox.Show("Извините но на складе только есть " + количество\_на\_складе + "ед. товара"); }

} else { MessageBox.Show("Вы некорректно заполнили поле(я) ");}}

private void dataGridView1\_UserDeletingRow(object sender, DataGridViewRowCancelEventArgs e){

DialogResult DR = MessageBox.Show("Запись будет удалена!\nПродолжить?", "Удаление записи", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button2);

if (DR == DialogResult.Cancel)e.Cancel = true;}

private void kol\_vo\_KeyPress(object sender, KeyPressEventArgs e){if (e.KeyChar >= '0' && e.KeyChar <= '9'){return;}

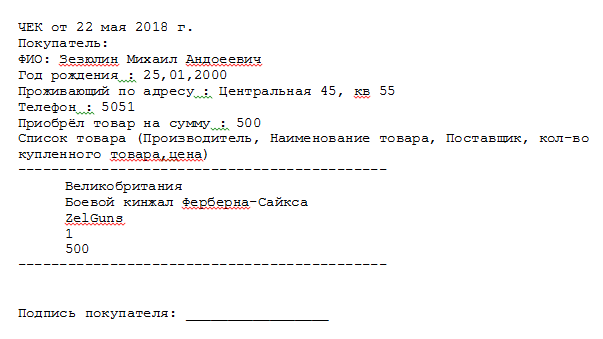
if (Char.IsControl(e.KeyChar)){if (e.KeyChar == (char)Keys.Enter)button1.Focus();return;}e.Handled = true;}

private void в\_корзину\_Click(object sender, EventArgs e) { Korzina k = new Korzina(); k.Show(); Close();}

private void granata\_KeyDown(object sender, KeyEventArgs e) { if (e.KeyCode == Keys.F1) { System.Diagnostics.Process.Start(@"Spravka.chm"); } }}}

# Приложение Б

(Выходные данные)



Б.1 - Чек