Министерство энергетики, промышленности и связи ставропольского края

Государственное бюджетное профессиональное образовательное учреждение «Ставропольский колледж связи имени Героя Советского Союза В.А. Петрова»

Специальность: 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
|  | Допустить к защите |
|  | приказом № \_\_от «\_\_» \_\_\_ 2022 г. |
|  | Заместитель директора по учебно-воспитательной работе |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.В. Анищенко |

ДИПЛОМНАЯ РАБОТА

на тему:

Разработка приложения для учета товаров для ИП Шитиков В.В. на языке программирования С#

Дипломник: Буржумов Замир Гаджибекович

Курс 4 группа ИП191

Руководитель ВКР*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/* С.В. Артемов

Ставрополь, 2022

Министерство энергетики, промышленности и связи ставропольского края

Государственное бюджетное профессиональное образовательное учреждение «Ставропольский колледж связи имени Героя Советского Союза В.А. Петрова»

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДАЮ |
|  |  | Председатель ЦК ВТ  \_\_\_\_\_\_\_\_\_\_\_\_\_/И.В. Ерёмина  «01» апреля 2022 г. |

ЗАДАНИЕ

на выпускную квалификационную работу

Студенту 4 курса, ИП191 группы, специальности 09.02.07 «Информационные системы и программирование»

Буржумов Замир Гаджибекович

Тема работы: Разработка приложения для учета товаров для ИП Шитиков В.В. на языке программирования С#

Утверждена приказом по ГБПОУ СКС № 153-удот «20» декабря 2021 г.

Перечень подлежащих разработке в ВКР вопросов или краткое содержание ВКР:

1. Обзор литературных источников;

2. Анализ средств разработки;

3. Выбор средства разработки;

4. Разработка дизайна и создание продукта;

5. Тестирование программного продукта.

Рассмотрено на заседании ЦК «Вычислительная техника»

Протокол № 7 от «17» апреля 2022 г.

Срок сдачи законченной работы «15» июня 2022 г.

Руководитель \_\_\_\_\_\_\_\_\_\_ / С.В. Артемов

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «01» апреля 2022 г.

(*подпись обучающегося)*

КАЛЕНДАРНЫЙ ПЛАН

Работы над дипломной работой студента Буржумов Замир Гаджибекович

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование этапов выполнения выпускной квалификационной работы | Сроки | Примечание |
|  | Утверждение тематики ВКР | 17.12.2021 |  |
|  | Закрепление тем ВКР, назначение руководителя | 20.12.2021 |  |
|  | Подбор и анализ исходной информации, подготовка и утверждение плана ВКР | 17.01.2022-04.02.2022 |  |
|  | Выполнение теоретической части ВКР | 07.02.2022-31.03.2022 |  |
|  | Выдача задания на ВКР и преддипломную практику | 01.04.2022 |  |
|  | Сбор информации и систематизация материала во время прохождения преддипломной практики | 18.04.2022-06.05.2022 |  |
|  | Оформление практической части ВКР | 11.05.2022-19.05.2022 |  |
|  | Оформление приложений | 20.05.2022 |  |
|  | Согласование содержания ВКР, устранение замечаний руководителя | 23.05.2022-25.05.2022 |  |
|  | Выполнение презентационной части | 26.05.2022-27.05.2022 |  |
|  | Проверка содержания и утверждение ВКР руководителем | 02.06.2022-03.06.2022 |  |
|  | Получение отзыва руководителя ВКР | 06.06.2022 |  |
|  | Рецензирование ВКР | 07.06.2022-08.06.2022 |  |
|  | Предварительная защита | 09.06.2022 |  |
|  | Предоставление ВКР в учебную часть на получение допуска к защите | 15.06.2022 |  |
|  | Демонстрационный экзамен | 02.06.2022-03.06.2022 |  |
|  | Защита ВКР | 21.06.2022 |  |

Студент / З. Г. Буржумов

Руководитель работы \_\_\_\_\_\_\_\_\_ / С. В. Артемов

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc105578347)

[1 ТЕХНИКО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc105578348)

[1.1 Общее представление об автоматизированных рабочих местах 7](#_Toc105578349)

[2 СТРУКТУРА ОБЪЕКТА АВТОМАТИЗАЦИИ 9](#_Toc105578350)

[2.1 Формализация функций для учета товаров 9](#_Toc105578351)

[2.2 Разработка технической документации на этапе проектирования объекта автоматизации 10](#_Toc105578352)

[2.3 Функциональная модель автоматизированного рабочего места 11](#_Toc105578353)

[2.4 Объектно-ориентированная модель объекта автоматизации 15](#_Toc105578354)

[2.5 Общая характеристика организации решения задачи на ПК 23](#_Toc105578355)

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА 24](#_Toc105578356)

[3.1. Выбор средства разработки 24](#_Toc105578357)

[3.2 Обзор аналогов разрабатываемого программного обеспечения 26](#_Toc105578358)

[3.3. Разработка базы данных 27](#_Toc105578359)

[4.1 Оценка эффективности внедряемого программного продукта 37](#_Toc105578360)

[Список информационных источников 39](#_Toc105578361)

[Приложение А 41](#_Toc105578362)

[Приложение Б 42](#_Toc105578363)

[Приложение В 43](#_Toc105578364)

# ВВЕДЕНИЕ

Научно-технический прогресс привел к тому, что практически каждый использует электронные вычислительные машины в своей деятельности, будь то обычный стационарный компьютер, ноутбук или планшет.

Кроме того, в настоящее время в России большое внимание на государственном уровне уделяется информированию общества в целом.

Постоянное увеличение объемов и интенсивности информационных потоков приводит к необходимости использования информационные средства и технологии для повышения эффективности обработки, надежности и удобства представления. Но для полного перехода на традиционный бумажный документооборот преподавательского состава необходимо предусмотреть инструментарий для отображения в электронном виде как минимум тех же функций, однако часть функций при этом необходимо автоматизировать.

Целью квалификационной работы является разработка приложения для учета товаров для ИП Шитиков В.В на языке программирования C#.

Для достижения целей и задач необходимо решить ряд задач:

* Анализ требований к разработке программного продукта.
* Выбор среды разработки – исследование существующих сред разработки и выбор наиболее подходящей для выбранного языка программирования и простой в использовании,
* Проектирование информационной системы – проектирование экранных форм, объектов данных и их структуры, определение способов взаимодействия пользователя с программой,
* Разработка схемы базы данных – описание таблиц для хранения данных о каждой сущности в базе данных, определение связей между таблицами и генерация сценария для подготовки схемы данных к использованию на сервере,
* Реализация информационной системы – преобразование результатов проектирования в готовый программный продукт с использованием выбранного стека технологий,
* Тестирование и отладка информационной системы – установление программных ошибок, их локализация и последующее устранение.

# 1 ТЕХНИКО-ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ПРЕДМЕТНОЙ ОБЛАСТИ

# Общее представление об автоматизированных рабочих местах

Автоматизированное рабочее место (АРМ) комплекс среди вычислительной техники и программного обеспечения, размещающийся, напрямую на рабочем месте работника и предназначенный для компьютеризации его работы в рамках квалификации. Автоматизированные рабочие места должны формироваться строго в соответствии с их рассчитываемым функциональным назначением. В прочем общие законы формирования автоматизированного рабочего места остаются незыблемыми, к ним относят:

* Системность
* Гибкость
* Устойчивость
* Эффективность

Автоматизация — это широкое понятие, которое применимо к современным рабочим местам. Практически в любом бизнесе и во всех отраслях есть возможности для большей автоматизации.

Автоматизация не заменит людей, но освободит их от выполнения рутинных и повторяющихся задач.

Существует распространенное заблуждение, что автоматизация предполагает высокую робототехнику, но это может быть так же просто, как набор инструментов, размещенных в обычных бизнес-программах. По своей сути автоматизация заключается во внедрении системы для выполнения повторяющихся и легко воспроизводимых задач без использования человеческого труда.

Автоматизируя повторяющиеся бизнес-процессы, сотрудники освобождаются для выполнения задач, которые являются более ценными, чем те, которые могут быть выполнены машинами.

# 2 СТРУКТУРА ОБЪЕКТА АВТОМАТИЗАЦИИ

# 2.1 Формализация функций для учета товаров.

Торговля является огромной отраслью экономики России. В эту сферу вовлечено практически все население страны либо в качестве продавцов, либо в качестве покупателей. В условиях рыночных отношений торговля занимает все более значительное место в структуре общественного производства.

Основными хозяйственными процессами в торговле являются приобретение товаров, доставка, хранение товаров и их дальнейшая реализация.  
 С развитием компьютерных технологий и программного обеспечения, появилась необходимость создания программного продукта, способного повысить производительность труда, при этом сократив усилия и затраты человека. Сегодня любое предприятие, в том числе и самое малое, заинтересованно во внедрении подобного программного обеспечения как неотъемлемой части своей деятельности.

Компьютерный учет имеет свои особенности и радикально отличается от обычного. Компьютер не только облегчает учет, сокращая время, требующееся на оформление документов и обобщение накопленных данных для анализа хода торговой деятельности, необходимого для управления ею.

Отчеты о положении в торговле, получаемые с помощью компьютера, можно получить и без него – никакой особой математики в компьютере не содержится – но на расчеты уйдет столько времени, что они уже ни на что не будут нужны; или ими придется занять такое количество расчетчиков, что на их зарплату уйдет значительно больше, чем будет получено прибыли в результате их расчетов.

Таким образом при применении компьютера “количество переходит в качество”: увеличение скорости расчетов делает возможным качественное улучшение самой схемы построения торговли. Программное обеспечение для работы с базами данных используется на персональных компьютерах уже довольно давно. К сожалению, эти программы либо были элементарными диспетчерами хранения данных и не имели средств разработки приложений, либо были настолько сложны и трудны, что даже хорошо разбирающиеся в компьютерах люди избегали работать с ними до тех пор, пока не получали полных, ориентированных на пользователя приложений.

# 2.2 Разработка технической документации на этапе проектирования объекта автоматизации

Техническое задание — документ или несколько документов, определяющих цель, структуру, свойства и методы какого-либо проекта, и исключающие двусмысленное толкование различными исполнителями.

Техническое задание используется для того, чтобы поставщики, покупатели и пользователи материалов, продуктов или услуг понимали и согласовывали все требования.

Техническое задание — это письменное соглашение между вами и подрядчиком, взаимное видение того, как должен выглядеть ваш веб-сайт или приложение. Это документ, в котором прописаны обязанности подрядчика, имеется возможность вносить изменения в техническое задание. Техническое задание представлено в Приложении А.

# 2.3 Функциональная модель автоматизированного рабочего места

Контекстная диаграмма, также называемая диаграммой потока данных уровня 0, является самым высоким уровнем в Диаграмме потока данных. Это инструмент, популярный среди бизнес-аналитиков, которые используют его для понимания деталей и границ системы, которая будет разработана в проекте. Он указывает на поток информации между системой и внешними компонентами.

Контекстная диаграмма является частью документа требований в проекте. В отличие от других проектных диаграмм, Контекстная диаграмма предназначена не для использования инженерами/техниками, а для заинтересованных сторон проекта. Поэтому он должен быть изложен простым и понятным языком, чтобы заинтересованные стороны могли легко понять пункты при его анализе.

Диаграмма контекста системы, имеющая большое значение для заинтересованных сторон проекта, привлекает все внимание к внешним факторам и событиям, которые необходимо учитывать при разработке всего набора ограничений и требований системы.

Диаграмма контекста системы представляет все внешние компоненты, которые могут взаимодействовать с системой, следовательно, отображает всю программную систему как единое целое. Этот тип диаграммы помещает систему в середину и окружает ее внешними частями, связанными объектами и окружением. Детали внутренней структуры системы явно опущены.

Кроме того, контекстная диаграмма (Рисунок 1) может быть использована для значительного снижения рисков в проекте.

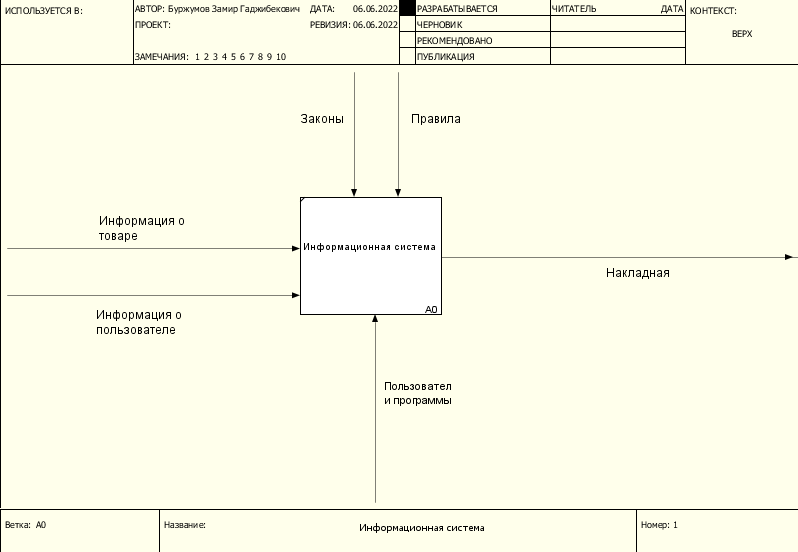


Рисунок 1 — Контекстная диаграмма информационной системы

Диаграмма декомпозиции процессов (часто называемая декомпозицией) объясняет разбивку процессов в рамках проекта, бизнес-области или функциональной области. Цель состоит в том, чтобы показать все процессы и определить взаимосвязи и зависимости между ними. Обратите внимание, что декомпозиция не углубляется в "как"; она просто описывает "что".

Процессы, под которыми находятся другие процессы, называются родительскими процессами. Процессы, которые подчиняются другому процессу, называются дочерними процессами.

Разбивается родительский процесс, по крайней мере на два дочерних процесса; в противном случае это не настоящий родительский процесс.

Все дочерние процессы вместе должны полностью описывать все действия в родительском процессе.

Диаграмма не следует какой-либо определенной последовательности.

В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 — это не элемент управления нижестоящими работами. Работы нижнего уровня — это то же самое, что и работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня — это то же самое, что и границы диаграммы декомпозиции (Рисунок 2).

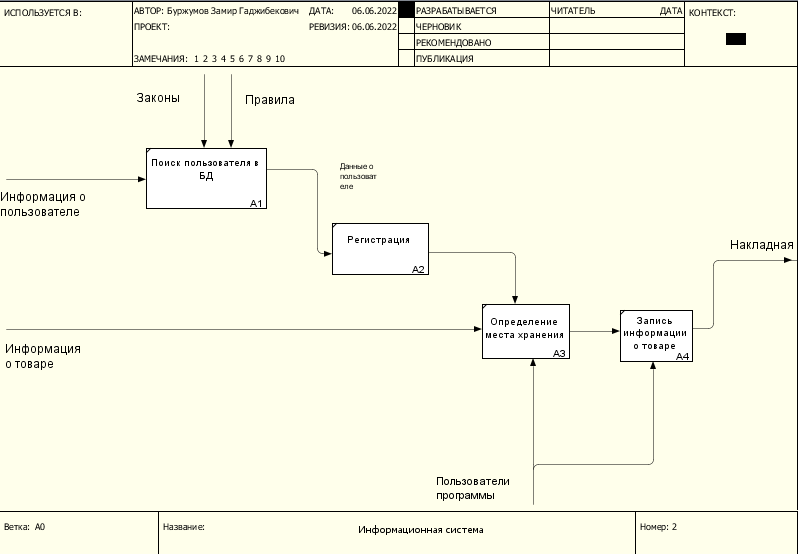


Рисунок 2 — Диаграмма декомпозиции первого уровня

Диаграмма декомпозиции предназначена для детализации работы. В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 — это не элемент управления нижестоящими работами. Работы нижнего уровня — это то же самое, что и работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня — это то же самое, что и границы диаграммы декомпозиции.

При декомпозиции функции входящие в нее и исходящие из нее дуги автоматически появляются на диаграмме декомпозиции (Рисунок 3), но при этом не касаются блоков.

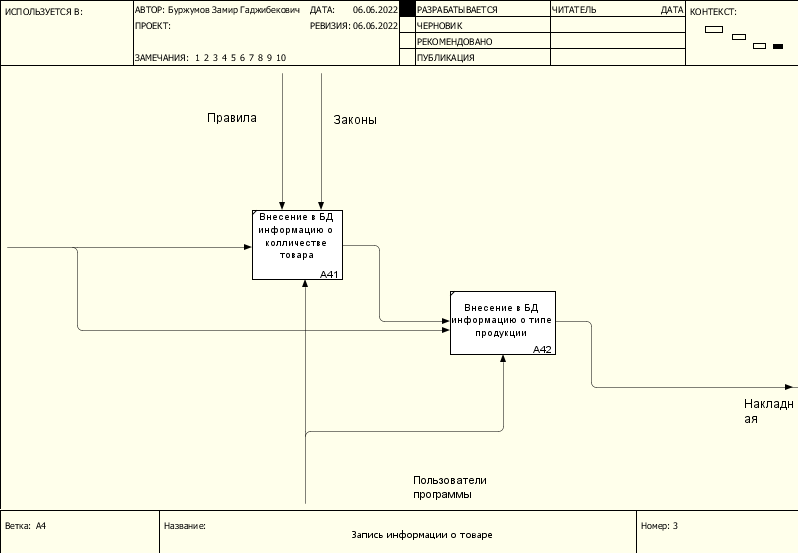


Рисунок 3 — Диаграмма декомпозиции второго уровня

# 2.4 Объектно-ориентированная модель объекта автоматизации

Диаграмма прецедентов

Диаграмма (Рисунок 4), отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

Диаграмма вариантов использования UML — это основная форма требований к системе или программному обеспечению для новой недостаточно разработанной программы. Варианты использования определяют ожидаемое поведение, а не точный метод его реализации.

Варианты использования, однажды заданные, могут быть обозначены как текстовым, так и визуальным представлением (т.е. диаграммой вариантов использования). Ключевой концепцией моделирования вариантов использования является то, что оно помогает нам проектировать систему с точки зрения конечного пользователя. Это эффективный метод передачи информации о поведении системы в терминах пользователя путем указания всего видимого извне поведения системы.

В схеме вариантов использования не отображаются подробные сведения о вариантах использования:

* В нем суммируются лишь некоторые взаимосвязи между вариантами использования, субъектами и системами.
* Он не показывает порядок, в котором выполняются шаги для достижения целей каждого варианта использования.

Диаграмма вариантов использования должна быть простой и содержать всего несколько фигур.

Диаграммы вариантов использования обычно разрабатываются на ранней стадии разработки, и люди часто применяют моделирование вариантов использования для следующих целей:

* Указать контекст системы
* Фиксировать требования системы
* Проверка архитектуры системы
* Управление внедрением
* Создание тестовых примеров

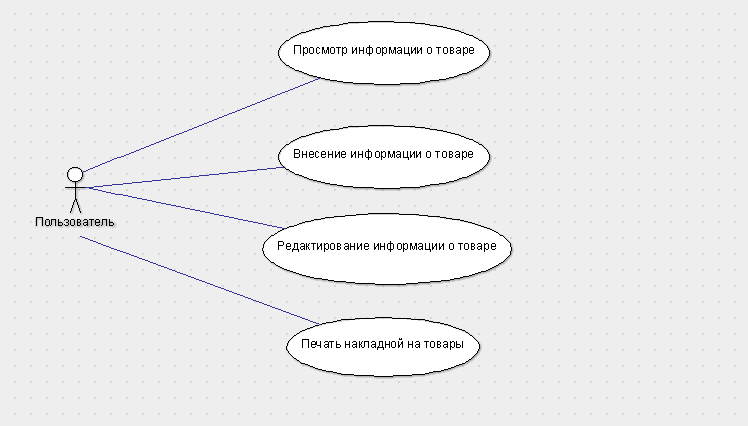


Рисунок 4 — Диаграмма прецедентов

Диаграмма кооперации

Диаграмма кооперации (Рисунок 5) отображает поток событий через конкретный сценарий варианта использования, упорядочена по времени, а кооперативные диаграммы больше внимания заостряют на связях между объектами.

Оно определяет набор участников и отношений, которые имеют значение для заданного набора целей:

* Сотрудничество между объектами, работающими вместе, обеспечивает новые желаемые функциональные возможности в объектно-ориентированных системах.
* Каждый объект частично поддерживает возникающие функциональные возможности.
* Объекты способны создавать (пригодные для использования) высокоуровневые функциональные возможности, работая вместе.
* Объекты сотрудничают, обмениваясь данными (передавая сообщения) друг с другом, чтобы работать вместе

В отличие от диаграммы последовательности, диаграмма кооперации показывает взаимосвязи между объектами. Диаграммы последовательности и диаграммы кооперации отображают схожую информацию, но показывают ее по-разному.

Из-за формата диаграммы кооперации они, как правило, лучше подходят для аналитических действий. В частности, они, как правило, лучше подходят для изображения более простых взаимодействий меньшего числа объектов. Однако, если количество объектов и сообщений растет, диаграмму становится все труднее читать. Кроме того, трудно отобразить дополнительную описательную информацию, такую как время, точки принятия решений или другую неструктурированную информацию, которую можно легко добавить к примечаниям на диаграмме последовательности.

Объекты в диаграммах кооперации можно использовать следующими способами:

* Каждый объект в совместной работе имеет имя и указан его класс
* Не все классы должны отображаться
* В классе может быть более одного объекта
* Класс объекта может быть не указан. Обычно вы сначала создаете диаграмму совместной работы с объектами, а затем указываете их классы.
* Объекты могут быть безымянными, но вы должны дать им имена, если хотите различать разные объекты одного и того же класса.

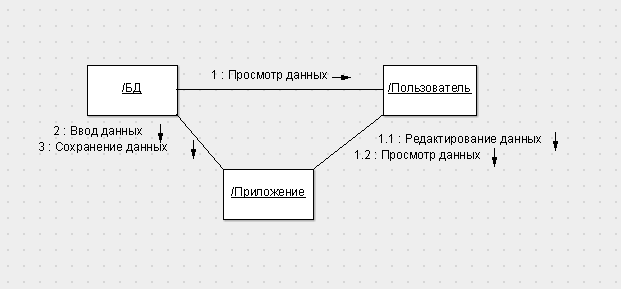


Рисунок 5 — Диаграмма кооперации

Диаграмма деятельности

Диаграмма деятельности (Рисунок 6) представляют собой графическое представление рабочих процессов поэтапных действий и действий с поддержкой выбора, итерации и параллелизма. Они описывают поток управления целевой системой, такой как исследование сложных бизнес - правил и операций, а также описание прецедентов и бизнес-процессов.

Объекты в диаграммах деятельности можно использовать следующими способами:

* Каждый объект в совместной работе имеет имя и указан его класс
* Не все классы должны отображаться
* В классе может быть более одного объекта
* Класс объекта может быть не указан. Обычно вы сначала создаете диаграмму совместной работы с объектами, а затем указываете их классы.
* Объекты могут быть безымянными, но вы должны дать им имена, если хотите различать разные объекты одного и того же класса.

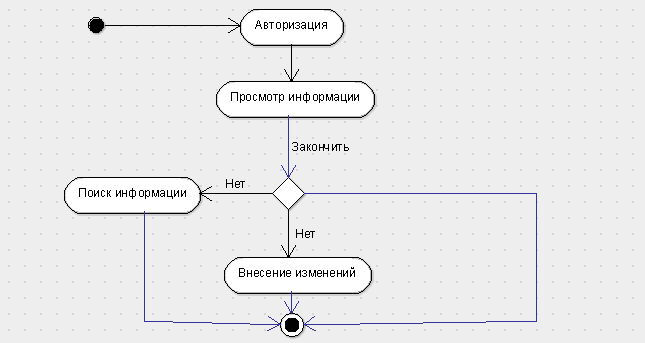


Рисунок 6 — Диаграмма деятельности

Диаграмма состояния

Диаграмма состояния (Рисунок 7) — тип диаграммы, используемый для описания поведения системы.

Поведение объекта является не только прямым следствием его входных данных, но и зависит от его предыдущего состояния. Прошлую историю объекта лучше всего можно смоделировать с помощью диаграммы состояния.

Диаграммы состояния UML (или иногда называемые конечным автоматом) показывают различные состояния объекта. Диаграммы состояния также могут показывать, как объект реагирует на различные события, переходя из одного состояния в другое. Диаграмма состояния (Рисунок 7) — это UML-диаграмма, используемая для моделирования динамической природы системы.

Диаграмма состояния обычно используется для описания зависящего от состояния поведения объекта. Объект по-разному реагирует на одно и то же событие в зависимости от того, в каком состоянии он находится. Диаграммы состояния обычно применяются к объектам, но могут быть применены к любому элементу, который имеет поведение по отношению к другим объектам, таким как: субъекты, варианты использования, методы, подсистемы систем и т.д., И они обычно используются в сочетании с диаграммами взаимодействия (обычно диаграммами последовательности).

Диаграмма конечного автомата описывает все события (а также состояния и переходы для одного объекта)

Диаграмма последовательности описывает события для одного взаимодействия между всеми задействованными объектами

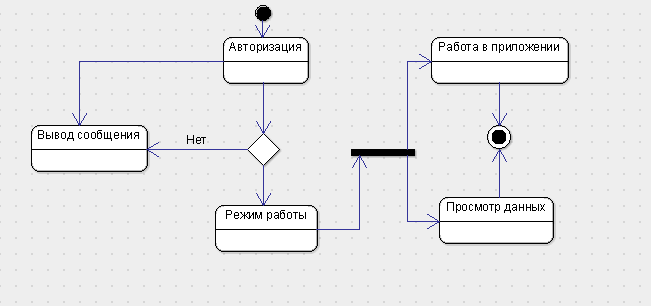


Рисунок 7 — Диаграмма состояний

Диаграмма классов

Структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов, методов, интерфейсов и взаимосвязей между ними.

В программной инженерии диаграмма классов на Унифицированном языке моделирования (UML) представляет собой тип статической структурной диаграммы, которая описывает структуру системы, показывая классы системы, их атрибуты, операции (или методы) и отношения между объектами. Назначение диаграмм классов показывает статическую структуру классификаторов в системе.

Диаграмма предоставляет базовую нотацию для других структурных диаграмм, предписанных UML

Диаграмма классов UML (Рисунок 8) состоит из набора классов и набора отношений между классами.

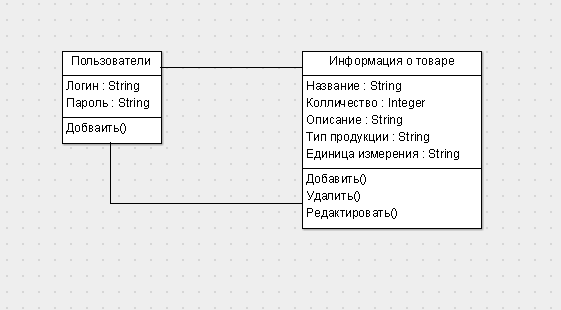


Рисунок 8 — Диаграмма классов

# 

# 2.5 Общая характеристика организации решения задачи на ПК

Объектом автоматизации выбрано рабочее место для учета товаров для АЗС.

Для того, что создать автоматизированное рабочее место необходимо:

* Разработать формы
* Выбрать информационную технологию
* Создать базу данных

К входной информации относятся сведения о товарах на АЗС.

К выходной информации относятся накладная по товарам АЗС.

Главной целью является внедрение разработанного программного продукта для сотрудников АЗС.

Приложение позволяет:

* упростить работу сотрудника.
* сократить количество бумажных документов.
* снизить финансовые затраты на печать документов.
* снизить вероятность человеческой ошибки.
* формализация деятельности.

# 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО РАБОЧЕГО МЕСТА

# 3.1. Выбор средства разработки

Хранение данных в работе: база данных MySQL Workbench.

Распечатать можно посредством документа Excel.

В программном средстве, благодаря среде разработки Microsoft Visual Studio 2019, был разработан удобный, понятный и простой в использовании интерфейс.

Программа реализована полностью в соответствии с поставленной задачей.

Проект реализован с помощью среды разработки Microsoft Visual Studio, с использованием языка C#.

Данная среда разработки предназначена для написания, проведения тестов и запуска кода. Широкий набор различных утилит, включенных в Visual Studio, позволяет выполнять проектирование и разработку полноценных приложений с использованием графического пользовательского интерфейса без привлечения стороннего ПО. Помимо разработки программ для настольных и десктопных операционных систем, поддерживается разработка веб-приложений.

Без того немалый функционал можно расширить за счет использования сторонних плагинов. В последних версиях приложения большой упор сделан на расширение средств для коллективной разработки. Внесено много улучшений и дополнений для обеспечения эффективной командной работы над проектами любого масштаба.

В Visual Studio представлены как стандартные в подобном софте инструменты — отладчик, редактор кода, — так и собственные наработки. Кроме этого, доступны решения сторонних компаний, реализованные в виде подключаемых плагинов. Для повышения продуктивности используются большое количество автоматизированных возможностей: машинные подсказки, автоматическая очистка кода, рефакторинг. Функция IntelliSense позволяет работать с документацией к программе в интерактивном режиме, получая необходимую информацию непосредственно во время написания кода. Как и во всех продуктах от Microsoft, в приложении есть строка умного поиска, благодаря которой найти нужный пункт меню или описание функционала становится значительно легче.

Список основных возможностей и преимуществ выглядит следующим образом:

* собственный компилятор;
* возможность для развертывания приложения собственными средствами;
* редактор кода, включающий в себя инструменты авто завершения и форматирования;
* отладка кода;

Автоматизированные возможности рефакторинга кода, включающего поддержку различных языков, помогают упростить процесс написания кода. В целом, Microsoft Visual Studio является хорошим выбором как для начинающих, так и для разработчиков профессионального уровня.

MySQL Workbench — это унифицированный визуальный инструмент для архитекторов баз данных и разработчиков БД.

Основной единицей хранения данных является база данных, представляющая собой совокупность таблиц с типизированными столбцами.

MySQL Workbench предоставляет возможность моделирование данных, разработку SQL и комплексные инструменты администрирования для конфигурации сервера, администрирования пользователей, резервного копирования и многое другое. MySQL Workbench доступен на Windows, Linux и Mac OS X.

# 3.2 Обзор аналогов разрабатываемого программного обеспечения

Автоматизированное рабочее место — это элемент системы управления, оснащенный средствами, с помощью которых обеспечивается участие человека в реализации функций автоматизации.

Иными словами, это проблемно-ориентированный комплекс программных, технических и лингвистических средств. Он устанавливается непосредственно на рабочем месте пользователя и предназначен для автоматизации обработки операций при проектировании и решении необходимых задач.

Было время, когда термин “автоматизация” был синонимом передовых производственных предприятий, полных робототехники. Хотя замена человеческого труда машинным трудом является ярким примером автоматизации рабочего места, это далеко не единственный пример. Автоматизация присутствует в современных компаниях всех размеров – включая тонкие функции в общих программных приложениях и более очевидные реализации, такие как самоуправляемые транспортные средства или автономные роботы.

В настоящее время рынок систем автоматизации достаточно насыщен и на нем представлены специализированные решения для учета товаров, которые изначально разрабатывались с учетом российских законов. Ниже представлены системы автоматизации деятельности учета товаров:

* Битрикс24.
* AmoCRM.
* ПланФикс.
* Мегаплан.
* Простой бизнес.
* CloudShop.

# 3.3. Разработка базы данных

Для того, чтобы разработать программный продукт я создал базы данных, функции хранения и заполнил таблицы (Рисунок 9, Рисунок 10, Рисунок 11 Рисунок 12, Рисунок 13, Рисунок 14, Рисунок 15)

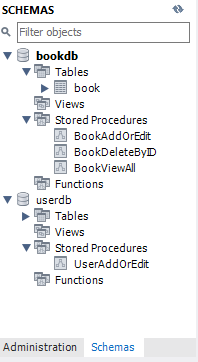


Рисунок 9 — Базы данных.

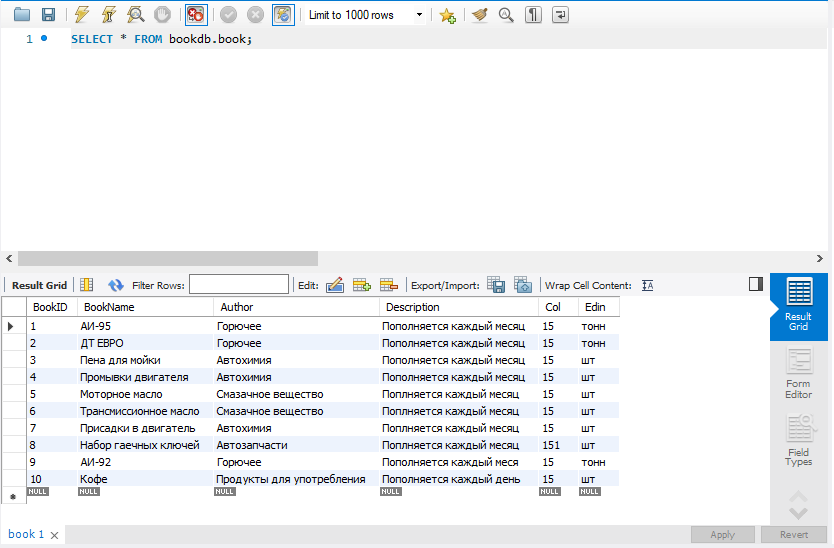


Рисунок 10 – Таблица «book»

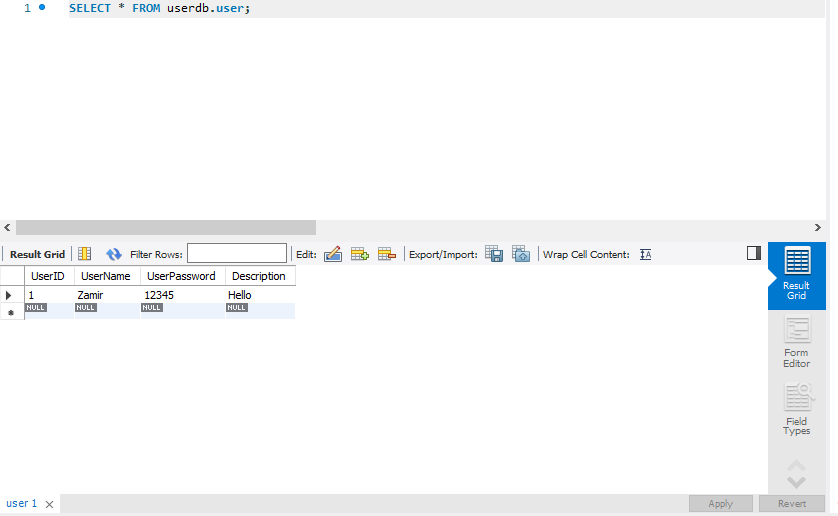


Рисунок 11- Таблица «user»

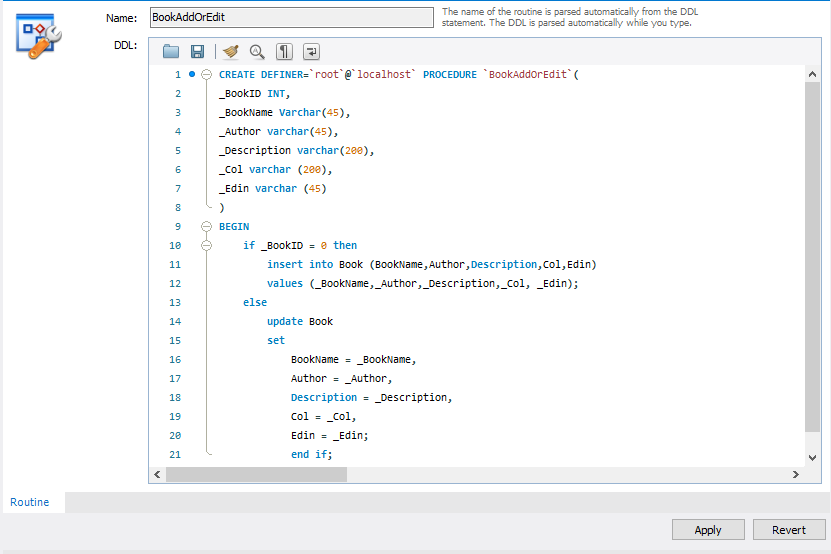


Рисунок 12 – Функция хранения «BookAddOrEdit»

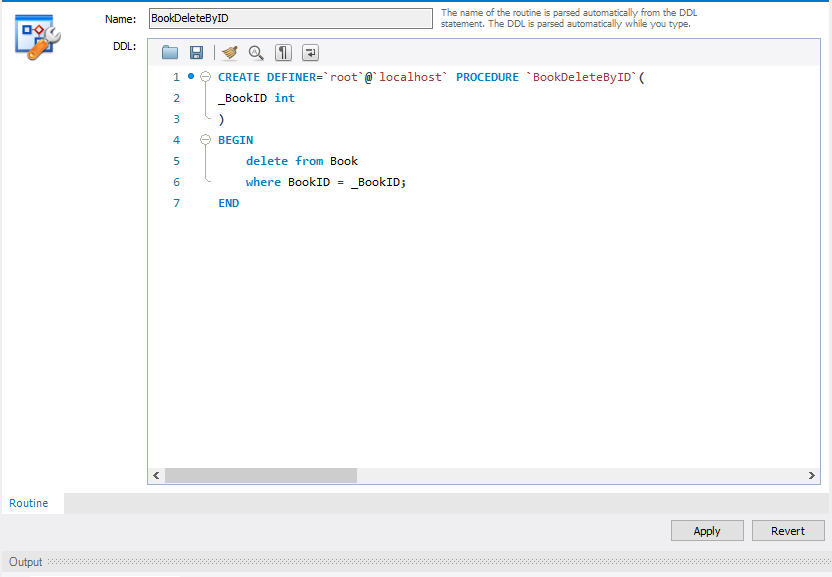


Рисунок 13 – Функция хранения «BookDeleteOrID»

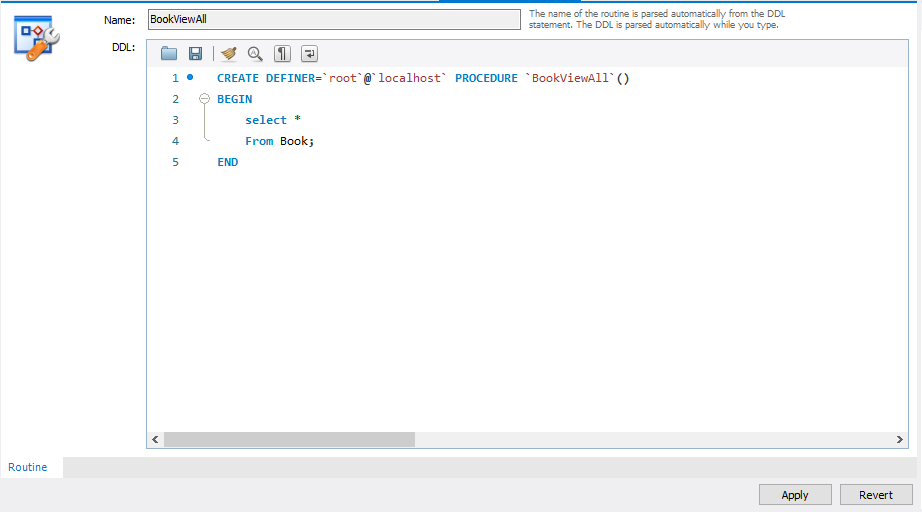


Рисунок 14 – Функция хранения «BookViewAll»

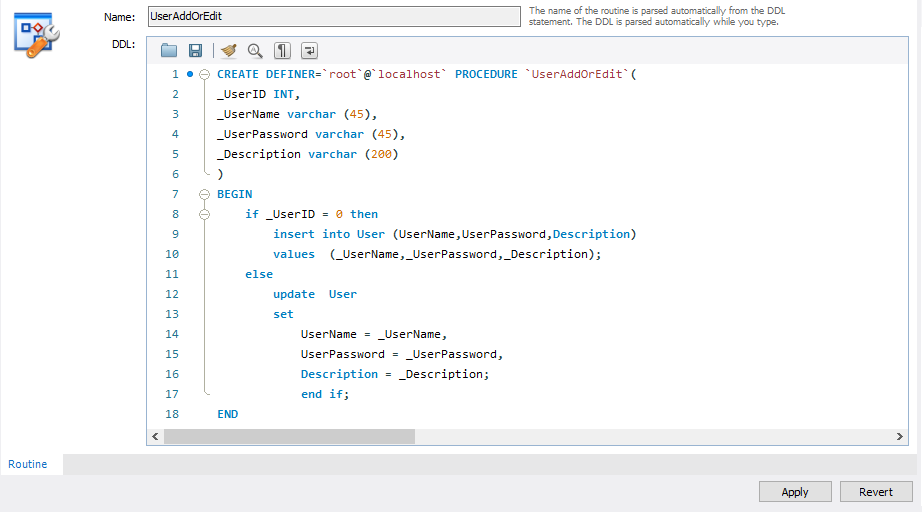


Рисунок 15 – Функция хранения «UserAddOrEdit»

**3.3 Разработка интерфейса**

В ходе разработки было принято решение использовать в системе графический интерфейс.

Графический интерфейс позволяет осуществлять взаимодействие человека с компьютером в форме диалога и с использованием окон, меню, и прочих элементов управления. Их относительно просто изучить и использовать. Пользователи, не имеющие опыта работы с компьютером, могут легко и быстро научиться работать с графическим интерфейсом.

С точки зрения удобства в программе важно создать такой пользовательский интерфейс, который сделает работу эффективной и производительной, а также обеспечит удовлетворенность пользователя от работы с программой.

В разрабатываемой информационной системе будет реализован интерфейс со свободной навигацией.

Интерфейс со свободной навигацией обеспечивает возможность осуществления любых допустимых в конкретном состоянии операций, доступ к которым возможен через различные интерфейсные компоненты. Предполагается, что взаимодействие с пользователем осуществляется посредством выбора и перемещения пиктограмм соответствующей объектно-ориентированной области.

Реализованный в информационной системе графический интерфейс включает в себя следующие формы:

* форма авторизации,
* форма регистрации,
* форма просмотра наличия товаров,
* форма редактирования товаров.

На Рисунке 15 изображена форма загрузки, которая запускает программу:



Рисунок 15 – Форма загрузки.

После формы загрузки появляется форма авторизация (Рисунок 16). После успешной аутентификации пользователь увидит форму пользователь.

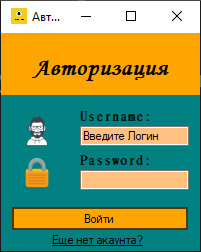


Рисунок 16 – Форма пользователь

Если пользователя первый раз в приложении, то он должен зайти в форму регистрации (Рисунок 17) нажав на «Регистрация». После успешной регистрации пользователь увидит форму «Авторизации».

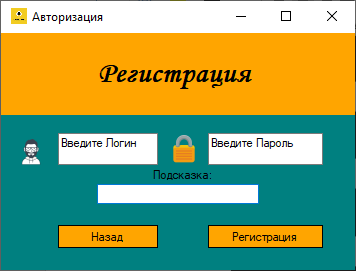


Рис 17 - Форма регистрация.

В форме Авторизация (Рисунок 16), нажав на ссылку «Еще нет аккаунта появиться форма регистрации для пользователя (Рисунок 17), где пользователь может зарегистрироваться для дальнейшей работы с приложением. После регистрации пользователю придется заново авторизоваться. Нажав на кнопку «Назад», пользователь увидит форму Авторизации (Рисунок 16).

В случае неверно набранного логина или пароля, пользователь увидит ошибку (Рисунок 18).

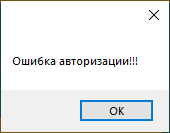


Рисунок 18

После успешной авторизации пользователь попадет на форму учета товаров (Рисунок 19).

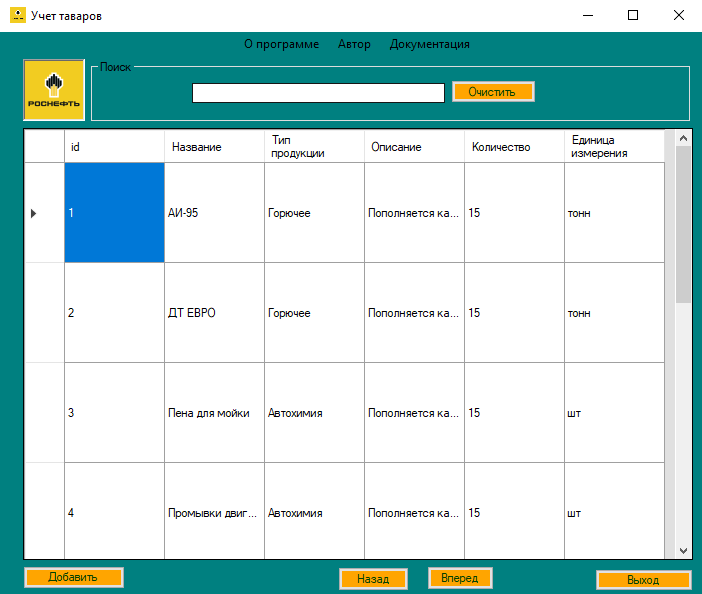


Рисунок 19 – Форма учета товаров.

Она предоставляет пользователю просмотр всех товаров, имеющихся в наличии. Нажав на любую кнопку, откроется соответствующая форма. Нажав на кнопку «Распечатать товары», пользователь увидит всплывающее окно, в котором будут выделенные материалы для печати. Нажав на кнопку «Добавить продукт», пользователь увидит форму добавления продукта (Рисунок 20), где он видит текущие данные и может как добавлять продукты, так и удалять.

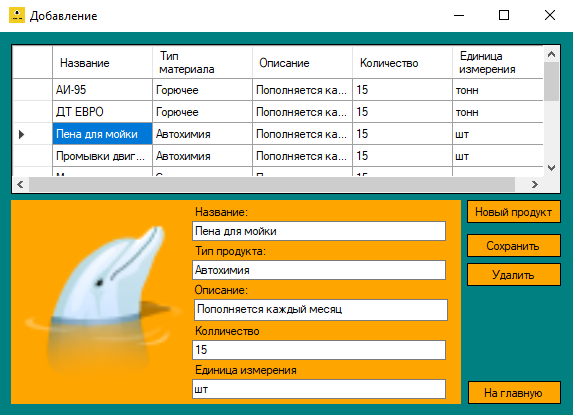


Рисунок 20 - Форма добавления.

После добавления, редактирования или удаления товаров, есть возможность импортировать ее в Excel. (Риc 21)

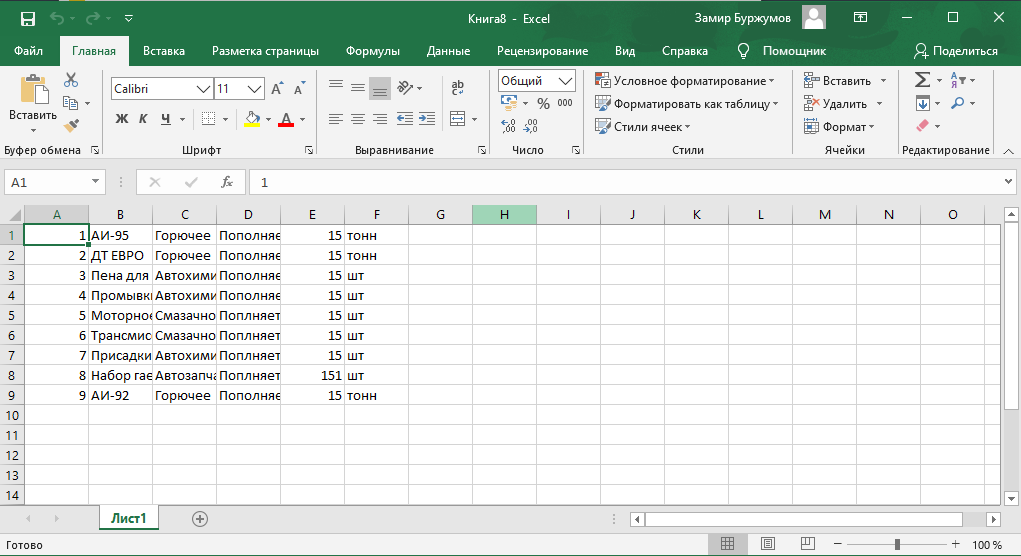


Рис 21 – Таблица “book” в Excel

Код форм представлен в приложении B.

**4 ОБОСНОВАНИЕ ПРОЕКТНЫХ РЕШЕНИЙ**

# 4.1 Оценка эффективности внедряемого программного продукта

Внедренное АРМ является надежным, своевременно удовлетворяет информационные потребности пользователей, прост в обслуживании и имеет дружественный интерфейс.

Выполняемая сотрудником работа является эффективной. Все проводимые автоматизированные процессы не вызывают дискомфорта, который оказал бы воздействие на изначальные функциональные обязанности работника.

Так же соответствует таким требованиям как:

* быстрая обработка запросов;
* соответствие уровню знаний работника;
* понятный управляемый интерфейс;
* возможность получения новых знаний, улучшения их.

Соответствует таким фундаментальным принципам формирования автоматизированного рабочего места как:

* Устойчивость
* Системность
* Гибкость
* Эффективность

Автоматизированное рабочее место является индивидуальным, оно применимо к конкретному сотруднику и представляет собой место пользователя специалиста. Решает информационные и вычислительные задачи.

**ЗАКЛЮЧЕНИЕ**

В данной дипломной работе было описано приложение для автоматизации рабочего места сотрудника АЗС. В ходе работы были выполнены такие задачи как:

* Информационные — к таким задачам относятся кодирование, классификация, сбор, структурная организация, корректировка, хранение, поиск и выдача информации. Так же включают несложные вычислительные и логические процедуры арифметического и текстового характера и отношения.
* Вычислительные — эти задачи являются как формализуемыми, так и не полностью формализуемыми. Формализуемые задачи решаются на базе формальных алгоритмов и делятся на две группы: задачи прямого счета и задачи на основе математических моделей. Задачи прямого счета решаются с помощью простейших алгоритмов.

Путем внедрения приложения была облегчена работа сотрудника. Приложение ориентировано на конечного пользователя, оно эргономично, то есть создает для пользователя комфортные условия труда и имеет дружественный и приятный интерфейс.

В результате внедрения приложения была облегчена работа сотрудника.

Система предусматривает дальнейшее совершенствование. Пути совершенствования и развития системы создание новых регистрирующих и отчетных модулей, модулей импорта и экспорта.

# Список информационных источников

1. Белов, В.В. Проектирование информационных систем: Учебник / В.В. Белов. - М.: Академия, 2018. - 144 c
2. Евдокимов П.В, C# на примерах. 4-е издание: Учебник / Евдокимов П.В. – М.:  Наука и Техника, 2019. - 320 с
3. Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова. - М.: Академия, 2018. - 272 c.
4. Кариев, Ч. А. Разработка Windows-приложений на основе Visual C# (+ CD-ROM) / Ч.А. Кариев. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, **2018**. - 768 c.
5. Дейтел, П. Как программировать на Visual C# 2012 / П. Дейтел. - М.: Питер, **2018**. **- 2180** c.
6. Троелсен Эндрю, Джепикс Филипп. Язык программирования C# 7 и платформы .NET и .NET Core. – Вильямс, 2018. – 1328с.
7. Климов, А. C#. Советы программистам / А. Климов. - М.: БХВ-Петербург, **2018**. - 544 c.
8. Гриффитс, Иэн Программирование на C# 5.0 / Иэн Гриффитс. - М.: Эксмо, **2018. - 1679** c.
9. Тестирование. Фундаментальная теория // Хабр [Электронный ресурс]. URL: https://habr.com/ru/post/279535/ (дата обращения: 05.05.2020)
10. Способы тестирования программного обеспечения// Хабр [Электронный ресурс]. URL: https://habr.com/ru/company/otus/blog/443418/ (дата обращения: 05.05.2020)
11. Полное руководство по языку программирования С# 8.0 и платформе .NET Core 3 // Metanit [Электронный ресурс]. URL: https://metanit.com/sharp/tutorial/ (дата обращения: 10.05.2020)
12. Техническое задание // Википедия [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Техническое\_задание (дата обращения: 10.05.2020)
13. Виды обеспечения АИС [Электронный ресурс]. URL: http://studopedia.ru/14\_93295\_vidi-obespecheniya-ais.html (дата обращения 20.05.2020).
14. Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс , Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, **2019**. - 736 c.
15. Культин, Н. С# в задачах и примерах / Н. Культин. - М.: БХВ-Петербург, **2020**. - **1293** c.
16. Мэтью, Мак-Дональд WPF: Windows Presentation Foundation в .NET 4.5 с примерами на C# 5.0 для профессионалов / Мак-Дональд Мэтью. - М.: Диалектика / Вильямс, **2017**. - **1100** c.

# Приложение А

**Техническое задание на разработку** **информационной системы   
учета товаров на языке программирования С# для   
 ИП Шитиков**

**Содержание**

1. Введение

1.1. Наименование программы

1.2. Назначение и область применения

2. Требования к программе

2.1. Требования к функциональным характеристикам

2.2. Требования к надежности

2.2.1. Требования к обеспечению надежного функционирования программы

2.2.2. Время восстановления после отказа

2.2.3. Отказы из-за некорректных действий пользователей системы

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

3.2. Требования к квалификации и численности персонала

3.3. Требования к составу и параметрам технических средств

3.4. Требования к информационной и программной совместимости

3.4.1. Требования к исходным кодам и языкам программирования

3.4.2. Требования к программным средствам, используемым программой

3.4.3. Требования к защите информации и программ

3.5. Специальные требования

4. Требования к программной документации

4.1. Предварительный состав программной документации

5. Технико-экономические показатели

5.1. Экономические преимущества разработки

6. Стадии и этапы разработки

6.1. Стадии разработки

6.2. Этапы разработки

6.3. Содержание работ по этапам

7. Порядок контроля и приемки

7.1. Виды испытаний

7.2. Общие требования к приемке работы

# 

# Приложение Б

Руководство пользователя по программному продукту «Учета товаров»

**1.Назначение программы.**

Программа предназначена для создания, управления содержимым базы данных, содержащим следующие данные:

* Пользователи
* Товары

Программа предоставляет интерфейс для управления содержимым базы данным в соответствии с предъявляемыми требованиями.

**2.Условия выполнения программы.**

Объем оперативного запоминающего устройства, свободного места на жестком диске 1 ГБ.

Программа должна работать на платформах Windows 7/8/10.

Программная документация поставляется в электронном и печатном виде.

Программа поставляется на лазерном носителе информации.

**3.Выполнение программы.**

Программа запускается через ярлык на рабочем столе.

# Приложение В

Код формы авторизации:

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 MySql.Data;

using MySql.Data.MySqlClient;

namespace Ychet\_Tavarov

{

public partial class Avtoriz : Form

{

public Avtoriz()

{

InitializeComponent();

txtLogin.Text = "Введите Логин";

txtLogin.ForeColor = Color.Black;

}

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

{

string login = txtLogin.Text;

string pass = txtPassword.Text;

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM userdb.user WHERE UserName = @ul AND UserPassword = @up", db.GetConnection());

command.Parameters.Add("@ul", MySqlDbType.VarChar).Value = login;

command.Parameters.Add("@up", MySqlDbType.VarChar).Value = pass;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

this.Hide();

ProductForm form = new ProductForm();

form.ShowDialog();

}

else

MessageBox.Show("Ошибка авторизации!!!");

}

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

{

this.Close();

}

Point lastpoint;

private void Avtoriz\_MouseMove(object sender, MouseEventArgs e)

{

if(e.Button == MouseButtons.Left)

{

this.Left += e.X - lastpoint.X;

this.Top += e.Y - lastpoint.Y;

}

}

private void Avtoriz\_MouseDown(object sender, MouseEventArgs e)

{

lastpoint = new Point(e.X, e.Y);

}

private void txtLogin\_Enter(object sender, EventArgs e)

{

if (txtLogin.Text == "Введите Логин")

{

txtLogin.Text = "";

txtLogin.ForeColor = Color.Black;

}

}

private void txtLogin\_Leave(object sender, EventArgs e)

{

if (txtLogin.Text == "")

{

txtLogin.Text = "Введите Логин";

txtLogin.ForeColor = Color.Black;

}

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

this.Hide();

Reg form = new Reg();

form.ShowDialog();

}

private void label1\_Click(object sender, EventArgs e)

{

}

}

}

Код формы Регистрация:

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 MySql.Data.MySqlClient;

namespace Ychet\_Tavarov

{

public partial class Reg : Form

{

string connectionString = @"server=localhost;userid=root;password=710052473459;database=userdb;charset=utf8";

int UserID;

public Reg()

{

InitializeComponent();

textBox3.Text = "Введите Логин";

textBox3.ForeColor = Color.Black;

textBox4.Text = "Введите Пароль";

textBox4.ForeColor = Color.Black;

}

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

{

}

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

{

using (MySqlConnection mysqlCon = new MySqlConnection(connectionString))

{

mysqlCon.Open();

MySqlCommand mySqlCmd = new MySqlCommand("UserAddOrEdit", mysqlCon);

mySqlCmd.CommandType = CommandType.StoredProcedure;

mySqlCmd.Parameters.AddWithValue("\_UserID", UserID);

mySqlCmd.Parameters.AddWithValue("\_UserName", textBox3.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_UserPassword", textBox4.Text.Trim());

mySqlCmd.Parameters.AddWithValue("\_Description", txtDescription.Text.Trim());

mySqlCmd.ExecuteNonQuery();

MessageBox.Show("Успешно");

}

}

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

{

this.Hide();

Avtoriz form = new Avtoriz();

form.ShowDialog();

}

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

{

if (textBox3.Text == "Введите Логин")

{

textBox3.Text = "";

textBox3.ForeColor = Color.Black;

}

}

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

{

if (textBox3.Text == "")

{

textBox3.Text = "Введите Логин";

textBox3.ForeColor = Color.Black;

}

}

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

{

if (textBox4.Text == "Введите Пароль")

{

textBox4.Text = "";

textBox4.ForeColor = Color.Black;

}

}

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

{

if (textBox4.Text == "")

{

textBox4.Text = "Введите Пароль";

textBox4.ForeColor = Color.Black;

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

}

}