**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ**

**ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«БАЛТИЙСКИЙ ИНФОРМАЦИОННЫЙ ТЕХНИКУМ»**

|  |
| --- |
| **К защите допустить:** |
| Заведующая отделом по УР  Дорофеева Е.Е. |
| Подпись |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. |
|  |

**КУРСОВАЯ РАБОТА**

**НА ТЕМУ «Разработка программы автоматизации для фирмы**

**По реализации солнечных очков»**

по профессиональному модулю ПМ. 01 «Разработка модулей программного обеспечения для компьютерных систем» / МДК 01.01 «Разработка программных модулей»

|  |  |  |
| --- | --- | --- |
|  |  | Выполнил: |
|  |  | Студентк  учебной группы П-21/2 |
|  |  | Специальность 09.02.07 |
| Курсовая работа |  | Информационные системы и |
| Выполнена с оценкой |  | программирование |
|  |  | Баусов Руслан Игоревич |
| Дата защиты |  | Подпись |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2023 г. |  |  |
|  |  |  |
| Секретарь. |  | Руководитель курсовой работы, преподаватель |
| Ф.И.О |  | Милютина Светлана Николаевна |
| Подпись |  | Подпись |

Калининград

2023

**Содержание**

[**Введение** 3](#__RefHeading___1)

[Глава 1 Назначение, область применения и UML - диаграммы 6](#__RefHeading___2)

[1.1. Описание предметной области 6](#__RefHeading___3)

[1.2. Постановка задачи 7](#__RefHeading___4)

[1.3. Диаграммы 7](#__RefHeading___5)

[Глава 2 Разработка базы данных и desktop-приложения 14](#__RefHeading___10)

[2.1. Логическая схема базы данных 14](#__RefHeading___11)

[2.2. Интерфейс приложения 18](#__RefHeading___12)

[2.3. Тестирование приложения 23](#__RefHeading___13)

[Список использованной литературы](#__RefHeading___14) 26

**Введение:**

В настоящее время солнечные очки являются неотъемлемой частью летнего гардероба. Они защищают глаза от вредного ультрафиолетового излучения и помогают сохранить зрение. В связи с этим, спрос на качественные и стильные модели растет каждый год, что делает производство таких товаров выгодным бизнесом. Для того чтобы удержать позиции на рынке и повысить эффективность работы компаний-производителей, необходимо использование новых технологий в процессе разработки продукта.

Цель данной курсовой работы - разработка программы автоматизации для фирмы по производству солнечных очков. Это позволит оптимизировать рабочие процессы, ускорить выпуск продукции и повысить ее качество.

Для достижения поставленной цели были проведены анализ текущих методик работы предприятия, изучены основные принципы автоматической обработки данных при создании программного обеспечения для бизнес-задач.

Результатом выполненной работы будет программа автоматического контроля всех этапов создания продукта: от проектирования до упаковки. Это позволит сократить время на производство, повысить качество продукции и увеличить прибыль компании.

В работе будут рассмотрены основные этапы разработки программного обеспечения, принцип работы систем автоматического контроля и анализ результатов внедрения новой технологии в бизнес-процесс предприятия.

**Актуальность темы:**

Актуальность данной курсовой работы заключается в том, что солнечные очки являются неотъемлемой частью жизни многих людей. Они защищают глаза от ультрафиолетовых лучей и помогают предотвратить различные заболевания глаз. Рынок солнечных очков постоянно растет, а значит, возникает потребность в автоматизации процессов производства и продажи этих товаров.

Разработка программы автоматизации для фирмы по реализации солнечных очков позволит оптимизировать бухгалтерский учет, контроль складских запасов и заказ товара у поставщиков. Также программа может обеспечить быстрое оформление заказов клиентами через интернет-магазин или приложение на мобильном телефоне. Кроме того, использование программного обеспечения позволяет повысить эффективность работы персонала компании за счет минимизации времени на выполнение рутинных задач и концентрации на более сложных задачах.

Таким образом, разработка программы автоматизации для фирмы по реализации солнечных очков имеет большую актуальность и может принести значительную выгоду для бизнеса.

**Цель работы:**

Цель данной курсовой работы - разработка программы автоматизации для фирмы по производству солнечных очков. Это позволит оптимизировать рабочие процессы, ускорить выпуск продукции и повысить ее качество.

**Этапы разработки Desktop-приложения:**

1. Определение требований: определить функциональные и нефункциональные требования к программе автоматизации для фирмы по реализации солнечных очков.
2. Проектирование архитектуры: разработать общую структуру приложения, выбрать технологии и инструменты для реализации задачи.
3. Разработка пользовательского интерфейса (UI): создание дизайна графического интерфейса пользователя, который будет удобен в использовании и соответствует брендингу компании.
4. Написание кода: написать код на языке программирования, используя выбранную архитектуру и технологии.
5. Тестирование приложения: проверить работоспособность приложения на соответствие заявленным требованиям; исправление ошибок в случае необходимости
6. Запуск Desktop-приложения

**Задачи работы:**

1. Изучение рынка солнечных очков и анализ конкурентов.
2. Определение требований к программе автоматизации для фирмы по производству солнечных очков.
3. Разработка функциональной спецификации программного обеспечения, включая описание основных модулей и возможностей системы.
4. Проектирование баз данных для хранения информации о клиентах, заказах на изготовление солнечных очков и технических характеристиках продукции.
5. Создание пользовательского интерфейса программы автоматизации, который будет удобен в использовании как менеджерам компании, так и клиентам-заказчикам товаров.
6. Обучение персонала работе со свежим инструментарием
7. Сопровождению новых версий приложений

**Объектом курсовой работы:** является разработка программы автоматизации фирмы по реализации солнечных очков.

Информационная база для написания работы:

Для написания курсовой работы использовались многочисленные печатные издания литературы и Интернет ресурсы по проектированию информационной системы, созданию базы данных и UML - разработке.

Структура работы состоит из двух глав: первая содержит описание предметной области, постановка задач, UML - диаграммы и их описание, вторая глава содержит разработку базы данных и desktop – приложение.

# Глава 1 Назначение, область применения и UML – диаграммы:

## 1.1 Описание предметной области:

## Предметная область курсовой работы "Разработка программы автоматизации для фирмы по реализации солнечных очков" включает в себя процесс продажи и управления складом товаров. Фирма занимается продажей различных моделей солнечных очков, которые могут отличаться цветом линз, формой оправы и другими характеристиками.

## Основными задачами программы автоматизации являются

## Учет товара на складе: программа должна позволять контролировать количество каждого типа очков на складе, а также предупреждать о необходимости заказывать новую партию при достижении минимального количества.

## Обработка заказов: программа должна позволять быстро обрабатывать заказы клиентов и проверять доступность нужной модели на складе.

## Отслеживание стадий выполнения заказа: после получения оплаты за товар, программа должна помочь отследить этап выполнения доставки до того момента, как клиент получит свою посылку.

## Анализ данных: система может использоваться для анализирования данных о продажам (например, самые популярные модели или время года со значительными изменением спрос), что может помочь компании принимать более обоснованные решения в будущем.

## В целом, разработка программы автоматизации для фирмы по продаже солнечных очков поможет улучшить процесс управления складом и заказами, а также повысит эффективность работы компании.

## 1.2 Постановка задачи:

После ознакомления с предметной областью фирмы по реализации солнечных очков необходимо поставить собственные задачи для реализации автоматизации рабочего места продавца (представителя фирмы по реализации солнечных очков).

С самого начала необходимо разобрать инфраструктуру фирмы и создать UML диаграммы следующего вида:

1. Диаграмма прецедентов, которая поможет в моделировании системы с точки зрения внешнего наблюдателя;
2. Диаграмма видов деятельности используется при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений;
3. Диаграмма последовательностей позволяет изобразить поведение нескольких объектов в рамках одного прецедента;
4. Диаграмма состояния используется для описания поведения сложных систем. Они определяют все возможные состояния, в которых может находиться объект, а также процесс смены состояний объекта в результате некоторых событий.

После подробного разбора фирмы, можно приступать непосредственно к работе.

1. Создание базы данных;
   1. Составление ER-диаграммы;
   2. Создание таблиц и связей между ними с помощью SQL management studio;
2. Разработка интерфейса приложения с помощью Visual Studio;
   1. Подключение базы данных к приложению;
3. Тестирование приложения методом Case - тестирования

## UML – диаграммы:

Проектирование UML-диаграммы помогут при [моделировании больших проектов](https://evergreens.com.ua/ru/development-services/software-architecture-design.html), в которой можно собрать как крупные, так и более мелкие детали и нарисовать каркас (схему) приложения. По нему впоследствии будет строиться код.

## Диаграмма прецедентов:

Диаграммы претендентов - это тип диаграммы поведения UML, который часто используется для анализа различных систем. Эта диаграмма позволяет визуализировать различные типы ролей в системе и то, как эти роли взаимодействуют с системой.

Этот тип диаграммы необходим для определения функциональных требований к системе. Там, где это необходимо, она описывает типичные взаимодействия между пользователями системы и самой системой и дает описание того, как работает система. Диаграмма прецедентов изображена на Рисунке 1.

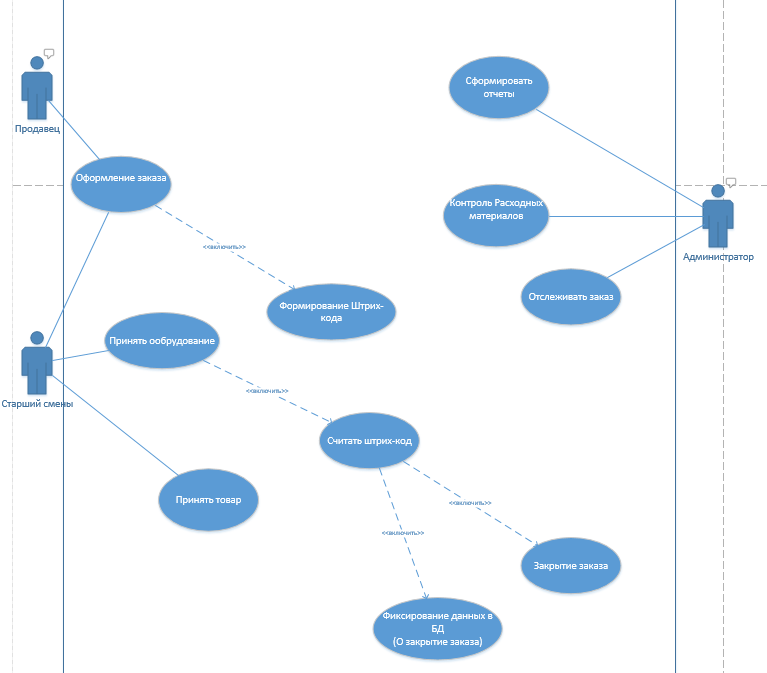


Рисунок 1 Диаграмма прецедентов

**Описание диаграммы прецедентов:**

Диаграмма прецедентов для курсовой работы на тему "Разработка программы автоматизации для фирмы по реализации солнечных очков" показывает основные действия, которые выполняет программа для автоматизации процессов фирмы. Диаграмма охватывает взаимодействие между пользователями и системой, а также между самой системой и другими системами и компонентами.

В качестве первичных акторов на диаграмме выступают Администратор и Покупатель. Администратор отвечает за управление данными на сайте, добавление и редактирование информации о продуктах, ценах и т.д. Покупатель, в свою очередь, покупает футболки на сайте и осуществляет оплату. На диаграмме представлены следующие прецеденты: "Добавление информации о продукте", "Редактирование информации о продукте", "Удаление информации о продукте", "Размещение заказа", "Оплата заказа", "Отслеживание статуса заказа", "Регистрация", "Авторизация". Каждый из этих прецедентов имеет свое описание и определенную логику работы. Например, для того чтобы зарегистрироваться на сайте, покупатель должен ввести свои персональные данные и выбрать логин и пароль. Администратор может добавлять, редактировать и удалять продукты на сайте. При размещении заказа, покупатель должен выбрать интересующий его товар, указать количество и адрес доставки.

Таким образом, диаграмма прецедентов для курсовой работы на тему "Разработка программы автоматизации для фирмы по реализации солнечных очков" является важным инструментом для определения функциональных требований к системе. Она позволяет определить основные прецеденты, которые должна выполнять система, и описать их логику работы, что является основой для создания конкретных модулей и функций в программе

## Диаграмма видов деятельности:

Диаграммы деятельности могут использоваться на всех этапах разработки программного обеспечения для различных целей. Они также обычно более популярны, чем другие типы диаграмм UML, поскольку очень похожи на блок-схемы. Диаграммы деятельности UML обеспечивают более детальную визуализацию конкретных случаев использования. Это диаграммы поведения, которые показывают поток действий в системе. Этот тип диаграмм также может использоваться для отображения потока событий в бизнес-процессе. Ее можно использовать для изучения бизнес-процессов и определения их протекания и требований.

На рисунке 2 показана диаграмма видов деятельности.

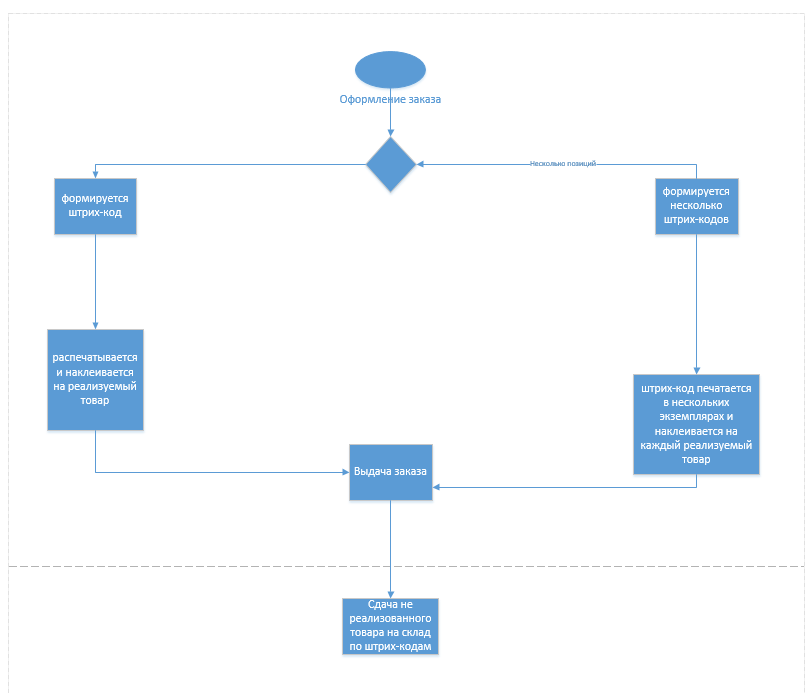


Рисунок 2 Диаграмма видов деятельности

**Описание диаграммы видов деятельности:**

Диаграмма видов деятельности для курсовой работы на тему «Разработка программы автоматизации для фирмы по реализации солнечных очков» представляет собой графическое представление основных видов деятельности, которые необходимо выполнить для успешной разработки программы автоматизации.

Диаграмма состоит из нескольких блоков-этапов, которые образуют цепочку работы. На первом этапе необходимо провести анализ бизнес-процессов и потребностей компании-заказчика. Для этого специалистам необходимо ознакомиться с основными бизнес-процессами в компании, выявить их особенности и нюансы, выяснить потребности и требования фирмы к программе автоматизации.На следующем этапе специалисты должны разработать техническое задание на программу автоматизации. В рамках технического задания будут определены основные требования к программе, ее основные функции, состав и структура системы**.** Далее наступает этап разработки самой программы автоматизации. Это основная часть работы, на этом этапе разработчики создают программный код, тестируют его и проводят отладку.

После завершения разработки программы происходит ее внедрение в работу компании-заказчика. На этом этапе специалисты проводят обучение сотрудников компании работе с новой программой, обеспечивают переход на новую систему, проводят необходимые технические настройки.И, наконец, на последнем этапе производится сопровождение и совершенствование программы автоматизации. По мере использования программы могут возникать новые потребности и требования, которые необходимо учитывать в периодическом внедрении новых обновлений и совершенствований программы.

Таким образом, диаграмма видов деятельности для курсовой работы на тему «Разработка программы автоматизации для фирмы по реализации солнечных очков» демонстрирует последовательность основных этапов работы и структуру процесса разработки программы, необходимой для автоматизации бизнес-процессов заказчика.

## Диаграмма последовательностей:

Диаграммы последовательностей, обычно используемые разработчиками, моделируют взаимодействия между объектами в едином сценарии использования. Они иллюстрируют, как различные части системы взаимодействуют друг с другом для выполнения функции, а также порядок, в котором происходит взаимодействие при выполнении конкретного случая использования. Проще говоря, диаграмма последовательности показывает различные части работы системы в “последовательности”, чтобы что-то сделать.

Схема последовательности построена таким образом, что она представляет собой временную шкалу, которая начинается сверху и постепенно опускается, чтобы отметить последовательность взаимодействий. Каждый объект имеет колонку, а сообщения, которыми обмениваются между собой, представлены стрелками.

Диаграмма последовательностей изображена на Рисунке 3.

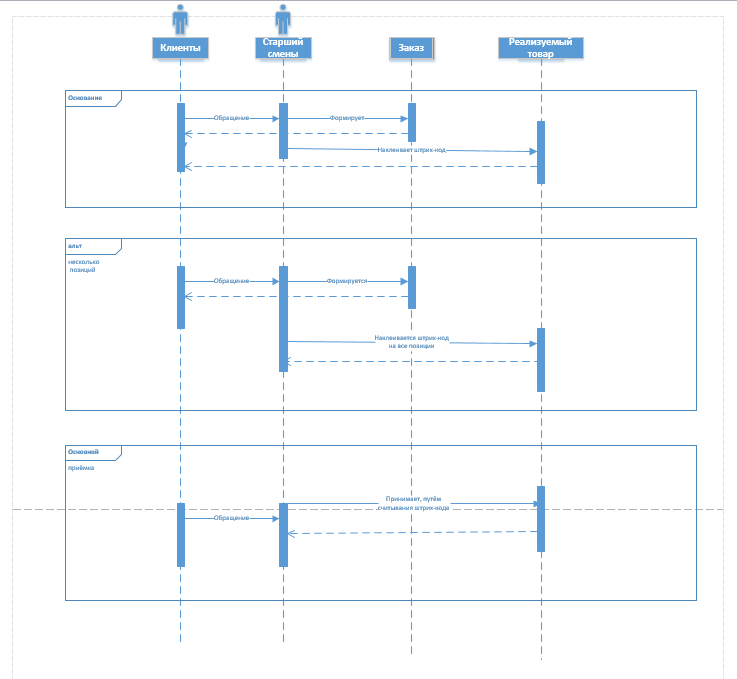


Рисунок 3 Диаграмма последовательностей

**Описание диаграммы последовательностей:**

Продавец занимается реализацией товара и подготавливает для реализации данный товар, а также продажей готового товара. Старший продавец занимается заключение договоров с поставщиком, организацией разрешений на реализацию и оформлением дизайна.

Компания занимается продажей реализацией солнечных очков, которые подготавливает продавец. Также обязанность компании заключается в правильном составлении договора при реализации и оплате труда продавцу. Продавец в свою очередь обязуется при составлении договора. В случае отказа от покупки товара по той или иной причине продавец либо оформляет возврат средств, делает замену, либо зовет старшего смены для решения вопроса.

## Диаграмма состояния:

Диаграмма состояния является одной из пяти диаграмм UML, используемых для моделирования динамической природы системы. Они определяют различные состояния объекта в течение его жизни, и эти состояния изменяются событиями. Данная диаграмма полезна для моделирования реактивных систем. Реактивные системы могут быть определены как система, которая реагирует на внешние или внутренние события.

Диаграмма состояния описывает поток управления из одного состояния в другое. Состояния определяются как состояние, при котором объект существует, и он изменяется при запуске какого-либо события. Наиболее важной целью диаграммы является моделирование времени жизни объекта от создания до завершения.

Диаграмма состояния изображена на Рисунок 4.



Рисунок 4 Диаграмма состояния

**Описание диаграммы состояния:**

Диаграмма состояния для курсовой работы на тему "Разработка программы автоматизации для фирмы по реализации солнечных очков" поможет визуализировать процесс работы программы и переходы между состояниями. На диаграмме состояний для этой программы можно выделить следующие состояния:

1. Начальное состояние - это тот момент, когда программа автоматизации запускается и ожидает действий пользователя.
2. Регистрация клиента - это состояние, когда пользователь регистрируется в программе, вводя свои данные, такие как имя, адрес электронной почты, номер телефона и т. д.
3. Выбор товара - в этом состоянии пользователь выбирает из ассортимента солнечных очков нужный ему товар.
4. Размещение заказа - после выбора товара пользователь размещает заказ, указывая желаемое количество и адрес доставки.

Диаграмма состояний для данной программы поможет увидеть последовательность действий пользователя и подскажет, на каком этапе возможны ошибки или задержки в обработке заказов. Это позволит улучшить качество работы программы и повысить удовлетворенность клиентов.

# Глава 2 Разработка базы данных и desktop-приложения:

## 2.1 Логическая схема базы данных:

ER-диаграмма является способом проектирования данных и позволяет идентифицировать понятия предметной области и связи между ними, и дает легкий удобный графический интерфейс, который описывает структуру базы данных.

ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

ER-диаграмма изображена на Рисунке 5.

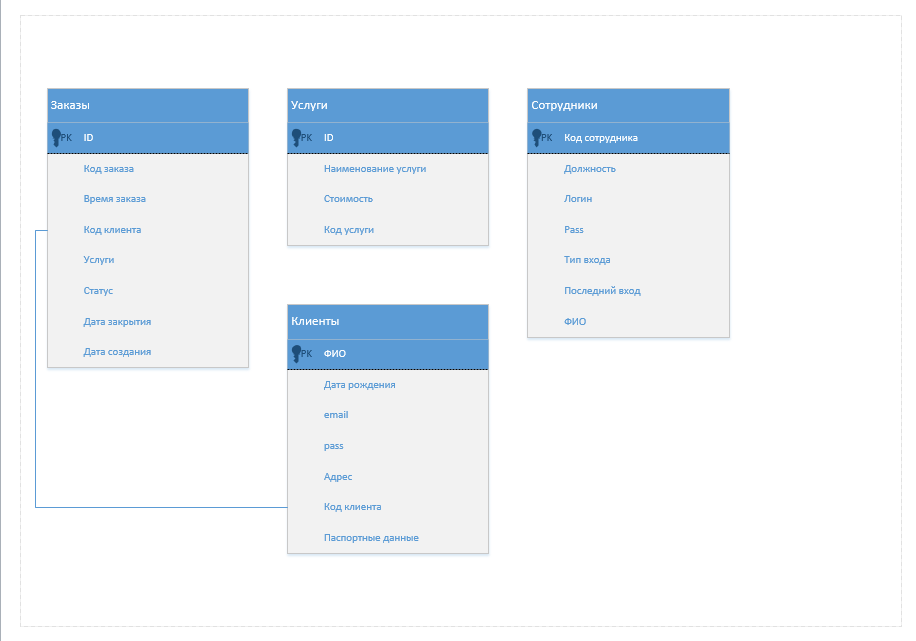


Рисунок 5 ER-диаграмма

Словари данных используются для централизованного хранения и описания ресурсов данных. Они описывают структуру базы данных и взаимосвязи между элементами базы данных.

Словари данных - это коллекция описаний данных, которая содержит информацию о хранимых в системе данных и их атрибутах. Данные словаря содержат информацию о метаданных, схемах и определениях данных, что позволяет упростить управление данными и обеспечить их последовательность. Словари данных могут использоваться в различных областях, таких как управление базами данных, анализ данных, бизнес-интеллект и т.д.

Словарь данных показан на Рисунок 6-9,9с.

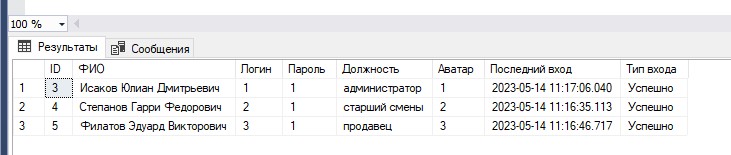


Рисунок 6 Словарь данных для таблицы «Сотрудники»

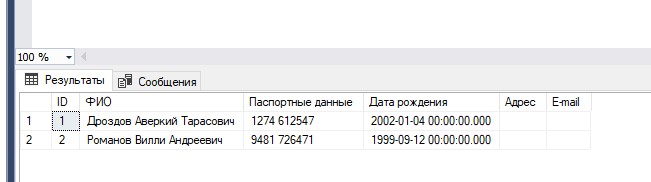


Рисунок 7 Словарь данных для таблицы «Клиенты»

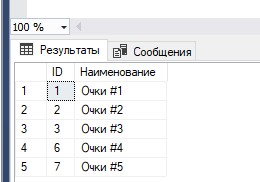


Рисунок 8 Словарь данных для таблицы «Товары»

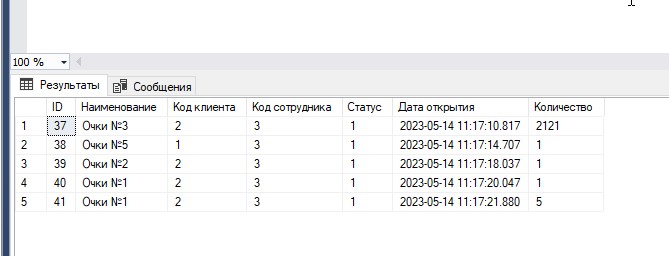


Рисунок 9 Словарь данных для таблицы «Заказы»

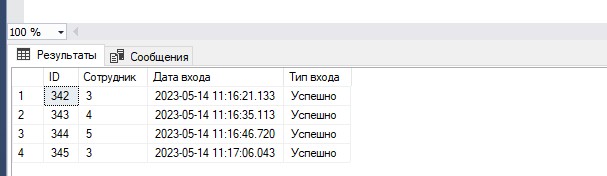


Рисунок 9с Словарь данных для таблицы «История Входа сотрудников»

Для создания базы данных было принято решение использовать SQL Management Studio 2018. База данных состоит из следующий четырех таблиц:

1. clients;
2. sotrudniks;
3. uslugs;
4. zakazs.

Рассмотрим таблицы.

Таблица clients (Рисунок 10) необходима для хранения информации о клиентах фирмы по реализации солнечных очков.

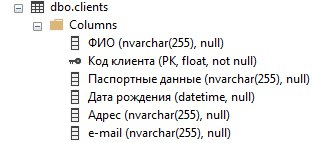


Рисунок 10 Поля таблицы «clients»

В ней имеются поля:

1. ФИО;
2. Код клиента - уникальный номер каждого клиента, поэтому данное поле выбрано в качестве первичного ключа;
3. Паспортные данные;
4. Дата рождения
5. Адрес;
6. e-mail;

Данную таблицу пополняет продавец или старший смены, когда формирует заказ, в случае, если человек пользуется услугами фирмы впервые.

Таблица sotrudniks (Рисунок 11) содержит данные о всех сотрудниках фирмы.

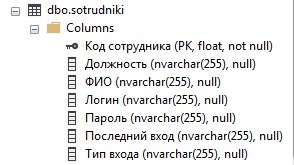


Рисунок 11 Поля таблицы «sotrudniks»

Имеющиеся поля:

1. Код сотрудника - уникальный номер каждого сотрудника, поэтому данное поле выбрано в качестве первичного ключа;
2. Должность;
3. ФИО;
4. Логин - данное поле необходимо для авторизации в приложении. Его присваивает администратор базы данных;
5. Пароль - также необходим для входа в систему;
6. Последний вход - помогает администратору баз контролировать историю входа сотрудников в программу;
7. Тип входа - обозначает успешной ли была попытка входа. Помогает администратору следить за работоспособностью приложения и исправлять ошибки.

Данную таблицу пополняет администратор, когда в фирме появляется новый сотрудник, присваивает ему логин и пароль, а также предоставляет необходимые права для работы в системе.

Таблица uslugs (Рисунок 12) хранит информацию о всех услугах, которые предоставляет компания.

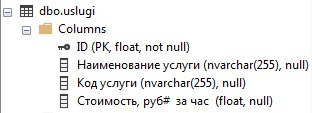


Рисунок 12 Поля таблицы «uslugs»

Поля:

1. ID - номер услуги, является первичным ключом таблицы;
2. Наименование услуги;
3. Код услуги - совмещает в себе номер услуги и дату реализации;
4. Стоимость;

Данную таблицу пополняет администратор, если у фирмы появляется новая услуга.

Таблица zakaz (Рисунок 13) необходима для хранения информации о всех заказах, выполненных компанией.

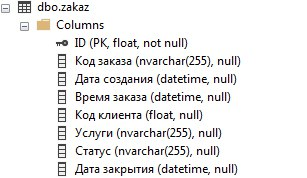


Рисунок 13 Поля таблицы «zakazs»

В ней имеются следующие поля:

1. ID - номер заказа, является первичным ключом таблицы;
2. Код услуги - совмещает в себе номер заказа и дату реализации;
3. Дата создания;
4. Время заказа;
5. Код клиента;
6. Услуги;
7. Статус - выполнен/отклонен/отменен;
8. Дата закрытия;

Данную таблицу пополняет продавец или старший смены, когда оформляет новый заказ.

## 2.2 Интерфейс приложения:

Главное окно программы по реализации солнечных очков представлено на Рисунке 14.



Рисунок 14 Главное окно приложения

Код на главное окно 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 System.Data.SqlClient;

namespace sunglasses

{

public partial class Login : Form

{

public static int idlogin = 0;

public Login()

{

InitializeComponent();

MaximizeBox = false;

this.FormClosing += new FormClosingEventHandler(Form1\_FormClosing);

}

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

{

SqlConnection conn = DBUtils.GetDBConnection();

string username, password, dolznost;

string[] fio;

int avatar, idlogin;

username = login\_text.Text;

password = password\_text.Text;

try

{

String querry = "SELECT \* FROM Сотрудники WHERE Логин = '" + login\_textbox.Text + "' AND Пароль = '" + password\_textbox.Text + "'";

SqlDataAdapter sda = new SqlDataAdapter(querry, conn);

DataTable dtable = new DataTable();

sda.Fill(dtable);

if (dtable.Rows.Count > 0)

{

DataRow collection = dtable.Rows[0];

dolznost = collection.Field<String>(4);

avatar = collection.Field<int>(5);

idlogin = collection.Field<int>(0);

fio = collection.Field<String>(1).Split(' ');

//MessageBox.Show(fio[0] + " \* " + fio[1] + " \* " + dolznost);

Form account = new Auth(fio[0], fio[1], dolznost, avatar, idlogin);

this.Hide();

account.Show();

}

else

{

MessageBox.Show("Неверное имя пользователя или пароль!");

}

}

catch (Exception ee)

{

MessageBox.Show(ee.Message + "\n" + ee.StackTrace);

}

}

private void visiblepassword\_CheckedChanged\_1(object sender, EventArgs e)

{

password\_textbox.UseSystemPasswordChar = !password\_textbox.UseSystemPasswordChar;

}

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

{

}

private void Form1\_FormClosing(object sender, System.ComponentModel.CancelEventArgs e)

{

Application.Exit();

}

}

}

В поле, справа от надписи «Логин», сотрудник вводит свой логин, выданный ему администратором. Ниже вводит свой пароль (Рисунок 15).

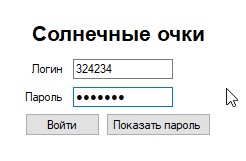


Рисунок 15 Данные для авторизации введены в поля

Пароль скрыт маской ввода, но при нажатии на кнопку «Показать», его можно увидеть (Рисунок 16).

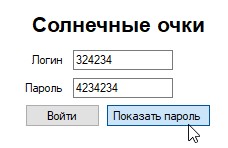


Рисунок 16 Отображение введенного пароля

При нажатии кнопки «Войти», расположенной ниже поля ввода пароля, программа сравнивает введенные данные в оба поля с данными таблицы sotrudniks в базе данных, а также проверяет должность, присвоенную этому сотруднику.

Если логин и пароль введены верно и соответствуют должности «продавец», то авторизация проходит успешно и на экране появляется сообщение о занимаемой должности, появляется новое, с выбором дальнейшего действия, а именно: «Оформление заказа» и «Заказы» (Рисунок 17).

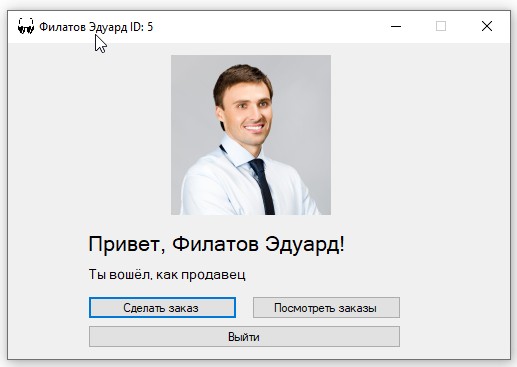


Рисунок 17 Окно продавца

Окно Администратора, если логин и пароль введены верно и соответствуют должности «администратор», то авторизация проходит успешно и на экране появляется сообщение о занимаемой должности (Рисунок 18).

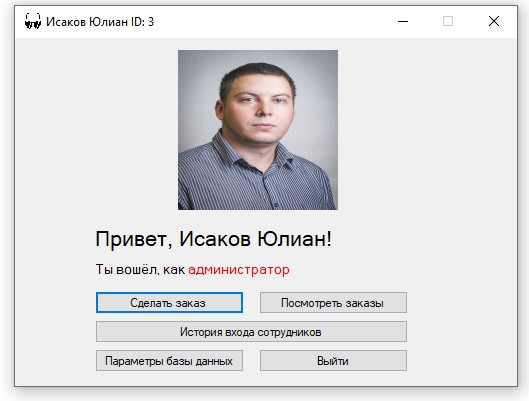


Рисунок 18 Окно администратора

После закрытия этого окна, появляется новое, с выбором дальнейшего действия, а именно: «Контроль истории входа» и «Работа с базой данных»

## 2.3 Тестирование приложения

Case-тест.Одна из важных вещей в разработке приложения. Это проверка работоспособности приложения. Case-тест— это способ тестирования программного продукта, основанный на анализе конкретных кейсов использования (сценариев работы) приложения. Каждый тест-кейс представляет собой определенную последовательность действий, которые пользователь может выполнить в приложении. В рамках такого тестирования проверяются функциональность, интерфейс и общая работоспособность программного продукта. Case-тестирование часто используется для тестирования мобильных приложений и веб-сайтов, где важна простота и интуитивность интерфейса. Тестирование изображено в таблицах 1-7.

Таблица 1

Расшифровка тестовых информационных полей

|  |  |
| --- | --- |
| Поле | Описание |
| Название проекта | Название тестируемого проекта |
| Рабочая версия | Версия проекта/программного обеспечения (первый тест считается 1.0) |
| Имя тестирующего | Имя того, кто проводил тесты |
| Дата(ы) теста | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста |
| Тестовый номер # | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. |
| Приоритет тестирования (низкий/средний/ высокий) | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет |
| Заголовок/ название теста | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем |
| Краткое изложение теста | Описание того, что должен достичь тест |
| Этапы теста | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея |
| Тестовые данные | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа |
| Ожидаемый результат | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране |
| Фактический результат | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста |
| Предварительное условие | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая |
| Постусловие | Каким должно быть состояние системы после выполнения теста? |
| Статус (зачет/незачет) | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено |
| Приоритет тестирования (низкий/средний/ высокий) | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет |
| Примечания/ комментарии | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами) |
| Этапы теста | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея |

Продолжение таблицы 1

|  |  |
| --- | --- |
| Примечания/ комментарии | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами) |

Таблица 2

Аннотация теста

|  |  |
| --- | --- |
| Название проекта | Автоматизация фирмы по реализации Солнечных очков |
| Рабочая версия | 1.0 |
| Имя тестирующего | Артем Губан |
| Дата(ы) тестирования | 2 апреля |

Таблица 3

Тестирование корректного запуска приложения

|  |  |
| --- | --- |
| Тестовый пример #1 |  |
| Тестовый пример # | TC\_AD\_1 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Запуск приложения |
| Краткое изложение теста | Запуск приложения |
| Этапы теста | 1. Произвести двойной щелчок левой кнопки мыши по ярлыку приложения на рабочем столе компьютера. |
| Тестовые данные | Ярлык приложения на рабочем столе компьютера |
| Ожидаемый результат | Приложение запустится без ошибок и на экране появится главное окно приложения |
| Фактический результат | Приложение запустилось без ошибок и на экране появилось главное окно приложения |
| Предварительное условие | Приложение установлено на компьютере |

Продолжение таблицы 3

|  |  |
| --- | --- |
| Постусловие | На экране главное окно приложения |
| Статус (зачет/незачет) | Зачет |
| Примечания/комментарии |  |
| Ожидаемый результат | Приложение запустится без ошибок и на экране появится главное окно приложения |

Таблица 4

Тестирование возможности просмотра пароля

|  |  |
| --- | --- |
| Тестовый пример #2 |  |
| Тестовый пример # | TC\_AD\_2 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Возможность просмотра пароля при авторизации |
| Краткое изложение теста | Снятие маски ввода пароля при нажатии на кнопку “Показать” |
| Этапы теста | 1. Запустить приложение; 2. Ввести данные в поле справа от надписи “Пароль”; 3. Нажать на кнопку “Показать” справа от поля ввода пароля. |
| Тестовые данные | R12fgsd |
| Ожидаемый результат | В поле для ввода пароля маска ввода в виде черных точек изменится на введенный текст |
| Фактический результат | В поле для ввода пароля маска ввода в виде черных точек изменилась на введенный текст |
| Предварительное условие | На экране главное окно приложения |
| Постусловие | В поле ввода пароля отображается введенный текст Rafgrjn |
| Статус (зачет/незачет) | Зачет |
| Этапы теста | 1. Запустить приложение; 2. Ввести данные в поле справа от надписи “Пароль”; 3. Нажать на кнопку “Показать” справа от поля ввода пароля. |
| Тестовые данные | R12312123 |

Таблица 5

Тестирование авторизации сотрудника с должностью продавца

|  |  |
| --- | --- |
| Тестовый пример #3 |  |
| Тестовый пример # | TC\_AD\_3 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация сотрудника с должностью продавца |
| Краткое изложение теста | Авторизация сотрудника с должностью продавца |
| Этапы теста | 1. Запустить приложение; 2. Ввести соответствующие данные в поля справа от слов “Логин” и “Пароль”; 3. Нажать на кнопку “Войти”. |
| Тестовые данные | 1. 24rsda 2. hdfgfqas23 |
| Ожидаемый результат | На экране появится сообщение “Вы продавец!”, после закрытия которого, отобразится окно “Продавец” |
| Фактический результат | На экране появилось сообщение “Вы продавец!”, после закрытия которого, отобразилось окно “Продавец” |
| Предварительное условие | На экране главное окно приложения |
| Постусловие | На экране окно “Продавец” |
| Статус (зачет/незачет) | Зачет |
| Примечания/комментарии |  |
| Фактический результат | На экране появилось сообщение “Вы продавец!”, после закрытия которого, отобразилось окно “Продавец” |
| Фактический результат | На экране появилось сообщение “Вы продавец!”, после закрытия которого, отобразилось окно “Продавец” |
| Предварительное условие | На экране главное окно приложения |
| Постусловие | На экране окно “Продавец” |
| Статус (зачет/незачет) | Зачет |
| Примечания/комментарии |  |

Таблица 6

Тестирование авторизации сотрудника с должностью старшего смены

|  |  |
| --- | --- |
| Тестовый пример #4 |  |
| Тестовый пример # | TC\_AD\_4 |
| Приоритет тестирования | Высокий |
| Заголовок/название теста | Авторизация сотрудника с должностью старшего смены |
| Краткое изложение теста | Авторизация сотрудника с должностью старшего смены |
| Этапы теста | 1. Запустить приложение; 2. Ввести соответствующие данные в поля справа от слов “Логин” и “Пароль”; 3. Нажать на кнопку “Войти”. |
| Тестовые данные | 1. Sfa2gdfgds 2. 42gsdfdsf |
| Предварительное условие | На экране главное окно приложения |
| Статус (зачет/незачет) | Зачет |
| Примечания/комментарии |  |

**Заключение:**

В заключение, можно сказать, что разработка программы автоматизации для фирмы по производству солнечных очков является важным шагом в улучшении эффективности и качества работы предприятия. Реализация данной программы позволит значительно ускорить процесс изготовления продукции, а также повысить точность и надежность ее изготовления.

Более того, использование автоматических систем контроля качества поможет минимизировать количество брака и повторных работ. В целом, разработка программного обеспечения для автоматизации процессов на предприятии – это инвестиция в будущее компании и гарантированный способ достичь успеха на рынке конкурентной продукции.

В качестве инструментов использовались Microsoft Visio 2018, SQL Management Studio 2020 и MS Visual Studio 2019 .

# Список используемых источников:

*1. "Разработка программного обеспечения для автоматизации бизнес-процессов" - А.С. Кузнецов, Издательство "БХВ-Петербург", 2018 г.*

*2. "Автоматизация бухгалтерского учета в малом и среднем бизнесе" - Е.В. Лебедева, Издательство "Эксмо", 2017 г.*

*3. "Программное обеспечение для автоматической оптимизации процессов проектирования и разработки изделий" - Владимир Соколовский, журнал «Научно-технические ведомости Санкт-Петербургского государственного политехнического университета», №4 (234), 2020 г.*

*4."Разработка системы автоматической документации на основе программных компонентов Microsoft Office Word и Excel"- Михайлов Денис Николаевич ,журнал «Информационные технологии», №12(96), 2020*

*5."Организация эффективной работы представительства фирмы с помощью CRM систем"- Белозуб А., журнал «Международный научно – практичный журнал», №4(12), 2021 г.*

# 