

Аттестационный лист
по производственной практике
студента Московского техникума космического приборостроения МГТУ им. Н.Э. Баумана

Бобылёва Сергея Сергеевича

(Ф.И.О. студента)

Группа **ТИП- 71**

Специальность **09.02.07 Информационные системы и программирование** Квалификация - **Программист**

(код, наименование специальности)

прошел (ла) производственную практику **ПП 04 ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем**

(наименование практики)

по профессиональному модулю **ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем** в объеме **144 часов**

(наименование профессионального модуля)

с «01» декабря 2021 года по «28» декабря 2021 года

на предприятии (организации) **Новая Басманная ул., 20, Москва,**

(юридический адрес предприятия (организации))

Виды и качество работ в период производственной практики

Виды работ, выполненные студентом во время практики, согласно программы производственной практики	Результат (по 5-ти бальной шкале)
Сопровождение и обслуживание программного обеспечения компьютерных систем	

В ходе производственной практики студентом освоены следующие профессиональные компетенции

Код и название профессиональной компетенции	Результат освоения (освоена/не освоена)
ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем.	
ПК 4.2. Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем	
ПК 4.3. Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика	
ПК 4.4. Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.	

Характеристика профессиональной деятельности обучающегося во время производственной практики: *за время производственной практики Бобылёв С.С. проявил высокую степень ответственности, трудолюбие, активное участие в работе при выполнении заданий.*

Рекомендуемая оценка по практике 5

Руководитель практики от предприятия (организации)

зам. начальника отдела

Должность

Подпись

М.П. *Будков О.О.*

Ф.И.О. руководителя практики

Итоговая оценка по практике

Руководитель практики от образовательного учреждения

Должность

Подпись

Ф.И.О. руководителя практики

Начальник управления
по работе с персоналом
АО "ЦНИРТИ
им. академика А.И. Берга"



А.В. Смоленская
















Дневник




Прохождения производственной практики Бобылёва Сергея Сергеевича
(ФИО)

по профессиональному модулю ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем

студента группы ТИП-71

Код, специальность 09.02.07

Дата	Тема занятия	Объем выполненной работы	Оценка, подпись руководителя
01.12.2021	Организационное собрание по практике.	Проходили инструктаж по организации практики, по технике безопасности, противопожарной защите. Работа по определению предметной области, должностной инструкции и нормативным документам.	5 
02.12.2021	Основные этапы проектирования и внедрения информационных систем.	Реализация на современном ПО основных этапов проектирования и внедрения информационных систем.	5 
03.12.2021	Методологии внедрения информационных систем.	Реализация на современном ПО методологии внедрения информационных систем (методология внедрения ASAP, методологии внедрения компании Oracle, методологии внедрения Microsoft, методология разработки и внедрения RUP).	5 
06.12.2021	Организация и документирование процесса внедрения информационной системы.	Организовывал и документировал процесс внедрения информационной системы.	5 
07.12.2021	Инструментарий внедрения информационной системы.	Реализация на современном ПО инструментариев внедрения информационной системы.	5 
08.12.2021	Установка, конфигурирование и настройка сетевых и телекоммуникационных средств.	Реализация на современном ПО установки, конфигурирования и настройки сетевых и телекоммуникационных средств.	5 
09.12.2021	Организация сопровождения и восстановления работоспособности системы.	Реализация на современном ПО сопровождения и восстановления работоспособности системы.	5 
10.12.2021	Обеспечение безопасности функционирования информационной системы.	Обеспечивал безопасность функционирования информационной системы.	5 
13.12.2021	Идентификация и устранение ошибок в информационной системе.	Искал и устранял ошибки в информационной системе.	5 
14.12.2021	Методы и инструменты тестирования приложений. Выявление аппаратных ошибок.	Изучал методы и инструменты тестирования приложений. Выявлял аппаратные ошибки.	5 
15.12.2021	Интеграция информационных систем.	Изучал методы интеграции информационных систем.	5 
16.12.2021	Информационные системы управления.	Изучал информационные системы управления (интегрированные информационные системы, системы поддержки принятия решений).	5 
17.12.2021	Анализ данных в информационных системах.	Реализация на современном ПО технологий анализа данных (технология OLAP, интеллектуальный анализ данных).	5 
20.12.2021	Информационные системы управления бизнес процессами.	Изучение информационных систем управления бизнес процессами (системы MRP и CRP, ERP системы, MES системы, SCM системы).	5 
21.12.2021	Виды информационных систем.	Изучение информационных систем (системы автоматизированного проектирования разработки и оформления конструкторской и проектной документации, системы автоматизированного проектирования технологических процессов, системы управления производственными данными PDM, системы удаленного управления и контроля объектов, информационные системы реального времени, информационные системы поисковосправочных служб библиотек, геоинформационные системы).	5 

22.12.2021	Надежность и качество информационных систем.	Определение надежности, контроль сбоя и отказов информационных систем, описание метрик качества программного продукта.	5 
23.12.2021	Надежность и качество информационных систем.	Определение надежности, контроль сбоя и отказов информационных систем, описание метрик качества программного продукта.	5 
24.12.2021	Оформление документации.	Оформление дневника, отчета по практике и производственной характеристики.	5 

Руководитель практики от
предприятия

М.П.

(подпись, расшифровка подписи)

« 27 » 12 20 21 г.

Начальник управления
по работе с персоналом
АО "ЦНИРТИ
им. академика А.И. Берга"





ХАРАКТЕРИСТИКА по освоению профессиональной компетенции в период прохождения ПРОИЗВОДСТВЕННОЙ практики

(вид практики)

Практикант	<i>Бобылёв Сергей Сергеевич</i> (фамилия, имя, отчество)			
Прибыл (-а) к месту практики	«01»	декабря	20	21 г.
Закончил (-а) практику	«28»	декабря	20	21 г.

За время прохождения практики: производственная

(вид практики: производственная, учебная, преддипломная)

В ходе производственной практики изучал администрирование сетей с помощью командной строки, разработал БД и изучал основы SQL. Разработал и протестировал программное обеспечение. На основании проведённой работы студент определил направления совершенствования своих практических навыков в части разработки ПО. Можно отметить ответственность, высокую степень организованности при выполнении практического задания. На основании вышеизложенного за выполнение задания практики Бобылёв Сергей Сергеевич заслуживает оценки "отлично".

Руководитель практики от
предприятия

М.П.

Будков О.В.
(подпись, расшифровка подписи)

«27» 12 2021 г.

Начальник управления
по работе с персоналом
АО "ЦНИРТИ
им. академика А.И. Берга"



А.В. Смирнов

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э.Баумана»
(национальный исследовательский университет)
Московский техникум космического приборостроения

ОТЧЕТ О ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ

по профессиональному модулю ПМ.04 СОПРОВОЖДЕНИЕ И ОБСЛУЖИВАНИЕ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Код, 09.02.07 Информационные системы и программирование

Квалификация Программист

Место прохождения практики АО «ЦНИРТИ им. академика А.И. Берга»
(полное название организации)

Выполнил студент Бобылёв Сергей Сергеевич
(фамилия, имя, отчество)

Курс 4 Группа ТИП-71

Подпись студента _____

Оценка _____

Дата приема зачета « _____ » _____ 2021 г.

Руководитель практики от техникума _____

Начальник управления
по работе с персоналом
АО «ЦНИРТИ

(подпись)

(фамилия, имя, отчество)

Руководитель практики от предприятия _____

А.В. Сидорова

(Будков Олег Дмитриевич)

(подпись)

(фамилия, имя, отчество)



Москва 2021

Индивидуальное задание
студента 4 курса Московского техникума космического приборостроения МГТУ имени
Н.Э.Баумана, специальность 09.02.07 Информационные системы и программирование

Квалификация - Программист

Бобылёва Сергея Сергеевича

(Фамилия, Имя, Отчество)

на период прохождения производственной практики

ПМ.04 Сопровождение и обслуживание программного

обеспечения компьютерных систем

Студент во время практики с 01 декабря по 28 декабря 2021 г.


Должен выполнить следующие виды работ:

Освоенные профессиональные компетенции	Виды деятельности (работы)
ПМ.04 Сопровождение и обслуживание программного обеспечения компьютерных систем	
ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем. ПК 4.2 Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем ПК 4.3 Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика ПК 4.4 Обеспечивать защиту программного обеспечения компьютерных систем программными средствами. ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие. ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с	- Инструктаж по организации практики. Инструктаж по технике безопасности, противопожарной защите и правилам внутреннего трудового распорядка на объектах практики. Изучение программного обеспечения предприятия. - Изучение документооборота организации и источников входящей информации. - Выполнение работ по ведению информационных систем предприятия. Специалист должен знать: • важность рассмотрения всех возможных вариантов и выбора лучшего решения на основе взвешенного аналитического суждения и интересов клиента; • важность использования системного анализа и методологий проектирования (например, унифицированного языка моделирования (Unified Modelling Language), программной платформы MVC (Model-View-Control), фреймворков, шаблонов проектирования); • необходимость быть в курсе новых технологий и принимать решение о целесообразности их применения; • важность оптимизации архитектуры системы с учетом модульности и повторного использования; • важность рассмотрения всех возможных вариантов и выбора лучшего решения для удовлетворения требований пользователя и интересов клиента; • важность использования методологий разработки системы (например, объектно-ориентированные технологии); • важность рассмотрения всех нормальных и ненормальных сценариев и обработки исключений;

<p>коллегами, руководством, клиентами. ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. ОК 6. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности ОК 9. Использовать информационные технологии в профессиональной деятельности. ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке. ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере</p>	<ul style="list-style-type: none"> • важность соблюдения стандартов (например, соглашения по формату кода, руководства по стилю, дизайна пользовательского интерфейса, управления каталогами и файлами); • важность точного и постоянного контроля версий; • важность использования существующего кода в качестве основы для анализа и модификации; • важность выбора наиболее подходящих средств разработки из предложенных вариантов. • принципы устранения распространенных проблем программных решений; • важность отладки программных решений; • важность тщательного тестирования программных решений. <p>Специалист должен уметь: Анализировать системы с помощью:</p> <ul style="list-style-type: none"> • структурного моделирования и анализа (например, объекты, классы, диаграммы классов предметной области); • инструментов и методов моделирования (например, диаграмма сущностей и связей, нормализация, словарь данных). Проектировать системы на основе: • диаграммы классов, диаграммы последовательностей, диаграммы состояний, диаграммы деятельности; • схемы реляционной или объектной базы данных и диаграмм потоков данных; • структуры человеко-машинного интерфейса / механизма взаимодействия с пользователем; • использовать системы управления базами данных для построения, хранения и управления структурами и наборами данных для требуемой системы на основе клиент-серверной архитектуры; • использовать подходящие версии программного обеспечения, среды разработки и инструменты, предназначенные для изменения, существующего и написания нового исходного кода клиент-серверного программного обеспечения; • использовать подходящие версии программного обеспечения, среды разработки и инструменты, предназначенные для изменения, существующего и написания нового исходного кода для системной интеграции с использованием веб-решений, веб-сервисов или единой подписки (например, с использованием службы каталогов) или API; • определять и интегрировать соответствующие библиотеки и фреймворки в программные решения; • строить и обслуживать многоуровневые приложения. • управлять версионностью разработанного программного решения.
--	--

	<ul style="list-style-type: none"> • осуществлять отладку программных решений; • разрабатывать модульные и интеграционные тесты; • устранять и исправлять ошибки в программных решениях. <p>- Оформление отчета по практике.</p>
--	---

Руководитель практики

 Будков О.О.
(подпись)

«___» декабря 2021 г.

Начальник управления
по работе с персоналом
АО "ЦИРТИ
им. академика А.И. Берга"



А.В. Вишневская

СОДЕРЖАНИЕ

Введение.....	6
1 Постановка задачи.....	7
2 Структура программы.....	9
3 Схема данных.....	13
4 Оптимизация программы	17
5 Тестирование программы.....	19
6 Руководство пользователя.....	25
Заключение.....	34
Список использованных источников.....	35
Приложение А Листинг программы.....	37
Приложение Б Результат выполнения программы.....	60
Приложение 1 Аттестационный лист.....	67
Приложение 2 Дневник.....	69
Приложение 3 Характеристика.....	74

ВВЕДЕНИЕ

База данных (БД) “sklad” создана в рамках прохождения производственной практики ПП.04.

Для ее реализации необходимо решить следующие задачи:

- 1) Изучить предметную область, т.е. изучить сферу деятельности организации.
- 2) Проанализировав информацию, полученную в ходе выполнения первого пункта, построить необходимые схемы для дальнейшей эффективной разработки базы данных.
- 3) Выбрать средство для разработки и приступить к началу работы.

После достижения результатов по вышеперечисленным пунктам заполнить созданную базу данных и создать запросы для корректного извлечения информации и удобства дальнейшей ее эксплуатации.

Для разработки были выбраны сервис MySQL и язык программирования C#.

MySQL— это система управления реляционными базами данных, разработанная корпорацией Microsoft. Эта СУБД учитывает все современные требования к работе с данными различных форматов. MySQL упрощает развертывание, передачу и интеграцию больших данных.

Для разработки приложения будет использоваться Microsoft Visual Studio.

1 Постановка задачи

Разработать приложение для работы с базой данных (БД) sklad. Приложение должно обеспечивать возможности внесения, изменения и удаления данных, разделение уровня доступа для администратора БД и сотрудника.

Для реализации нужно:

- а) Создать базу данных, используя наиболее подходящую к выданному заданию платформу (MySQL / Microsoft SQL Server).
- б) Создать приложение, используя наиболее приемлемую для решения задачи платформу: .NET (или Java).
- в) Создать форму для авторизации зарегистрированных пользователей.
- г) Реализовать разделение уровня доступа для администратора БД и сотрудников.
- д) Пользователи должны входить в систему, используя форму авторизации. Форма авторизации должна соответствовать следующим требованиям:

- 1) Для успешной авторизации необходимо ввести свой логин и пароль.
- 2) При попытке входа введенные данные проверяются на совпадение в базе данных. Должно выдаваться сообщение об ошибке в случае неправильного ввода связки логин/пароль.
- 3) После успешного входа пользователя в зависимости от его роли происходит перенаправление на конкретные формы, предназначенные для работы этого типа пользователя.
- 4) При нажатии на кнопку "Выход" в любом окне/форме системы, пользователи должны быть отключены от системы и совершен переход на Форму авторизации.
- 5) Необходимо предусмотреть возможность регистрации новых пользователей определенной роли. Поэтому нужно добавить в систему форму регистрации. При регистрации пароль должен отвечать следующим требованиям:

– должен содержать от 6 до 16 символов;

– не должно быть символов из набора: * & { } | +;

– должны встречаться заглавные буквы;

– должны встречаться цифры.

е) Должны быть реализованы основные формы для работы с сущностями, базы данных.

ж) Необходимо реализовать автоматизацию не менее 2 бизнес-процессов для каждого из видов пользователей и необходимых для этого форм.

з) Реализовать форму вывода отчета с возможностью печати документа.

и) Для выполнения процедуры тестирования корректного ввода пароля при регистрации заказчика вам нужно создать два модульных теста, которые будут проверять различные комбинации условий ввода пароля.

к) Необходимо, чтобы модульные тесты демонстрировали как положительный, так и отрицательный результаты при разных входных данных. Набор модульных тестов должен быть реализован в отдельном проекте. Все тесты должны выполняться последовательно.

2 Структура программы

На рисунках 2.1-2.3 представлена структура программы.

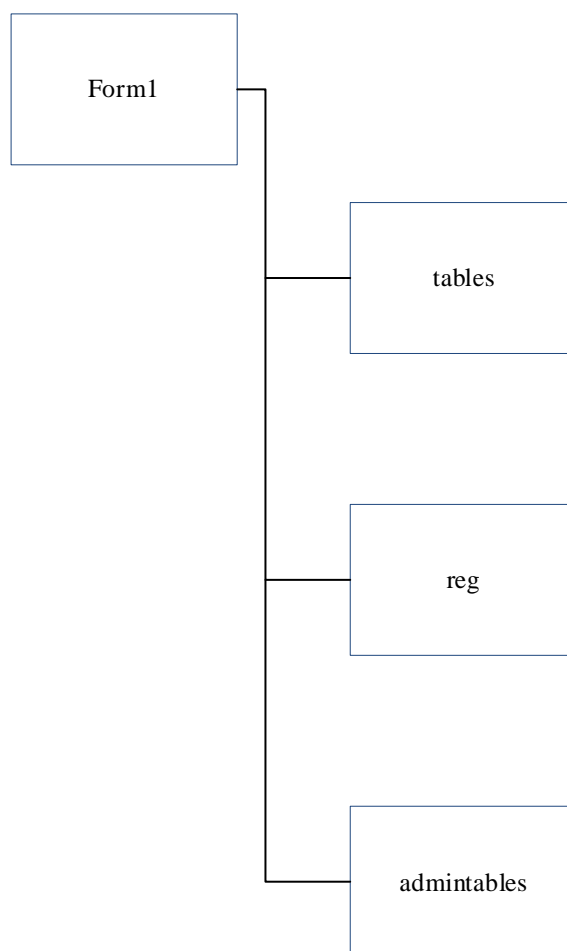


Рисунок 2.1 – Структура программы

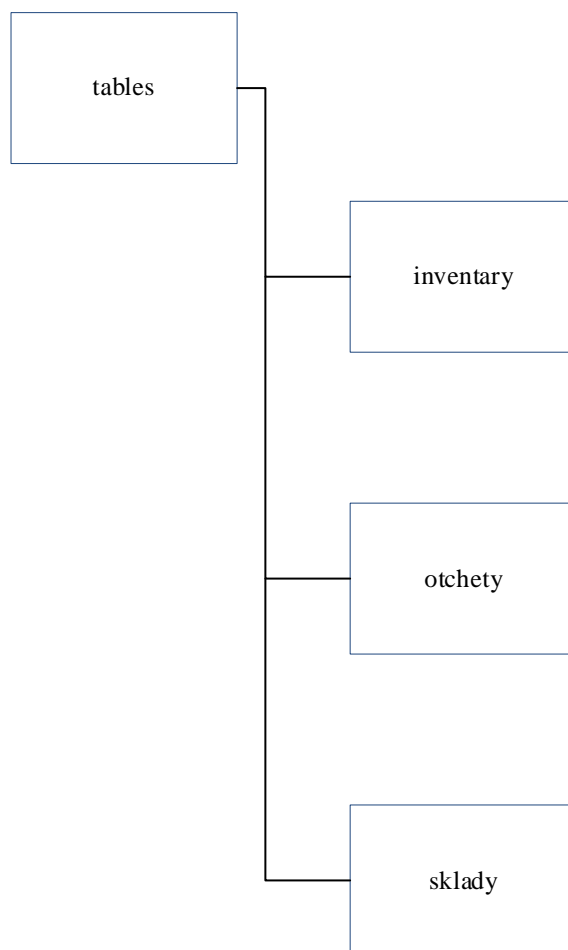


Рисунок 2.2 – Структура программы

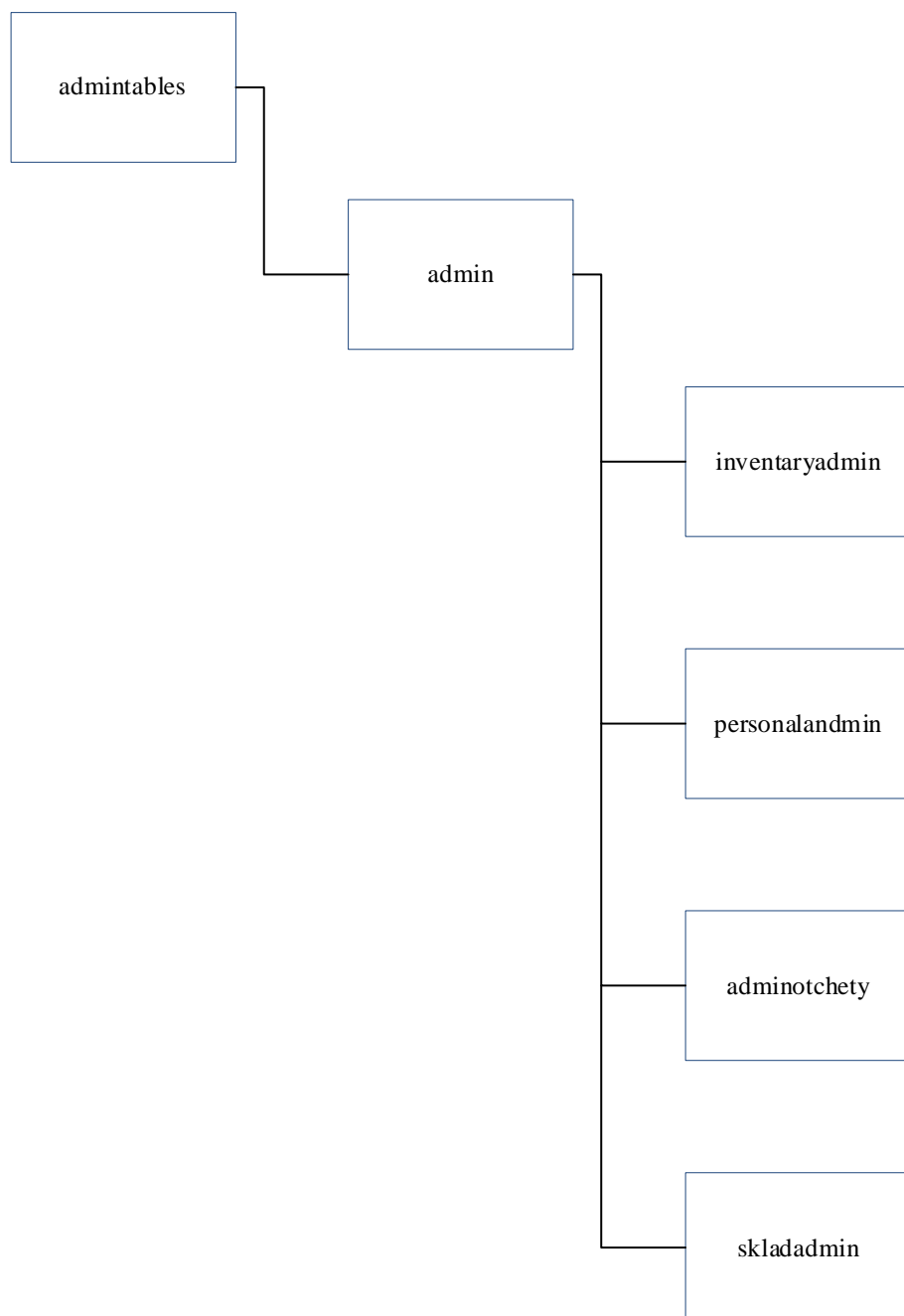


Рисунок 2.3 – Структура программы

Формы, используемые в основной программе, приведены в таблице 2.1.

Таблица 2.1 – Формы, используемые в основной программе

Форма	Назначение
Form1	Форма авторизации для пользователя
reg	Форма регистрации
tables	Форма выбора таблиц для пользователя
inventory	Пользовательская форма для работы с таблицей «инвентарь»
sklady	Пользовательская форма для работы с таблицей «склад»
otchety	Пользовательская форма для работы с таблицей «отчёты»
admintables	Форма авторизации для администратора
admin	Форма выбора таблиц для администратора
inventoryadmin	Форма администратора для работы с таблицей «инвентарь»
personaladmin	Форма администратора для работы с таблицей «персонал»
skladadmin	Форма администратора для работы с таблицей «склад»
adminotchety	Форма администратора для работы с таблицей «отчёты»

3 Схема данных

Для создания базы данных склада нужно знать данные: о предмете, находящемся в распоряжении отдела, количество предметов, его дату поставки и цену. Конечным этапом работы программы будет таблица со всеми предметами, его расположением, в виде таблиц.

Исходя из этого, можно составить контекстную диаграмму, которая представлена на рисунке 3.1.

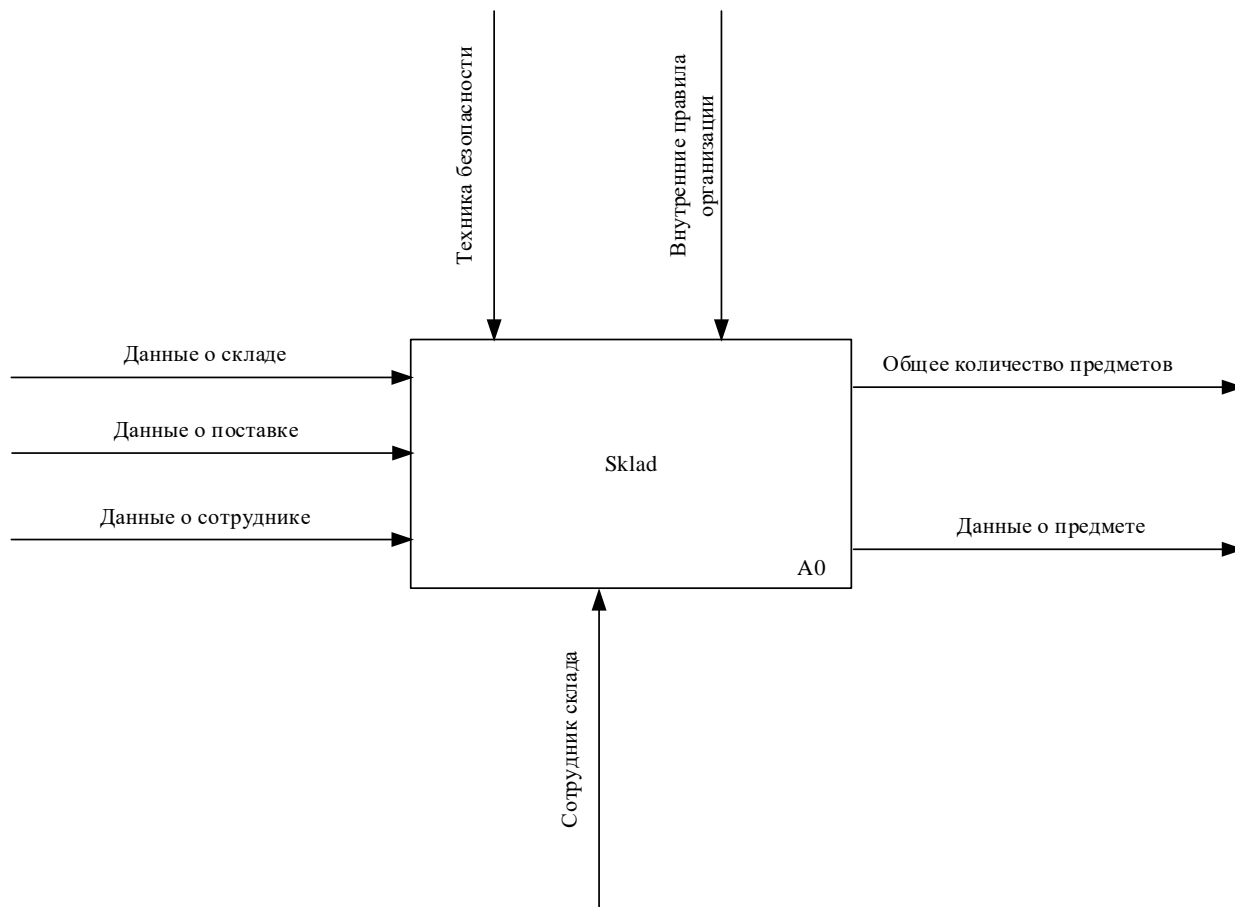


Рисунок 3.1 – Контекстная диаграмма

Можно выделить три основные сущности: отчёты о поставках, инвентарь, склад. Данные сущности представлены на рисунке 3.2.

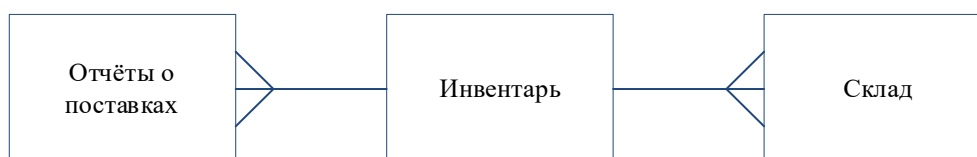


Рисунок 3.2 – Основные сущности базы данных

Логическая схема (рисунок 3.3).

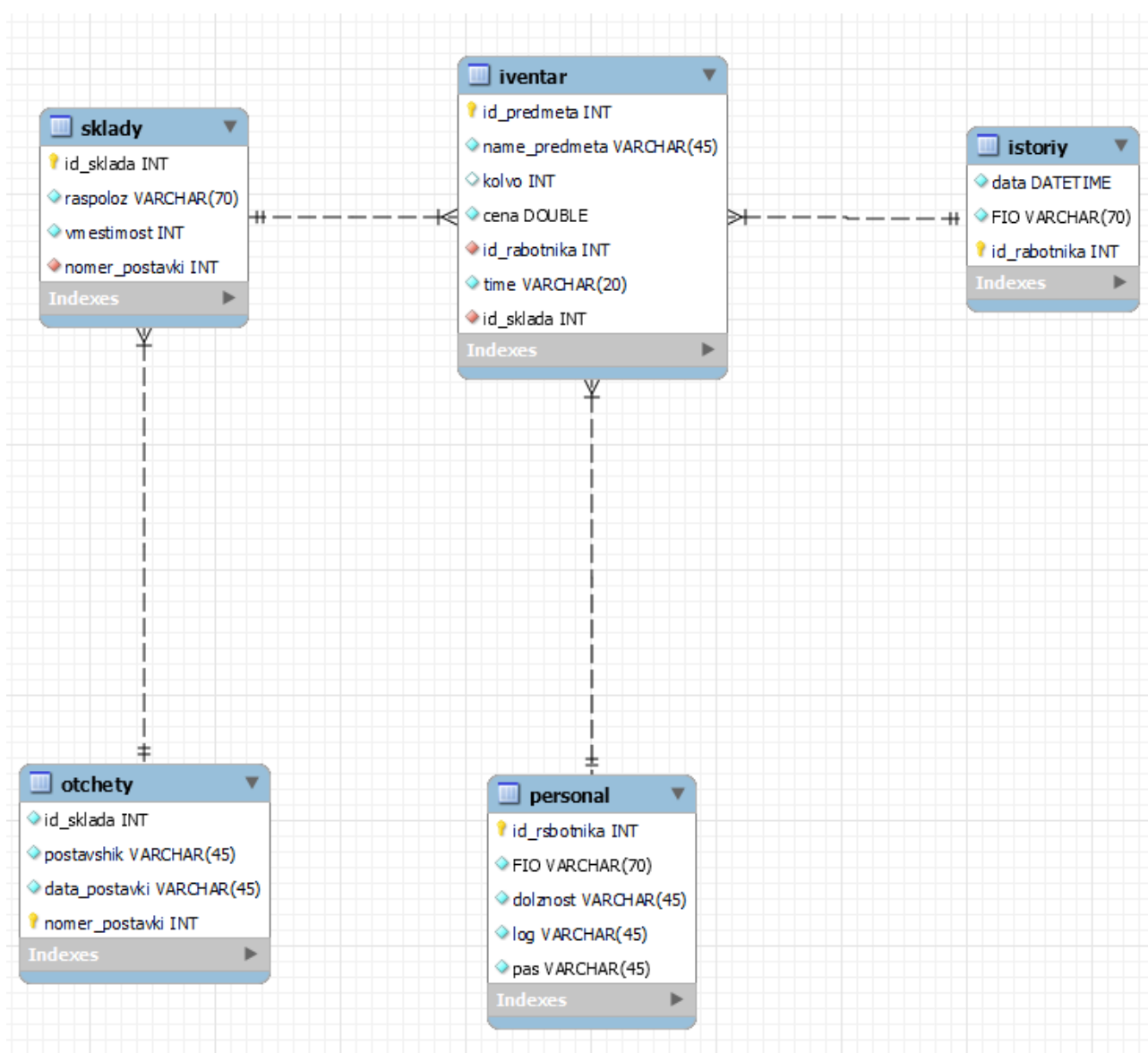


Рисунок 3.3 – Логическая схема

Структура таблиц базы данных представлена в таблицах 3.1 – 3.5.

Таблица 3.1 – Таблица «istoriy»

Идентификатор	Описание	Тип данного	Размер
data	Дата входа	Дата	–
ФИО	ФИО входившего	Текстовый	70
id_rabotnika(PK)	ID входившего	Числовой	-

Таблица 3.2 – Таблица «iventar»

Идентификатор	Описание	Тип данного	Размер
id_predmeta(PK)	ID предмета	Числовой	–
name_predmeta	Наименование предмета	Текстовый	45
kolvo	Количество предметов	Числовой	-
cena	Цена предмета	Денежный	-
id_rabotnika(FK)	ID работника	Числовой	-
time	Время добавления предмета	Дата	-
id_sklada(FK)	ID склада в котором храниться предмет	Числовой	-

Таблица 3.3 – Таблица «otchety»

Идентификатор	Описание	Тип данного	Размер
id_sklada(FK)	ID склада куда была поставка	Числовой	–
postavshik	Поставщик	Текстовый	45
data_postavki	Дата поставки	Дата	-
nomer_postavki(PK)	Номер поставки	Числовой	-

Таблица 3.4 – Таблица «personal»

Идентификатор	Описание	Тип данного	Размер
id_rsbotnika(PK)	ID работника	Числовой	–
FIO	ФИО работника	Текстовый	45
dolznost	Должность работника	Тестовый	45
log	Логин работника	Тестовый	45
pas	Пароль работника	Тестовый	45

Таблица 3.5 – Таблица «sklady»

Идентификатор	Описание	Тип данного	Размер
id_sklada	ID склада	Числовой	–
raspoloz	Месторасположение склада	Текстовый	70
vmestimost	Оставшееся свободное место	Числовой	-
nomer_postavki	Номера поставок	Числовой	-

4 Оптимизация программы

Оптимизация программы – это процесс модификации кода программы с целью улучшения её эффективности.

Основной задачей программирования является создание правильных, а не эффективных программ. Зачастую для повышения производительности программ большую значимость имеют такие факторы как:

четко выработанные требования к программе до начала ее разработки;

- хорошая архитектура программы;

- хорошо спроектированные модули;

- правильно выбранный компилятор со встроенными средствами оптимизации.

Ввиду данных факторов основной целью разработки являлось создание правильной, хорошо спроектированной программы. В большинстве случаев к оптимизации следует приступать только после того, как программа создана и выдает требуемые результаты. В противном случае можно получить не просто неудобочитаемый код и неработающую программу, но еще и неоптимизированное решение. Однако неизвестно, насколько эффективной будет программа, оптимизированная после ее разработки, поэтому определять требования к эффективности следует на стадии проектирования.

Несмотря на вышеперечисленные факты, оптимизация оказывается очень полезной, когда программа работает правильно. Производя небольшие изменения в некоторых фрагментах кода, называемых критическими областями, оптимизация помогает повысить эффективность программы в несколько раз.

Так как современные ЭВМ отличаются высоким быстродействием, то оптимизация отдельных, редко встречающихся операторов является бесполезной тратой времени. Однако повышение эффективности на мощных машинах можно получить за счет правильной оптимизации циклов и операторов, находящихся внутри тела циклов.

Программа, разрабатываемая в ходе практической работы, была оптимизирована по памяти. Программа не имеет постоянного подключения к базе данных, а подключается к ней только когда необходимо произвести манипуляции с данными из базы данных, после чего отключается от неё.

Вышеприведённый приём помог существенно повысить эффективность работы программы.

5 Тестирование программы

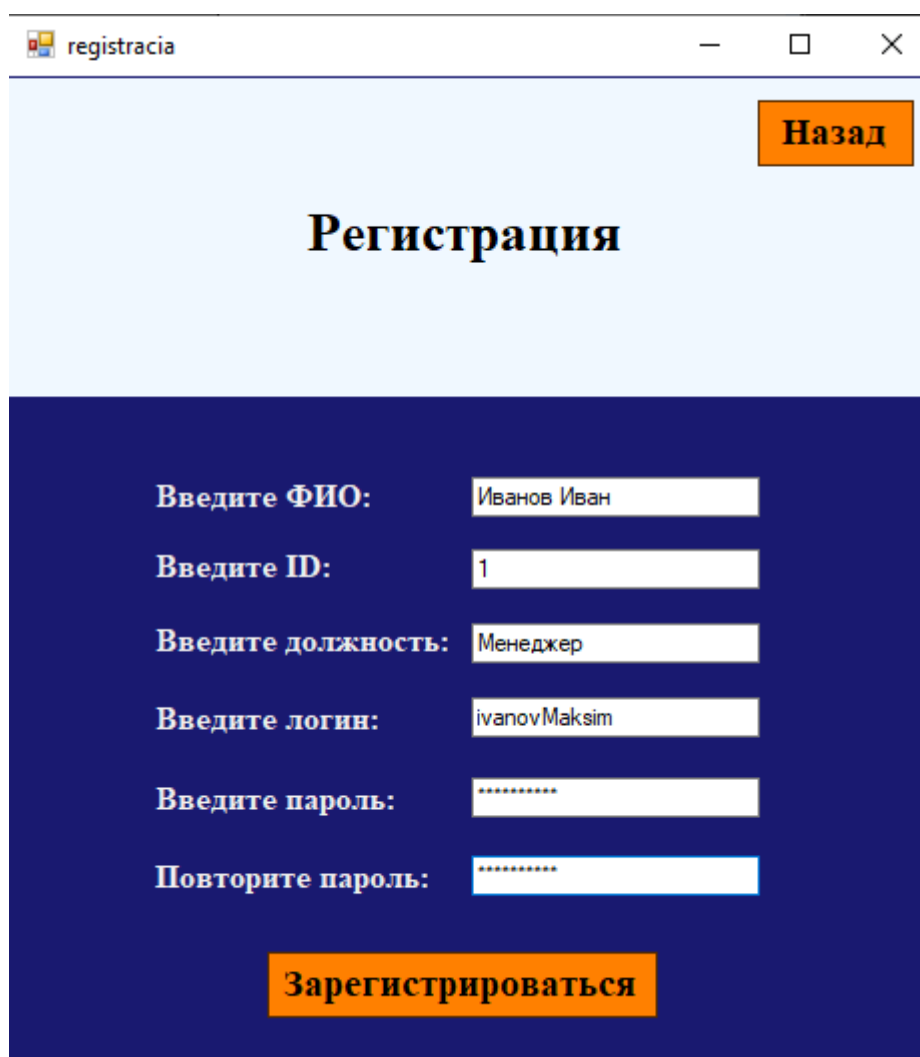
Тестирование — это набор процедур и действий, предназначенных для демонстрации правильности работы программы в заданных режимах и внешних условиях. Цель тестирования — выявить наличие ошибок или убедительно продемонстрировать их отсутствие. Процесс тестирования проходит в 3 этапа:

- проверка поведения программы в нормальных условиях;
- проверка поведения программы в экстремальных условиях;
- проверка поведения программы в исключительных ситуациях.

Каждый из этапов предполагает задание определенного, характерного для данного этапа набора данных.

5.1 Тестирование программы в нормальных условиях

Тестирование в нормальных условиях было проведено со всеми формами. Были протестированы переходы от формы к форме, корректная выборка данных из базы данных, автоматическое обновление форм. Особое внимание было уделено проверке регистрации нового пользователя, при котором обязательна проверка введенного пароля на соответствие требованиям безопасности. Тестирование процедуры регистрации нового пользователя представлено на рисунках 5.1, 5.2, 5.3.



The image shows a web browser window titled 'registracia'. The page has a light blue header with a 'Назад' (Back) button in an orange box. The main content area has a dark blue background. It contains a registration form with the following fields and values:

Field Label	Value
Введите ФИО:	Иванов Иван
Введите ID:	1
Введите должность:	Менеджер
Введите логин:	ivanovMaksim
Введите пароль:	*****
Повторите пароль:	*****

At the bottom of the form is an orange button labeled 'Зарегистрироваться' (Register).

Рисунок 5.1 – Окно «Регистрация»

Form1

Авторизация

Введите логин: ivamovMaksim

Введите пароль: *****

Войти

Зарегистрироваться

Я администратор

Рисунок 5.2 – Окно «Авторизация»

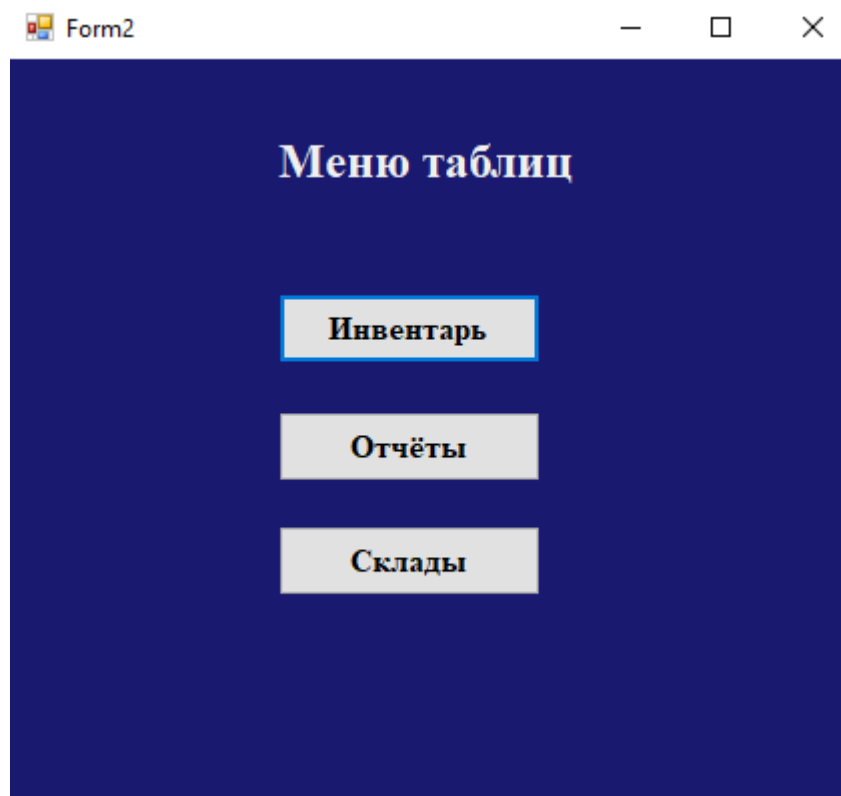


Рисунок 5.3 – Главное меню

Во время тестирования программы в нормальных условиях ошибок не выявлено, так как форма с главным меню, представленная на рисунке 5.3, появляется только в случае правильности ввода данных. Тестирование было подтверждено визуально, и оно показало, что программа работает правильно.

5.2 Тестирование программы в экстремальных условиях

Тестирование в экстремальных условиях было проведено со всеми формами. Была протестирована функция предотвращения ошибки при попытке добавить уже существующую запись в базу данных. Особое внимание было уделено проверке входа пользователя при котором обязательна проверка занятости логина и пароля. При попытке ввести в окно авторизации неправильный логин и пароль (рисунок 5.4) появляется сообщение об ошибке (рисунок 5.5).

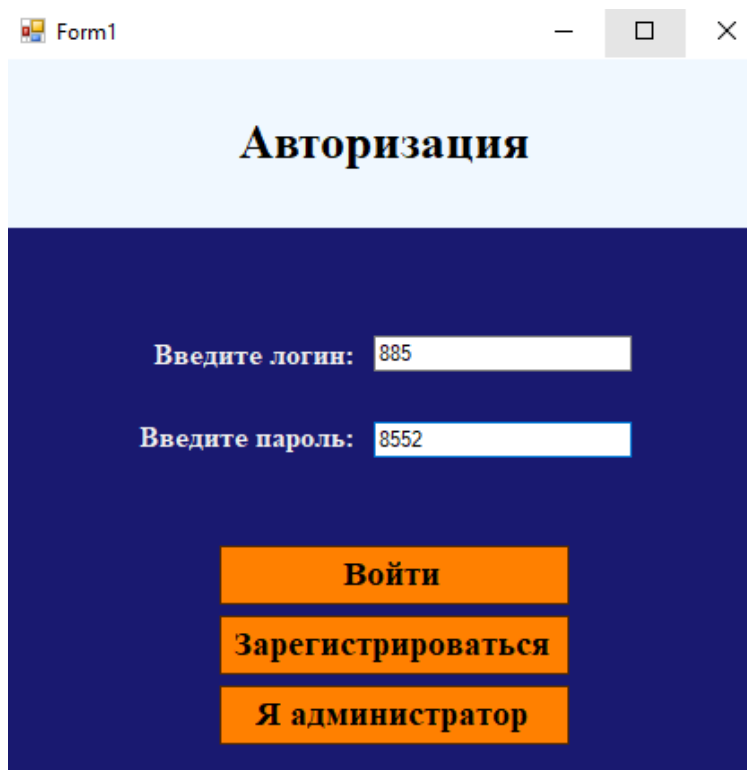


Рисунок 5.4 – Форма авторизации

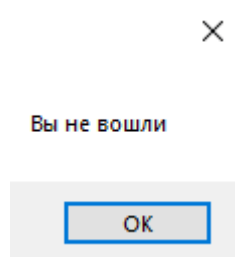


Рисунок 5.5 – Сообщение об ошибке

5.3 Тестирование программы в исключительных ситуациях

Тестирование в исключительных ситуациях было проведено со всеми формами. Была протестирована функция предотвращения ошибок при попытке отправить на сервер пустые поля. Особое внимание было уделено проверке входа пользователя при котором обязательна проверка полей. При попытке нажать на кнопку регистрации с пустыми полями ввода появляется сообщение об ошибке (рисунок 5.6).

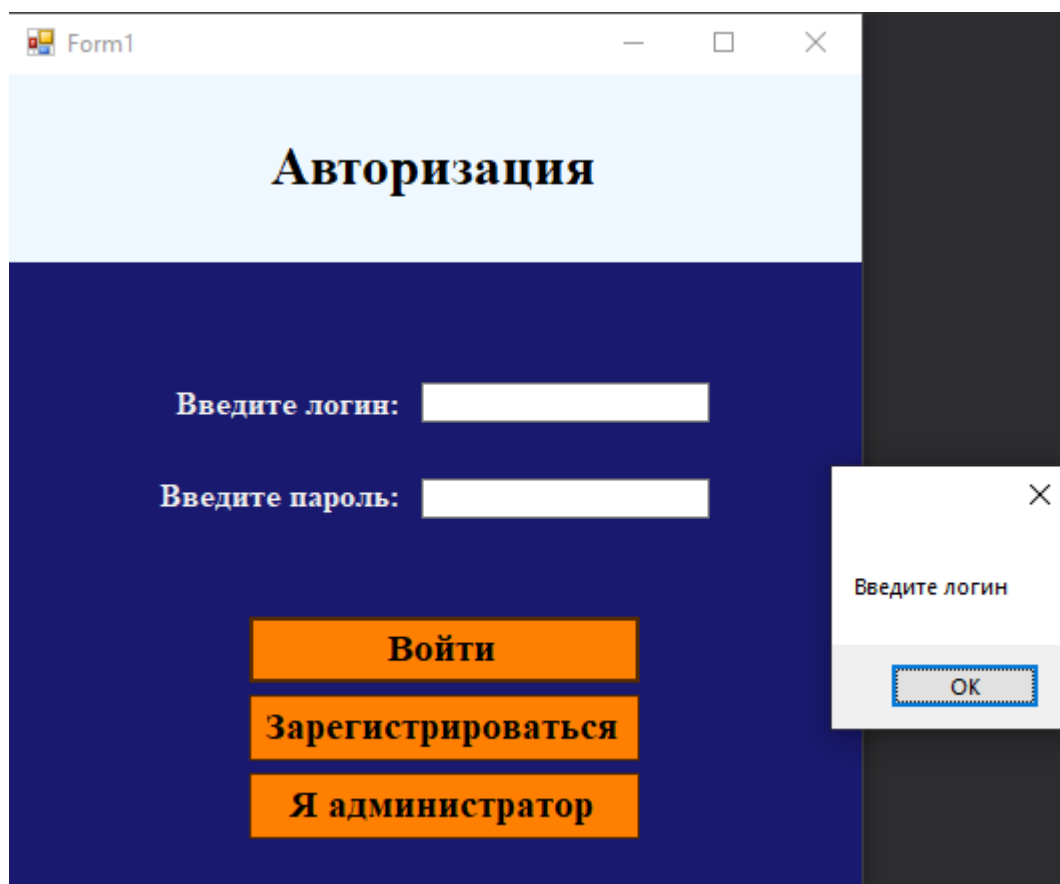


Рисунок 5.6 – Сообщение об ошибке

Так как тестирование в нормальных, экстремальных условиях и исключительных ситуациях не выявило наличие ошибок, можно сказать, что программа работает правильно. Листинг программы представлен в приложении А, а результаты программы – в приложении Б.

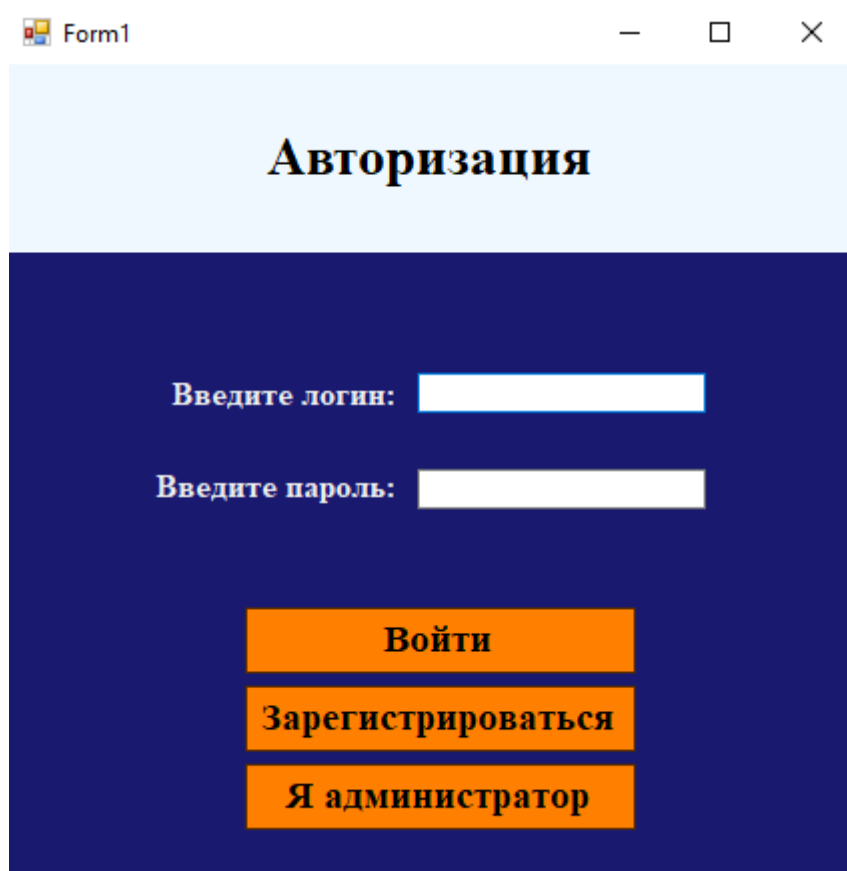
6 Руководство пользователя

Минимальные системные требования:

- Windows 7;
- процессор от 1,2 ГГц;
- ОЗУ от 64 МБ;
- свободное место на жестком диске 5 МБ;
- .NET Framework 4.5 и выше.

Для работы с программой следуйте пошаговому руководству, представленному ниже:

- 1) для запуска приложения необходимо запустить файл prog1.sln;
- 2) при запуске на экране появится форма для авторизации пользователя (рисунок 6.1) с возможностью переключения на форму регистрации (рисунок 6.2);



The image shows a screenshot of a Windows application window titled 'Form1'. The window contains a form for user authentication. The form has a light blue header bar with the word 'Авторизация' (Authorization) in bold black text. Below the header, the background is dark blue. There are two white input fields: the first is preceded by the text 'Введите логин:' (Enter login) and the second by 'Введите пароль:' (Enter password). At the bottom of the form, there are three orange buttons with black text: 'Войти' (Login), 'Зарегистрироваться' (Register), and 'Я администратор' (I am administrator).

Рисунок 6.1 – Форма авторизации

The image shows a web browser window with the title 'registracia'. The page has a light blue header with a 'Назад' (Back) button in an orange box. The main content area has a dark blue background. It features a registration form with the following fields and labels:

- Введите ФИО: (Text input)
- Введите ID: (Text input)
- Введите должность: (Text input)
- Введите логин: (Text input)
- Введите пароль: (Text input)
- Повторите пароль: (Text input)

At the bottom of the form is an orange button labeled 'Зарегистрироваться' (Register).

Рисунок 6.2 – Форма регистрации

- 3) после авторизации пользователь попадет в главное меню (рисунок 6.3), где можно будет перейти к нужной таблице;

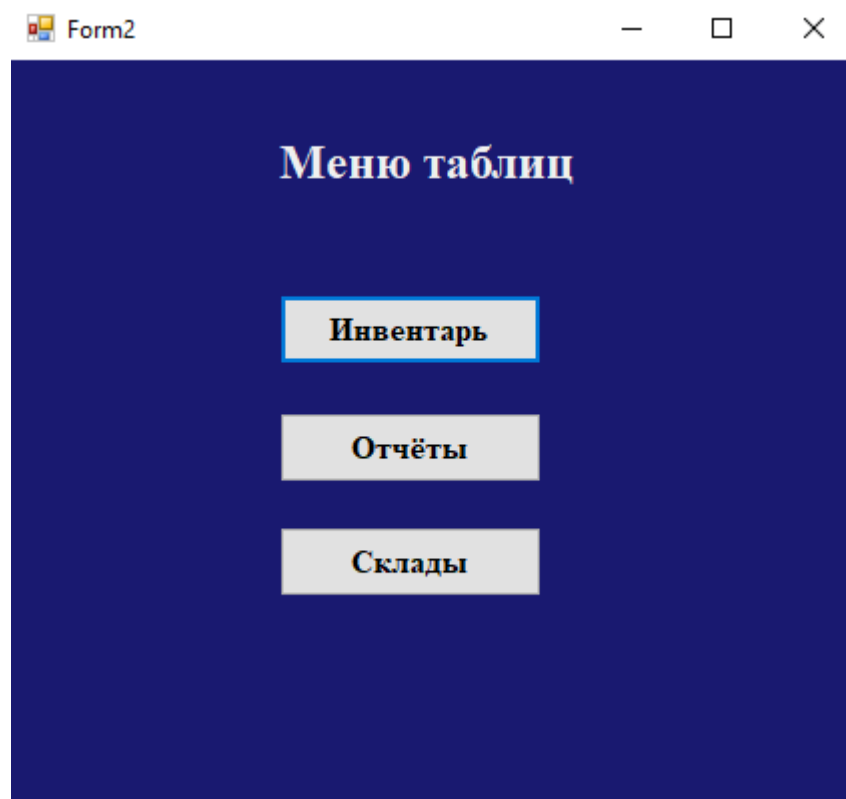


Рисунок 6.3 – Главное меню

- 4) при выборе одной из таблиц откроется соответствующее окно (рисунок 6.4);

The image shows a software window titled 'sklady'. It contains a table with the following data:

	id_sklada	raspoloz	vmestimost	nomer_postavki
▶	1	1	1	1
	3	4	4	4
*				

The first row of the table is highlighted in blue. Below the table, there is a large gray rectangular area. At the bottom right of the window, there is a button labeled 'Назад' (Back).

Рисунок 6.4 – Таблица «sklad»

5) после авторизации в роли администратора пользователь попадет в главное меню (рисунок 6.5), где можно будет перейти к нужной таблице;

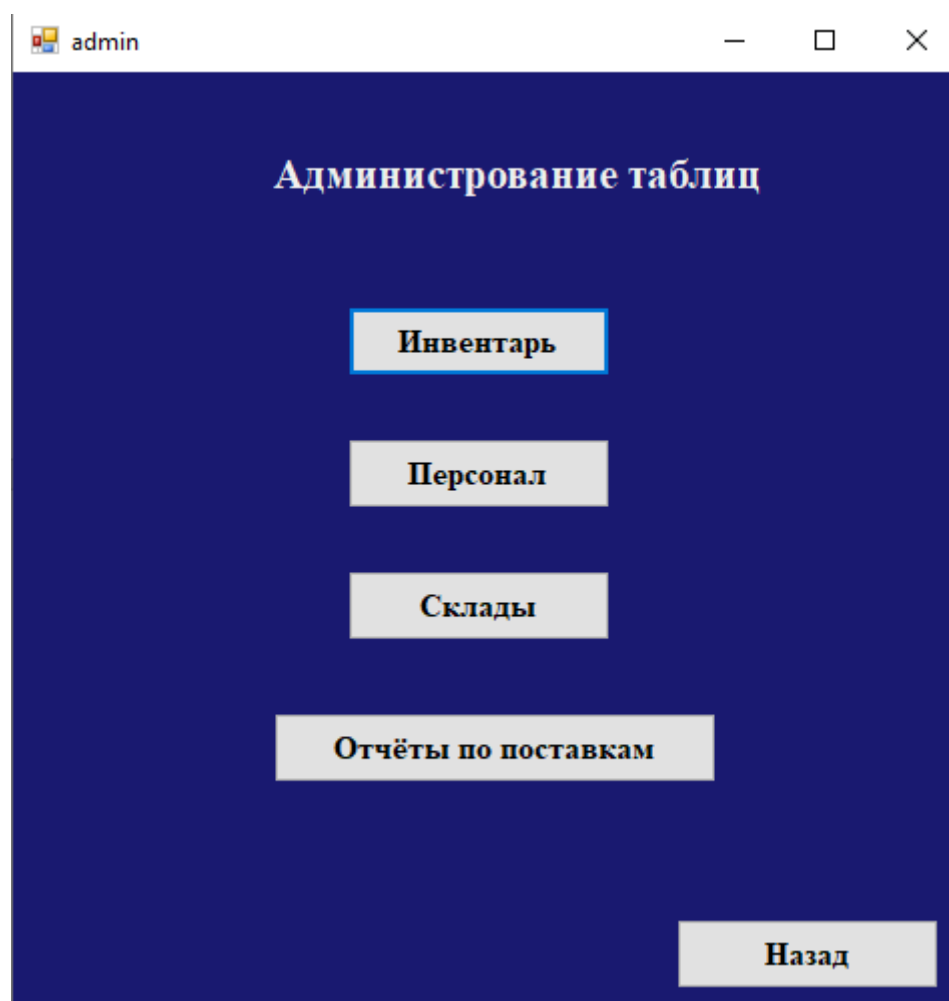


Рисунок 6.5 – Главное администратора

6) при выборе одной из таблиц откроется соответствующее окно (рисунок 6.6);

personaladmin

	id_rsbтника	FIO	dolznost	log	pas
▶	1	ivan	sklad	1	1
	2	admin	admin	1	1
	312	Мик	mi	Mik	SuckmeDuck 1
*					

Редактирование

Id работника

ФИО

Должность

Редактировать

Удаление и поиск

Поиск

Удаление

Назад

Рисунок 6.6 – Таблица «personal»

7) для добавления записей в таблицу необходимо заполнить соответствующие поля для ввода данных и нажать кнопку “Добавить” (рисунок 6.7);

Добавление и редактирование

Id предмета

Название предмета

Коллическтво

Цена

id работника

19 декабря 20: ▾ Дата поставки

id склада

Добавить Редактировать

Рисунок 6.7 – Добавление данных

8) при нажатии на кнопку “Добавить” появится соответствующее сообщение (рисунок 6.8);

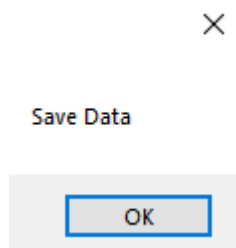


Рисунок 6.8 – Сообщение об успешном добавлении

9) при наличии пустых полей программа потребует их заполнения (рисунок 6.9);

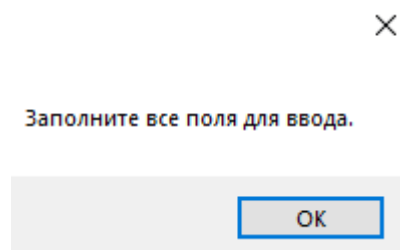


Рисунок 6.9 – Сообщение о наличии незаполненных полей для ввода

10) для поиска нужных данных необходимо ввести данные в соответствующей графе (рисунок 6.10);



Рисунок 6.10 – Поиск данных

11) после нажатия кнопки «поиск» будут выделены результаты поиска (рисунок 6.11);

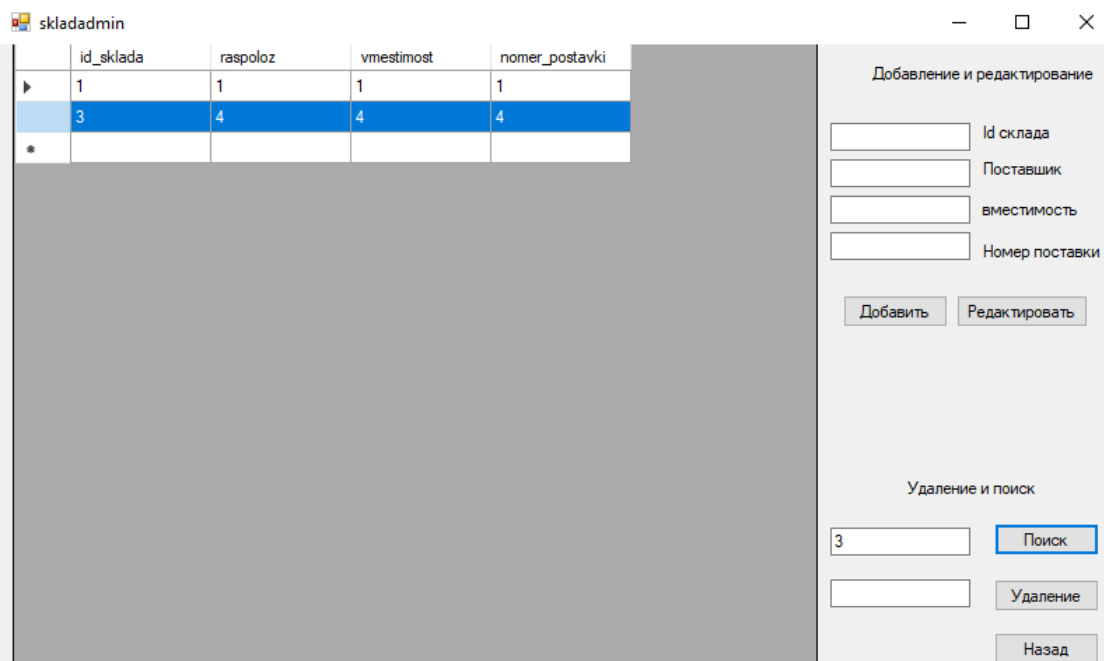


Рисунок 6.11 – Результаты поиска

- 12) для обновления данных необходимо заполнить поля для ввода данных, ввести id редактируемой строки в специальное поле и кликнуть на кнопку “Редактировать по ID” (рисунок 6.12);

Добавление и редактирование

Id склада
 Поставщик
 вместимость
 Номер поставки

Рисунок 6.12 – Обновление данных

- 13) после нажатия кнопки «Редактировать по ID» появится сообщение (рисунок 6.11);

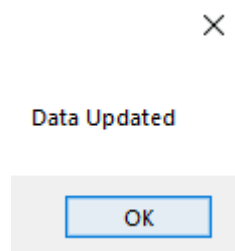


Рисунок 6.13 – Сообщение об успешном обновлении данных

- 14) при нажатии кнопки «Удаление» появится сообщение об успешном удалении (рисунок 6.14);



Рисунок 6.12 – Удаление данных

- 15) при нажатии кнопки «Удаление» появится сообщение об успешном удалении (рисунок 6.15);

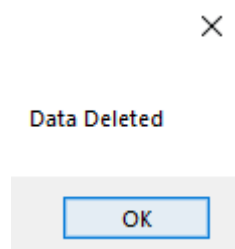


Рисунок 6.15 – Сообщение об успешном удалении

- 16) при нажатии кнопки «Назад» откроется главное меню (рисунок 6.16);

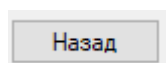


Рисунок 6.16 – Кнопка «Назад»

- 17) для выхода из программы нажмите кнопку «X» в любой форме приложения (рисунок 6.17).



Рисунок 6.17 – Кнопка закрытия приложения

ЗАКЛЮЧЕНИЕ

Основная задача практической работы – создание информационной системы для работы с базой данных. Поставленная задача была полностью выполнена в процессе реализации данного проекта.

В ходе создания данной программы были получены практические навыки в области проектирования программы для базы данных. Опыт, полученный при написании работы подтвердил необходимость в первую очередь выработки правильных требований к программе и грамотного подхода к её проектированию перед началом написания кода.

Также были получены практические навыки в области оптимизации и тестирования программы.

Результатом проведенной работы является правильно спроектированное приложение, а также работающий, эффективный, отлаженный и корректно отвечающий тестам проект.

При разработке программы были изучены и применены такие элементы управления как: TextBox, DataGridView и другие.

Программа имеет надлежащую структуру и необходимые комментарии.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Верзух Э. [пер. с англ. О. Л. Пелявского] Управление проектами: ускоренный курс по программе MBA / М., С-П: Диалектика, 2017
- 2) Грашина М. Н. Основы управления проектами / М.: БИНОМ. Лаборатория знаний, 2015
- 3) Желязны Д. Говори на языке диаграмм. – М: Манн, Иванов и Фербер, 3-е издание, 2016
- 4) Иванов П. В. Управление проектами / учеб. Пособие / Издательство: Феникс, Ростов-на-Дону, 2016
- 5) Кремлев Н.Д. Управление инновационными проектами / Учебное пособие М.: НИЦ ИНФРА-М, 2017
- 6) Лич Л. [пер. с англ. У. Саламатовой] Вовремя и в рамках бюджета: Управление проектами по методу критической цепи / М.: Альпина Паблишер, 2016
- 7) Мороз О. А. Управление проектами в ProjectLibre [учебно-практическое пособие] / Издательство: Феникс, Ростов-на-Дону, 2015
- 8) Ньютон Р. Управление проектами от А до Я [Пер. с англ.] / М.: Альпина Паблишер, 2016
- 9) Петин В. А. Проекты с использованием контроллера Arduino / С-П: БХВ-Петербург, 2017
- 10) Снедакер С. Управление IT-проектом, или Как стать полноценным СЮ [пер. с англ.] / М.: ДМК Пресс, 2016
- 11) Тихомирова О. Г. Управление проектом: комплексный подход и системный анализ / Монография М.: ИНФРА-М, 2017
- 12) Тихомирова О. Г. Управление проектами: практикум / Учебное пособие / М.: Инфра-М, 2017
- 13) Управление проектом: основы проектного управления / учебник для студентов вузов под. ред. М. Л. Разу /М.: КноРус, 2018
- 14) Управление проектами: учебник и практикум для среднего профессионального образования / А. И. Балашов, Е. М. Рогова, М. В. Тихонова,

Е. А. Ткаченко ; под общей редакцией Е. М. Роговой. — Москва : Издательство Юрайт, 2019

15) Управление проектом в сфере графического дизайна [Пер. с англ.] / М.: Альпина Паблишер, 2016

16) Хелдман К [Пер. с англ. А. В. Шаврина] Профессиональное управление проектом / М.: Лаборатория знаний, Москва, 2016

17) Шаврин А. В. Руководство по управлению проектами на основе стандарта ISO 21500 / М.: Лаборатория знаний, 2017

18) Краткая документация по ProjectLibre / [Электронный ресурс] – Режим доступа: <http://kostigoff.ru/projectlibre.15/> (проверено 19.08.2017)

19) ООО «Спутникс» <https://sputnix.ru/ru/>

20) Открытая замена коммерческому продукту Microsoft Project / ProjectLibre [Электронный ресурс] – Режим доступа: <http://www.projectlibre.com/> (проверено 19.08.2017)

21) Основы управления проектами / [Электронный ресурс] – Режим доступа: <http://www.studfiles.ru/preview/6325721/> (проверено 19.08.2017)

22) Разработка проекта / [Электронный ресурс] – Режим доступа: <http://www.studfiles.ru/preview/2905944/page:3/> (проверено 19.08.2017)

23) Создание проекта в ProjectLibre, видеоурок / [Электронный ресурс] – Режим доступа: <https://www.youtube.com/watch?v=aTuCTUjZ5WM> (проверено 19.08.2017)

24) Управление проектами, видеоуроки / [Электронный ресурс] – Режим доступа: <https://www.youtube.com/watch?v=F7yMJos3SRk&app=d..> (проверено 19.08.2017)

25) ЭБС biblio-online.ru

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программы

Листинг программы «Практика»

/*

Практическая работа по профессиональному модулю ПМ.02 «Осуществление интеграции программных модулей».

Название: «Практика».

Разработал: Бобылёв Сергей Сергеевич.

Группа: ТИП-61.

Дата: 12.06.2021год.

Краткое описание: данная программа является информационной системой для отдела ТСО.

Задание:

а) Создать базу данных, используя наиболее подходящую к выданному заданию платформу (MySQL / Microsoft SQL Server).

б) Создать приложение, используя наиболее приемлемую для решения задачи платформу: .NET (или Java).

в) Создать форму для авторизации зарегистрированных пользователей.

г) В системе, в зависимости от предметной области должны быть выделены не менее 2 различных типов пользователей.

д) Пользователи должны входить в систему, используя форму Авторизации:

1) Для успешной авторизации необходимо ввести свой логин и пароль.

2) При попытке входа введенные данные проверяются на совпадение в базе данных. Должно выдаваться сообщение об ошибке в случае неправильного ввода связки логин/пароль.

3) После успешного входа пользователя в зависимости от его роли происходит перенаправление на конкретные формы, предназначенные для работы этого типа пользователя.

4) При нажатии на кнопку "Выход" в любом окне/форме системы, пользователи должны быть отключены от системы и совершен переход на Форму авторизации.

5) Необходимо предусмотреть возможность регистрации Новых пользователей определенной роли. Поэтому нужно добавить в систему форму регистрации. При регистрации пароль должен отвечать следующим требованиям:

- должен содержать от 6 до 16 символов;
- не должно быть символов из набора: * & { } | +;
- должны встречаться заглавные буквы;
- должны встречаться цифры.

е) Должны быть реализованы основные формы для работы с сущностями, реализованными в базе данных.

*/

//листинг начальной формы

```
using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void singup_Click(object sender, EventArgs e)
        {
            if (avtlog.Text == "")
            {
                MessageBox.Show("Введите логин");
                return;
            }
            if (avtpas.Text == "")
            {
                MessageBox.Show("Введите пароль");
                return;
            }
            string login = avtlog.Text;
            string password = avtpas.Text;

            db DB = new db();
            DataTable table = new DataTable();
```

```

        MySqlDataAdapter adapter = new MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("SELECT * FROM personal WHERE log =
@userlogin AND pas = @userpassword", DB.GetConnection());

        command.Parameters.Add("@userlogin", MySqlDbType.VarChar).Value = login;
        command.Parameters.Add("@userpassword", MySqlDbType.VarChar).Value = password;

        adapter.SelectCommand = command;
        adapter.Fill(table);

        if (table.Rows.Count > 0)
        {
            MessageBox.Show("Вы авторизовались");
            this.Hide();
            tables tables = new tables();
            tables.Show();
        }
        else
            MessageBox.Show("Вы не вошли");

    }

    private void zareg_Click(object sender, EventArgs e)
    {
        this.Hide();
        reg reg= new reg();
        reg.Show();
    }

    private void admin_Click(object sender, EventArgs e)
    {
        this.Hide();
        admintables admintables = new admintables();
        admintables.Show();
    }
}
}

```

//листинг формы регистрации

```

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class reg : Form
    {
        public reg()
        {
            InitializeComponent();
        }

        private void zareg_Click(object sender, EventArgs e)
        {

```

```

if (zaregfio.Text == "")
{
    MessageBox.Show("Введите ФИО");
    return;
}
if (zaregid.Text == "")
{
    MessageBox.Show("Введите ID");
    return;
}
if (zaregdolz.Text == "")
{
    MessageBox.Show("Введите вашу должность");
    return;
}
if (zareglog.Text == "")
{
    MessageBox.Show("Введите логин");
    return;
}
if (zaregpas.Text == "")
{
    MessageBox.Show("Введите пароль");
    return;
}
if (zaregpas.Text != repitpas.Text)
{
    MessageBox.Show("Вы повторили пароль не верно");
    return;
}

if (zaregpas.Text == zareglog.Text)
{
    MessageBox.Show("Ошибка - Логин не может быть равен паролю");
    zareglog.Text = "";
    zaregpas.Text = "";
    return;
}

if (zaregpas.MaxLength < 20 && zareglog.MaxLength < 20)
{
    MessageBox.Show("Ошибка - Логин или пароль больше 20 символов");
    zareglog.Text = "";
    zaregpas.Text = "";
    return;
}

if (zaregpas.Text.Length < 5 && zareglog.Text.Length < 5)
{
    MessageBox.Show("Ошибка - Логин или пароль меньше 5 символов");
    zareglog.Text = "";
    zaregpas.Text = "";
    return;
}

int p = 0;
for (int i = 0; i < zaregpas.Text.Length; i++)
{
    if (char.IsLower(zaregpas.Text[i]))
    {
        p = 1;
    }
}

```

```

        if (p == 0)
        {
            MessageBox.Show("Должна быть хотя одна маленькая буква");
            zareglog.Text = "";
            zaregpas.Text = "";
            return;
        }
        {
            int l = 0;
            for (int i = 0; i < zaregpas.Text.Length; i++)
            {
                if (char.IsUpper(zaregpas.Text[i]))
                {
                    l = 1;
                }
            }
            if (l == 0)
            {
                MessageBox.Show("Должна быть хотя одна заглавная буква");
                zareglog.Text = "";
                zaregpas.Text = "";
                return;
            }
        }
    }
}

```

```

if (checkUser())
    return;

```

```

    db DB = new db();
    MySqlCommand command = new MySqlCommand ("INSERT INTO personal (id_rsbotnika,
FIO, dolznost, log, pas ) VALUES (@id, @fio, @dolz, @login, @pass)", DB.GetConnection());
    command.Parameters.Add("@fio", MySqlDbType.VarChar).Value = zaregfio.Text;
    command.Parameters.Add("@id", MySqlDbType.VarChar).Value = zaregid.Text;
    command.Parameters.Add("@dolz", MySqlDbType.VarChar).Value = zaregdolz.Text;
    command.Parameters.Add("@login", MySqlDbType.VarChar).Value = zareglog.Text;
    command.Parameters.Add("@pass", MySqlDbType.VarChar).Value = zaregpas.Text;

```

```

    DB.openConnection();

```

```

    if (command.ExecuteNonQuery() == 1)
    {
        MessageBox.Show("Аккаунт создан");
        this.Hide();
        Form1 Form1 = new Form1();
        Form1.Show();
    }
    else
        MessageBox.Show("Аккаунт не создан");

```

```

    DB.closeConnection();

```

```

}
public Boolean checkUser()
{

```

```

    db DB = new db();

    DataTable table = new DataTable();

```

```

        MySqlDataAdapter adapter = new MySqlDataAdapter();

        MySqlCommand command = new MySqlCommand("SELECT * FROM personal WHERE
id_rsbotnika = @id", DB.GetConnection());
        command.Parameters.Add("@id", MySqlDbType.VarChar).Value = zaregid.Text;

        adapter.SelectCommand = command;
        adapter.Fill(table);

        if (table.Rows.Count > 0)
        {
            MessageBox.Show("Данный ID уже занят");
            return true;
        }
        else
            return false;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        Form1 Form1 = new Form1();
        Form1.Show();
    }
}

```

//листинг формы входа для администратора

```

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class admintables : Form
    {
        public admintables()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (adminlog.Text == "")
            {
                MessageBox.Show("Введите логин");
                return;
            }
            if (adminpass.Text == "")
            {
                MessageBox.Show("Введите пароль");
                return;
            }
        }
    }
}

```

```

    }
    string login = adminlog.Text;
    string password = adminpass.Text;
    string dolz = "admin";

    db DB = new db();
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT * FROM personal WHERE log =
@userlogin AND pas = @userpassword AND dolznost = @dolznost", DB.GetConnection());

    command.Parameters.Add("@userlogin", MySqlDbType.VarChar).Value = login;
    command.Parameters.Add("@userpassword", MySqlDbType.VarChar).Value = password;
    command.Parameters.Add("@dolznost", MySqlDbType.VarChar).Value = dolz;

    adapter.SelectCommand = command;
    adapter.Fill(table);

    if (table.Rows.Count > 0)
    {
        MessageBox.Show("Вы авторизовались");
        this.Hide();
        admin admin = new admin();
        admin.Show();
    }
    else
        MessageBox.Show("Вы не вошли");

}

private void back_Click(object sender, EventArgs e)
{
    this.Hide();
    Form1 Form1 = new Form1();
    Form1.Show();
}

private void admintables_Load(object sender, EventArgs e)
{
}
}
}

```

//листинг формы выбора таблицы пользователя

```

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;

namespace производственная_практика
{
    public partial class tables : Form
    {
        public tables()

```



```

    {
        InitializeComponent();
    }

    private void inventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        inventory inventory = new inventory();
        inventory.Show();
    }

    private void otcity_Click(object sender, EventArgs e)
    {
        this.Hide();
        otchety otchety = new otchety();
        otchety.Show();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        sklady sklady = new sklady();
        sklady.Show();
    }

    private void tables_Load(object sender, EventArgs e)
    {
    }
}

```

//ЛИСТИНГ просмотра таблицы инвентарь от лица пользователя

```

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class inventory : Form
    {
        public inventory()
        {
            InitializeComponent();
        }

        private void back_Click(object sender, EventArgs e)
        {
            this.Hide();
            tables tables = new tables();
            tables.Show();
        }

        private void inventory_Load(object sender, EventArgs e)
        {

```

```

        db DB = new db();

        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("SELECT * FROM inventar ",
        DB.GetConnection());

        adapter.SelectCommand = command;
        adapter.Fill(table);
        otobraz.DataSource = table;

    }

}

}

//листинг просмотра таблицы

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class otchety : Form
    {
        public otchety()
        {
            InitializeComponent();
        }

        private void otchety_Load(object sender, EventArgs e)
        {
            db DB = new db();

            DataTable table = new DataTable();

            MySqlDataAdapter adapter = new MySqlDataAdapter();
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("SELECT * FROM otchety ",
            DB.GetConnection());

            adapter.SelectCommand = command;
            adapter.Fill(table);
            otobraz.DataSource = table;
        }

        private void back_Click(object sender, EventArgs e)
        {
            this.Hide();
            tables tables = new tables();
            tables.Show();
        }
    }
}

```

```

    }
}

//листинг просмотра таблицы склад от лица пользователя

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class sklady : Form
    {
        public sklady()
        {
            InitializeComponent();
        }

        private void sklady_Load(object sender, EventArgs e)
        {
            db DB = new db();

            DataTable table = new DataTable();

            MySqlDataAdapter adapter = new MySqlDataAdapter();
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("SELECT * FROM sklady ",
            DB.GetConnection());

            adapter.SelectCommand = command;
            adapter.Fill(table);
            otobraz.DataSource = table;
        }

        private void back_Click(object sender, EventArgs e)
        {
            this.Hide();
            tables tables = new tables();
            tables.Show();
        }
    }
}

//просмотр таблицы инвентарь от лица администратора

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class invetryadmin : Form
    {
        public invetryadmin()
        {
            InitializeComponent();

            private void invetryadmin_Load(object sender, EventArgs e)
            {
                db DB = new db();

                DataTable table = new DataTable();

                MySqlDataAdapter adapter = new MySqlDataAdapter();
                // Формирование запроса
                MySqlCommand command = new MySqlCommand("SELECT * FROM inventar ",
                DB.GetConnection());

                adapter.SelectCommand = command;
                adapter.Fill(table);
                otobraz.DataSource = table;
            }

            private void button2_Click(object sender, EventArgs e)
            {
                db DB = new db();
                try
                {
                    // Формирование запроса
                    MySqlCommand command = new MySqlCommand("insert into inventar
(id_predmeta,name_predmeta,kolvo,cena,id_rabotnika,time,id_sklada)
values(@id,@name,@kol,@cen,@rab,@data,@sklad);", DB.GetConnection());
                    command.Parameters.Add("@id", MySqlDbType.VarChar).Value = idsk.Text;
                    command.Parameters.Add("@name", MySqlDbType.VarChar).Value = post.Text;
                    command.Parameters.Add("@kol", MySqlDbType.VarChar).Value = textBox1.Text;
                    command.Parameters.Add("@cen", MySqlDbType.VarChar).Value = nomer.Text;
                    command.Parameters.Add("@rab", MySqlDbType.VarChar).Value = textBox2.Text;
                    command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
                    command.Parameters.Add("@sklad", MySqlDbType.VarChar).Value = textBox3.Text;

                    MySqlDataReader MyReader2;
                    DB.openConnection();
                    MyReader2 = command.ExecuteReader();
                    MessageBox.Show("Save Data");
                    while (MyReader2.Read())
                    {
                    }
                    DB.closeConnection();
                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
                // Добавление данных
                DataTable table = new DataTable();

                MySqlDataAdapter adapter = new MySqlDataAdapter();
                // Формирование запроса
                MySqlCommand command2 = new MySqlCommand("Select * From inventar ",
                DB.GetConnection());

```

```

        adapter.SelectCommand = command2;
        adapter.Fill(table);
        otobraz.DataSource = table;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        db DB = new db();
        try
        {
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("update inventar set name_predmeta = @name, kolvo = @kol, cena=@cen, id_rabotnika=@rab,time=@data, id_sklada=@sklad where id_predmeta = @id", DB.GetConnection());
            command.Parameters.Add("@id", MySqlDbType.VarChar).Value = idsk.Text;
            command.Parameters.Add("@name", MySqlDbType.VarChar).Value = post.Text;
            command.Parameters.Add("@kol", MySqlDbType.VarChar).Value = textBox1.Text;
            command.Parameters.Add("@cen", MySqlDbType.VarChar).Value = nomer.Text;
            command.Parameters.Add("@rab", MySqlDbType.VarChar).Value = textBox2.Text;
            command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
            command.Parameters.Add("@sklad", MySqlDbType.VarChar).Value = textBox3.Text;
            MySqlDataReader MyReader2;
            DB.openConnection();
            MyReader2 = command.ExecuteReader();
            MessageBox.Show("Data Updated");
            while (MyReader2.Read())
            {
            }
            DB.closeConnection();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        } // Редактирование данных
        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();
        // Формирование запроса
        MySqlCommand command2 = new MySqlCommand("Select * From inventar ",
        DB.GetConnection());

        adapter.SelectCommand = command2;
        adapter.Fill(table);
        otobraz.DataSource = table;
    }

    private void search_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < otobraz.RowCount; i++)
        {
            otobraz.Rows[i].Selected = false;
            for (int j = 0; j < otobraz.ColumnCount; j++)
            {
                if (otobraz.Rows[i].Cells[j].Value != null)
                {
                    if (otobraz.Rows[i].Cells[j].Value.ToString().Contains(idsearsh.Text))
                    {
                        otobraz.Rows[i].Selected = true;
                        break;
                    }
                }
            }
        }
    }

    private void del_Click(object sender, EventArgs e)
    {

```

```

        { //Проверка на наличие незаполненных полей
            if (iddel.Text == "")
            {
                MessageBox.Show("Заполните поле ID.");
                return;
            }
            db DB = new db();
            try
            {
                // Формирование запроса
                MySqlCommand command = new MySqlCommand("DELETE FROM inventar WHERE
id_predmeta=@nomer", DB.GetConnection());
                command.Parameters.Add("@nomer", MySqlDbType.Int32).Value = iddel.Text;
                MySqlDataReader MyReader2;
                DB.openConnection();
                MyReader2 = command.ExecuteReader();
                MessageBox.Show("Data Deleted");
                while (MyReader2.Read())
                {
                }
                DB.closeConnection();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            } // Удаление данных
            DataTable table = new DataTable();

            MySqlDataAdapter adapter = new MySqlDataAdapter();
            // Формирование запроса
            MySqlCommand command2 = new MySqlCommand("SELECT * FROM inventar ",
            DB.GetConnection());

            adapter.SelectCommand = command2;
            adapter.Fill(table);
            otobraz.DataSource = table;
        }
    }

    private void back_Click(object sender, EventArgs e)
    {
        this.Hide();
        admin admin = new admin();
        admin.Show();
    }
}

//листинг таблицы выбора таблиц от лица администратора
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;

namespace производственная_практика
{
    public partial class admin : Form
    {
        public admin()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void inventory_Click(object sender, EventArgs e)
    {
        this.Hide();
        invetryadmin invetryadmin = new invetryadmin();
        invetryadmin.Show();
    }

    private void personal_Click(object sender, EventArgs e)
    {
        this.Hide();
        personaladmin personaladmin = new personaladmin();
        personaladmin.Show();
    }

    private void otchet_Click(object sender, EventArgs e)
    {
        this.Hide();
        adminotchet adminotchet = new adminotchet();
        adminotchet.Show();
    }

    private void back_Click(object sender, EventArgs e)
    {
        this.Hide();
        Form1 Form1 = new Form1();
        Form1.Show();
    }

    private void sklady_Click(object sender, EventArgs e)
    {
        this.Hide();
        skladadmin skladadmin = new skladadmin();
        skladadmin.Show();
    }

    private void admin_Load(object sender, EventArgs e)
    {
    }
}
}

```

//листинг таблицы персонал от лица администратора

```

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class personaladmin : Form
    {
        public personaladmin()
        {
            InitializeComponent();
        }
    }
}

```

```

private void personaladmin_Load(object sender, EventArgs e)
{
    db DB = new db();

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command = new MySqlCommand("SELECT * FROM personal ",
    DB.GetConnection());

    adapter.SelectCommand = command;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void button1_Click(object sender, EventArgs e)
{
    db DB = new db();
    try
    {
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("update personal set FIO = @fio,
dolznost = @post where id_rsbotnika = @id", DB.GetConnection());
        command.Parameters.Add("@id", MySqlDbType.Int32).Value = idsk.Text;
        command.Parameters.Add("@fio", MySqlDbType.VarChar).Value = post.Text;
        command.Parameters.Add("@post", MySqlDbType.VarChar).Value = data.Text;

        MySqlDataReader MyReader2;
        DB.openConnection();
        MyReader2 = command.ExecuteReader();
        MessageBox.Show("Data Updated");
        while (MyReader2.Read())
        {
        }
        DB.closeConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } // Редактирование данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("Select * From personal ",
    DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void search_Click(object sender, EventArgs e)
{
    for (int i = 0; i < otobraz.RowCount; i++)
    {
        otobraz.Rows[i].Selected = false;
        for (int j = 0; j < otobraz.ColumnCount; j++)
            if (otobraz.Rows[i].Cells[j].Value != null)
                if
                    (otobraz.Rows[i].Cells[j].Value.ToString().Contains(idsearh.Text))
                {
                    otobraz.Rows[i].Selected = true;
                }
            }
    }
}

```



```

        break;
    }
}

private void del_Click(object sender, EventArgs e)
{
    //Проверка на наличие незаполненных полей
    if (id1.Text == "")
    {
        MessageBox.Show("Заполните поле ID.");
        return;
    }
    db DB = new db();
    try
    {
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("DELETE FROM personal WHERE
id_rsbotnika=@nomerpost", DB.GetConnection());
        command.Parameters.Add("@nomerpost", MySqlDbType.Int32).Value =
id1.Text;

        MySqlDataReader MyReader2;
        DB.openConnection();
        MyReader2 = command.ExecuteReader();
        MessageBox.Show("Data Deleted");
        while (MyReader2.Read())
        {
        }
        DB.closeConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    // Удаление данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("SELECT * FROM personal ",
DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void back_Click(object sender, EventArgs e)
{
    this.Hide();
    admin admin = new admin();
    admin.Show();
}
}

//листинг таблицы склад от лица администратора

using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class skladadmin : Form
    {
        public skladadmin()
        {
            InitializeComponent();
        }
        private void skladadmin_Load_1(object sender, EventArgs e)
        {
            db DB = new db();

            DataTable table = new DataTable();

            MySqlDataAdapter adapter = new MySqlDataAdapter();
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("SELECT * FROM sklady ",
            DB.GetConnection());

            adapter.SelectCommand = command;
            adapter.Fill(table);
            otobraz.DataSource = table;
        }

        private void search_Click_1(object sender, EventArgs e)
        {
            for (int i = 0; i < otobraz.RowCount; i++)
            {
                otobraz.Rows[i].Selected = false;
                for (int j = 0; j < otobraz.ColumnCount; j++)
                {
                    if (otobraz.Rows[i].Cells[j].Value != null)
                    {
                        if
                        (otobraz.Rows[i].Cells[j].Value.ToString().Contains(idsearsh.Text))
                        {
                            otobraz.Rows[i].Selected = true;
                            break;
                        }
                    }
                }
            }
        }

        private void button2_Click_1(object sender, EventArgs e)
        {
            db DB = new db();
            try
            {
                // Формирование запроса
                MySqlCommand command = new MySqlCommand("insert into sklady
(id_sklada,raspoloz,vmestimost,nomer_postavki) values(@id,@postav,@data,@nomerpost);",
            DB.GetConnection());
                command.Parameters.Add("@id", MySqlDbType.VarChar).Value = idsk.Text;
                command.Parameters.Add("@postav", MySqlDbType.VarChar).Value = post.Text;
                command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
                command.Parameters.Add("@nomerpost", MySqlDbType.VarChar).Value =
            nomer.Text;

                MySqlDataReader MyReader2;
                DB.openConnection();
                MyReader2 = command.ExecuteReader();
                MessageBox.Show("Save Data");
                while (MyReader2.Read())
                {

```

```

        }
        DB.closeConnection();

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } // Добавление данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("Select * From sklady ",
    DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void button1_Click_1(object sender, EventArgs e)
{
    db DB = new db();
    try
    {
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("update sklady set raspoloz = @id,
vmestimost = @post, nomer_postavki = @data where id_sklada = @nomer", DB.GetConnection());
        command.Parameters.Add("@id", MySqlDbType.Int32).Value = idsk.Text;
        command.Parameters.Add("@post", MySqlDbType.VarChar).Value = post.Text;
        command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
        command.Parameters.Add("@nomer", MySqlDbType.Int32).Value = nomer.Text;
        MySqlDataReader MyReader2;
        DB.openConnection();
        MyReader2 = command.ExecuteReader();
        MessageBox.Show("Data Updated");
        while (MyReader2.Read())
        {
        }
        DB.closeConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } // Редактирование данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("Select * From sklady ",
    DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void del_Click_1(object sender, EventArgs e)
{
    { //Проверка на наличие незаполненных полей
        if (iddel.Text == "")
        {
            MessageBox.Show("Заполните поле ID.");
            return;
        }
    }
}

```

```

        db DB = new db();
        try
        {
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("DELETE FROM sklady WHERE
id_sklada=@nomerpost", DB.GetConnection());
            command.Parameters.Add("@nomerpost", MySqlDbType.Int32).Value =
iddel.Text;

            MySqlDataReader MyReader2;
            DB.openConnection();
            MyReader2 = command.ExecuteReader();
            MessageBox.Show("Data Deleted");
            while (MyReader2.Read())
            {
            }
            DB.closeConnection();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        // Удаление данных
        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();
        // Формирование запроса
        MySqlCommand command2 = new MySqlCommand("SELECT * FROM sklady ",
DB.GetConnection());

        adapter.SelectCommand = command2;
        adapter.Fill(table);
        otobraz.DataSource = table;
    }

    private void back_Click_1(object sender, EventArgs e)
    {
        this.Hide();
        admin admin = new admin();
        admin.Show();
    }
}

//листинг таблицы отчёты от лица администратора
using MySql.Data.MySqlClient;
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;

namespace производственная_практика
{
    public partial class adminotch : Form
    {
        public adminotch()
        {
            InitializeComponent();
        }

        private void adminotch_Load(object sender, EventArgs e)
        {

```

```

        db DB = new db();

        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("SELECT * FROM otchety ",
        DB.GetConnection());

        adapter.SelectCommand = command;
        adapter.Fill(table);
        otobraz.DataSource = table;
    }

    private void back_Click(object sender, EventArgs e)
    {
        this.Hide();
        admin admin = new admin();
        admin.Show();
    }

    private void del_Click(object sender, EventArgs e)
    {
        { //Проверка на наличие незаполненных полей
            if (iddel.Text == "")
            {
                MessageBox.Show("Заполните поле ID.");
                return;
            }
        }
        db DB = new db();
        try
        {
            // Формирование запроса
            MySqlCommand command = new MySqlCommand("DELETE FROM otchety WHERE
            nomer_postavki=@nomer", DB.GetConnection());
            command.Parameters.Add("@nomer", MySqlDbType.Int32).Value = iddel.Text;
            MySqlDataReader MyReader2;
            DB.openConnection();
            MyReader2 = command.ExecuteReader();
            MessageBox.Show("Data Deleted");
            while (MyReader2.Read())
            {
            }
            DB.closeConnection();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        // Удаление данных
        DataTable table = new DataTable();

        MySqlDataAdapter adapter = new MySqlDataAdapter();
        // Формирование запроса
        MySqlCommand command2 = new MySqlCommand("SELECT * FROM otchety ",
        DB.GetConnection());

        adapter.SelectCommand = command2;
        adapter.Fill(table);
        otobraz.DataSource = table;
    }

    private void search_Click(object sender, EventArgs e)
    {
        for (int i = 0; i < otobraz.RowCount; i++)

```

```

        {
            otobraz.Rows[i].Selected = false;
            for (int j = 0; j < otobraz.ColumnCount; j++)
                if (otobraz.Rows[i].Cells[j].Value != null)
                    if
                        (otobraz.Rows[i].Cells[j].Value.ToString().Contains(idsearsh.Text))
                        {
                            otobraz.Rows[i].Selected = true;
                            break;
                        }
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    db DB = new db();
    try
    {
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("insert into otchety
(id_sklada,postavshik,data_postavki,nomer_postavki) values(@id,@postav,@data,@nomerpost);",
DB.GetConnection());
        command.Parameters.Add("@id", MySqlDbType.VarChar).Value = idsk.Text;
        command.Parameters.Add("@postav", MySqlDbType.VarChar).Value = post.Text;
        command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
        command.Parameters.Add("@nomerpost", MySqlDbType.VarChar).Value =
nomer.Text;

        MySqlDataReader MyReader2;
        DB.openConnection();
        MyReader2 = command.ExecuteReader();
        MessageBox.Show("Save Data");
        while (MyReader2.Read())
        {
        }
        DB.closeConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    // Добавление данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("Select * From otchety ",
DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}

private void button1_Click(object sender, EventArgs e)
{
    db DB = new db();
    try
    {
        // Формирование запроса
        MySqlCommand command = new MySqlCommand("update otchety set id_sklada = @id,
postavshik = @post, data_postavki = @data where nomer_postavki = @nomer",
DB.GetConnection());
        command.Parameters.Add("@id", MySqlDbType.VarChar).Value = idsk.Text;

```

```

        command.Parameters.Add("@post", MySqlDbType.VarChar).Value = post.Text;
        command.Parameters.Add("@data", MySqlDbType.VarChar).Value = data.Text;
        command.Parameters.Add("@nomer", MySqlDbType.VarChar).Value = nomer.Text;
        MySqlDataReader MyReader2;
        DB.openConnection();
        MyReader2 = command.ExecuteReader();
        MessageBox.Show("Data Updated");
        while (MyReader2.Read())
        {
        }
        DB.closeConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    } // Редактирование данных
    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();
    // Формирование запроса
    MySqlCommand command2 = new MySqlCommand("Select * From otchety ",
    DB.GetConnection());

    adapter.SelectCommand = command2;
    adapter.Fill(table);
    otobraz.DataSource = table;
}
}
}

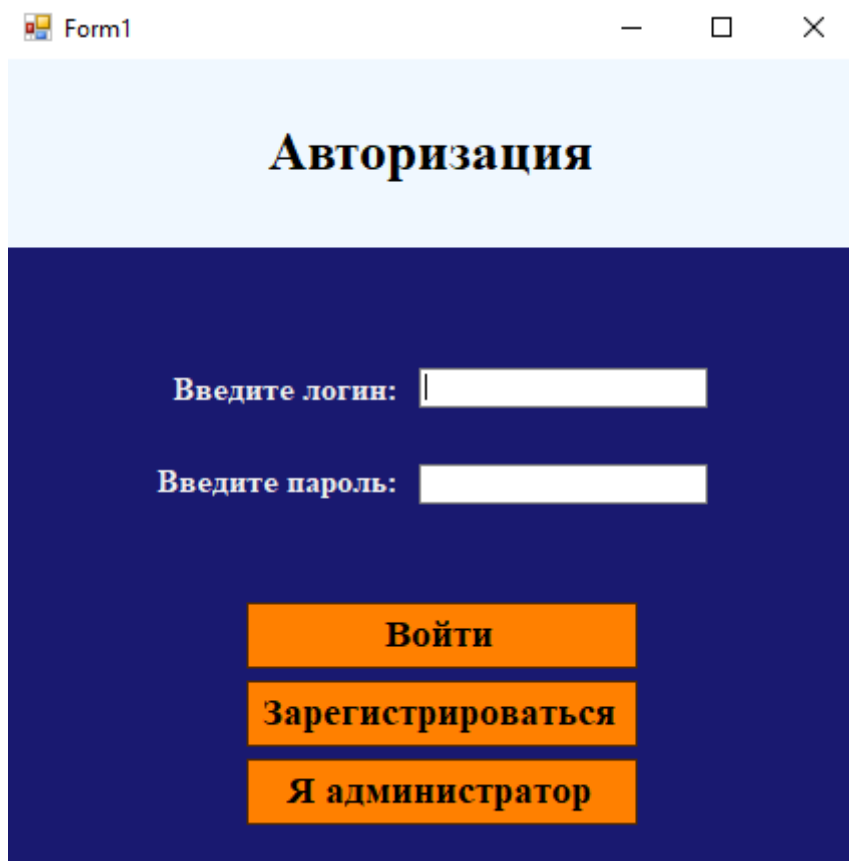
```

ПРИЛОЖЕНИЕ Б

(обязательное)

Результаты выполнения программы

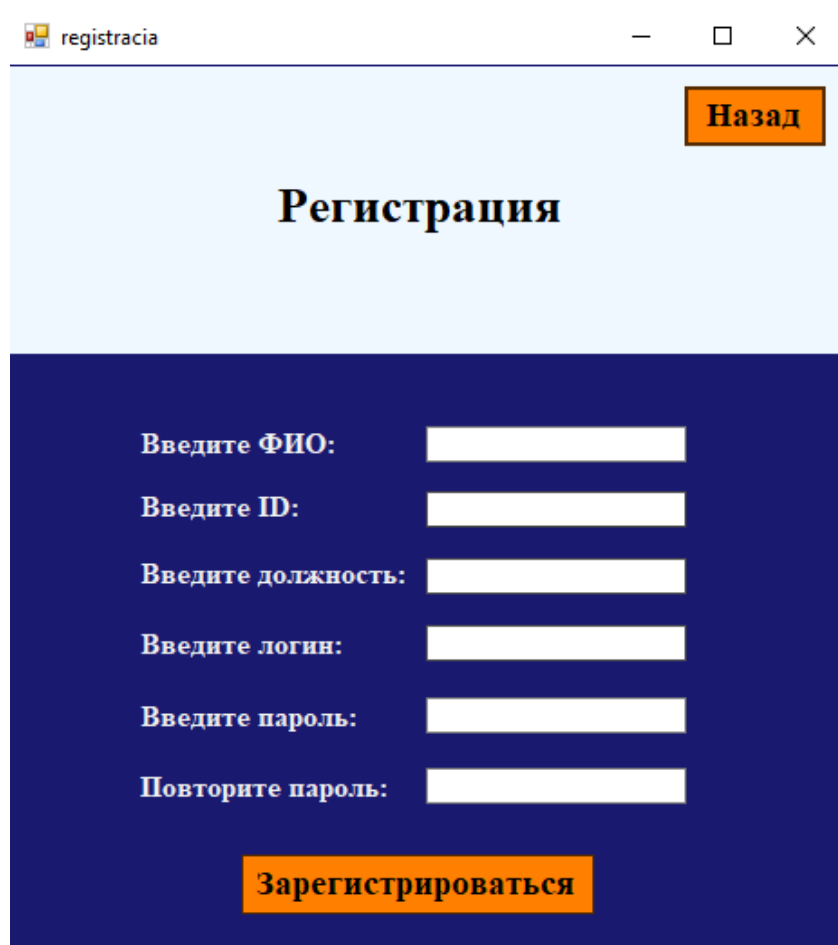
На рисунке Б.1 представлена форма авторизации пользователя.



The image shows a screenshot of a Windows application window titled 'Form1'. The window contains a login form with a light blue header bar with the title 'Авторизация' in bold black text. The main area has a dark blue background. It features two white input fields: the first is preceded by the text 'Введите логин:' and the second by 'Введите пароль:'. Below these fields are three orange buttons with black text, stacked vertically: 'Войти', 'Зарегистрироваться', and 'Я администратор'.

Рисунок Б.1 – Форма авторизации

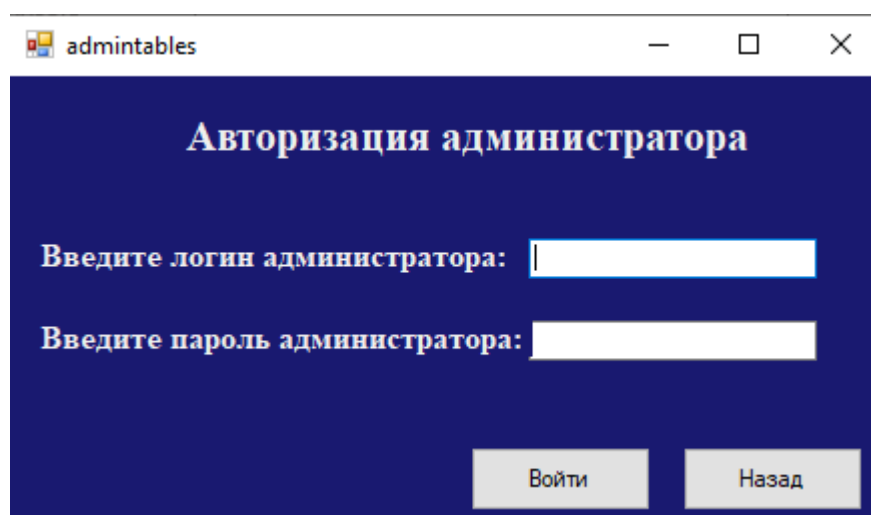
На рисунке Б.2 представлена форма регистрации нового пользователя.



The screenshot shows a window titled 'registracia' with standard Windows window controls. The form has a light blue header with the title 'Регистрация' and an orange 'Назад' button. The main area has a dark blue background with white text and input fields. The fields are: 'Введите ФИО:', 'Введите ID:', 'Введите должность:', 'Введите логин:', 'Введите пароль:', and 'Повторите пароль:'. An orange 'Зарегистрироваться' button is at the bottom.

Рисунок Б.2 – Форма регистрации

На рисунке Б.3 представлена форма авторизации для администратора.



The screenshot shows a window titled 'admintables' with standard Windows window controls. The form has a dark blue background with white text. The title 'Авторизация администратора' is at the top. Below it are two input fields: 'Введите логин администратора:' and 'Введите пароль администратора:'. At the bottom are two buttons: 'Войти' and 'Назад'.

Рисунок Б.3 – Форма авторизации администратора

На рисунке Б.4 представлена форма главного меню пользователя.

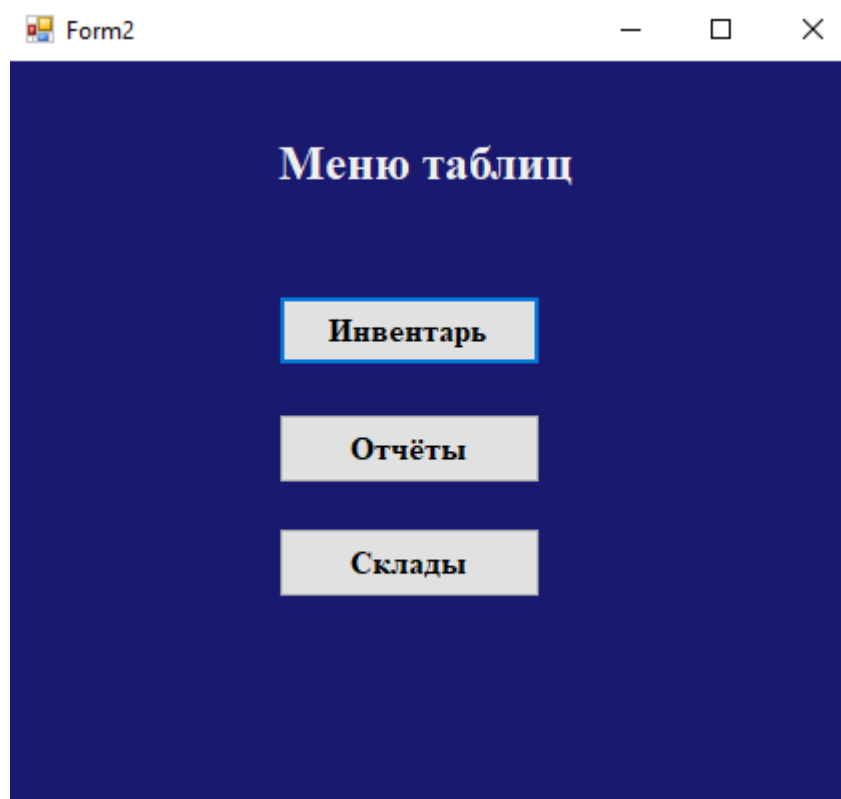


Рисунок Б.4 – Форма главного меню

На рисунке Б.5 представлена форма таблицы «инвентарь».

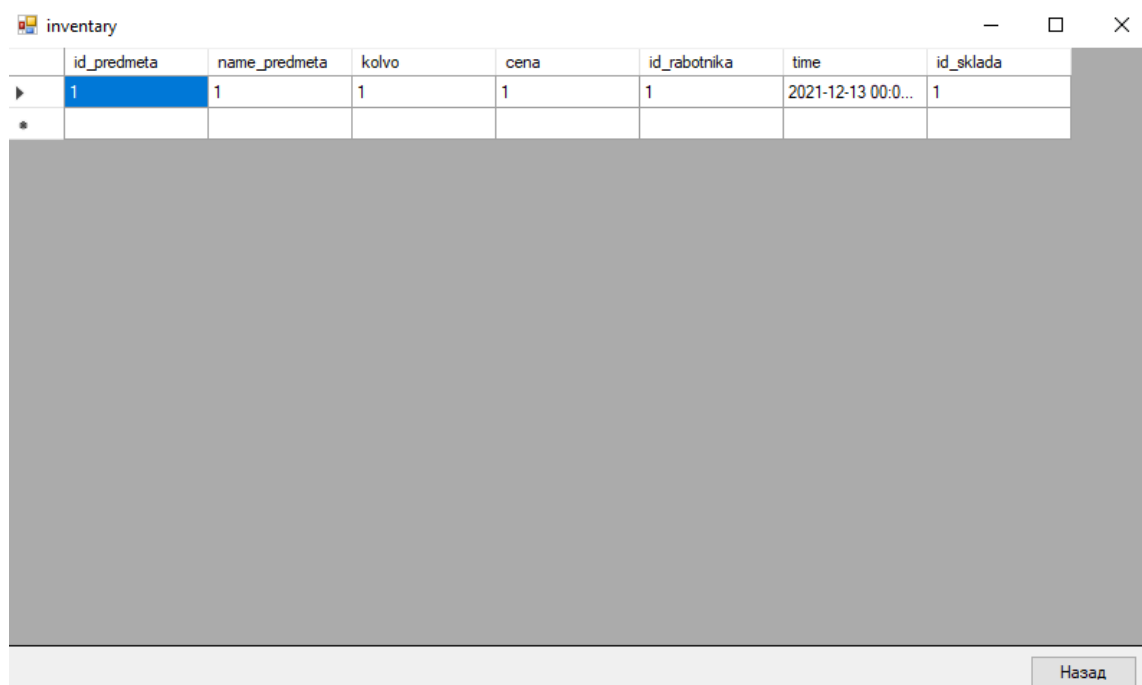


Рисунок Б.5 – Форма таблицы «инвентарь»

На рисунке Б.6 представлена форма таблицы «отчёты».

	id_sklada	postavshik	data_postavki	nomer_postavki
▶	1	1	14 декабря 202...	1
	7	7	13 декабря 202...	4
*				

Назад

Рисунок Б.6 – Форма таблицы «отчёты»

На рисунке Б.7 представлена форма таблицы «склады».

	id_sklada	raspoloz	vmestimost	nomer_postavki
▶	1	1	1	1
	3	4	4	4
*				

Назад

Рисунок Б.7 – Форма таблицы «склады»

На рисунке Б.8 представлена форма главного администратора.

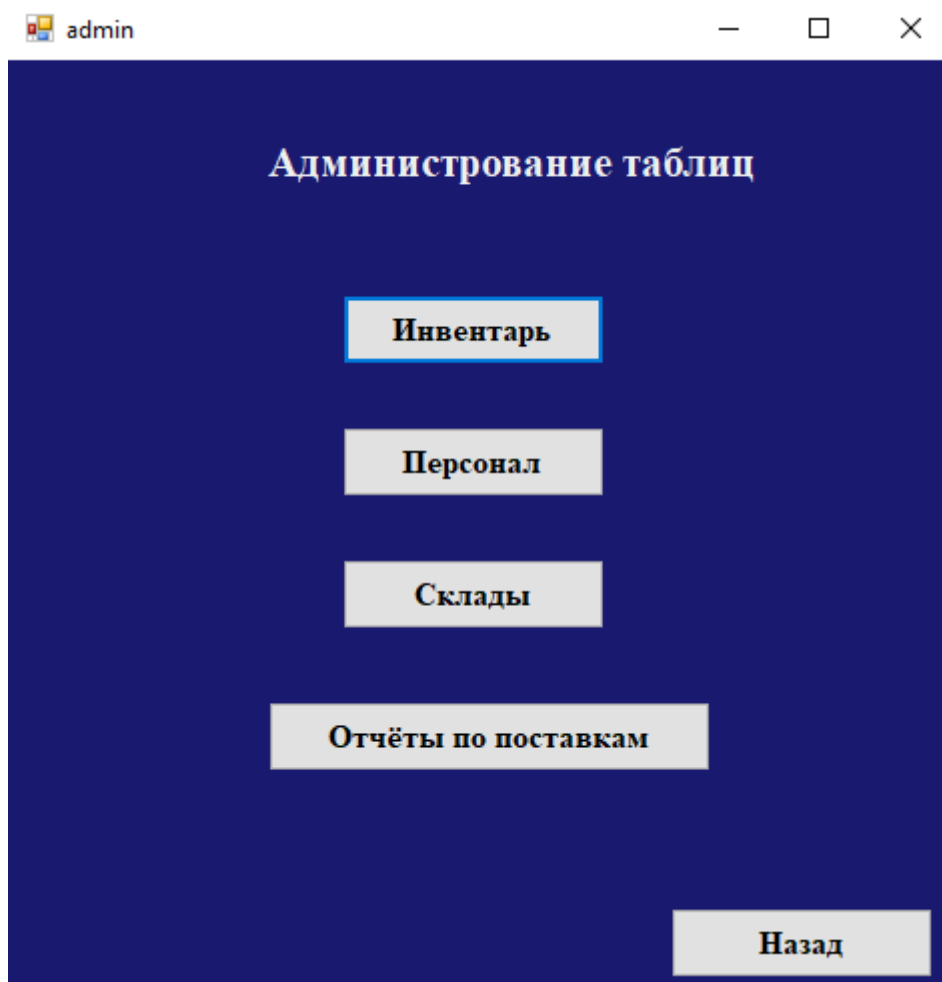


Рисунок Б.8 – Форма главного администратора

На рисунке Б.9 представлена форма таблицы «инвентарь».

The screenshot shows a web application window titled 'inventoryadmin'. On the left, there is a table with the following columns: id_predmeta, name_predmeta, kolvo, cena, id_robтника, time, and id_sklada. The first row contains the values 1, 1, 1, 1, 1, 2021-12-13 00:0..., and 1. Below the table is a large grey rectangular area. On the right side of the window, there is a panel titled 'Добавление и редактирование' (Add and edit). This panel contains several input fields for: Id предмета, Название предмета, Колличество, Цена, id работника, Дата поставки (with a dropdown menu showing '19 декабря 20'), and id склада. Below these fields are two buttons: 'Добавить' (Add) and 'Редактировать' (Edit). Further down, under the heading 'Удаление и поиск' (Delete and search), there are two more input fields and three buttons: 'Поиск' (Search), 'Удаление' (Delete), and 'Назад' (Back).

Рисунок Б.9 – Форма таблицы «инвентарь»

На рисунке Б.10 представлена форма таблицы «персонал».

The screenshot shows a web application window titled 'personaladmin'. On the left, there is a table with the following columns: id_rsbтника, FIO, dolznost, log, and pas. The first three rows contain the values: (1, ivan, sklad, 1, 1), (2, admin, admin, 1, 1), and (312, Мик, mi, Mik, SuckmeDuck1). Below the table is a large grey rectangular area. On the right side of the window, there is a panel titled 'Редактирование' (Edit). This panel contains three input fields for: Id работника, ФИО, and Должность. Below these fields is a button: 'Редактировать' (Edit). Further down, under the heading 'Удаление и поиск' (Delete and search), there are two input fields and three buttons: 'Поиск' (Search), 'Удаление' (Delete), and 'Назад' (Back).

Рисунок Б.10 – Форма таблицы «персонал»

На рисунке Б.11 представлена форма таблицы «склады».

The screenshot shows a window titled 'skladadmin'. It contains a table with the following data:

	id_sklada	raspoloz	vmestimost	nomer_postavki
▶	1	1	1	1
	3	4	4	4
*				

To the right of the table is a form titled 'Добавление и редактирование' (Add and edit). It includes four input fields labeled 'Id склада', 'Поставщик', 'вместимость', and 'Номер поставки'. Below these fields are two buttons: 'Добавить' (Add) and 'Редактировать' (Edit). Further down is a section titled 'Удаление и поиск' (Delete and search), which contains two input fields and three buttons: 'Поиск' (Search), 'Удаление' (Delete), and 'Назад' (Back).

Рисунок Б.11 – Форма таблицы «склады»

На рисунке Б.12 представлена форма таблицы «отчёты по поставкам».

The screenshot shows a window titled 'adminotch'. It contains a table with the following data:

	id_sklada	postavshik	data_postavki	nomer_postavki
▶	1	1	14 декабря 202...	1
	7	7	13 декабря 202...	4
*				

To the right of the table is a form titled 'Добавление и редактирование' (Add and edit). It includes four input fields labeled 'Id склада', 'Поставщик', 'Дата поставки' (with a dropdown arrow), and 'Номер поставки'. Below these fields are two buttons: 'Добавить' (Add) and 'Редактировать' (Edit). Further down is a section titled 'Удаление и поиск' (Delete and search), which contains two input fields and three buttons: 'Поиск' (Search), 'Удаление' (Delete), and 'Назад' (Back).

Рисунок Б.12 – Форма таблицы «отчёты по поставкам»