**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ**

**ТВЕРСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение**

**«Вышневолоцкий колледж»**

**ОТЧЕТ**

На тему: **«Учебная практика по ПМ.02»**

По специальности: 09.02.07 «Информационные системы и программирование»

По ПМ.02 «Осуществление интеграции программных модулей»

Выполнил студент группы П-46-18

Павлов Евгений Геннадьевич

Руководитель проекта:

Яковлева Жанна Сергеевна

г. Вышний Волочек

2022 г.

**Оглавление**

[Глава 1. Работа в MS Excel 4](#_Toc96181097)

[1.1. Разработка простейшей базы данных в программе MS Excel 4](#_Toc96181098)

[1.2. Функции ВПР и ГПР, операции с диапазонами 6](#_Toc96181099)

[Глава 2. Руководство пользователя по работе с GitHub 9](#_Toc96181100)

[2.1. Область применения 9](#_Toc96181101)

[2.2. Краткое описание возможностей 9](#_Toc96181102)

[2.3. Уровень подготовки пользователя 9](#_Toc96181103)

[2.4. Перечень эксплуатационной документации 9](#_Toc96181104)

[2.5. Список обозначений и сокращений 9](#_Toc96181105)

[2.6. Назначение и условия применения 9](#_Toc96181106)

[2.7. Git+GitHub — назначение и возможности 9](#_Toc96181107)

[2.8. Условия пользования 10](#_Toc96181108)

[2.9. Подготовка к работе 10](#_Toc96181109)

[2.9.1. Установка необходимого набора инструментов на Windows 10](#_Toc96181110)

[2.9.2. Порядок проверки работоспособности 11](#_Toc96181111)

[2.10. Описание операций 12](#_Toc96181112)

[2.10.1. Регистрация на GitHub и онлайн репозиторий 12](#_Toc96181113)

[2.10.2. Использование команд 15](#_Toc96181114)

[2.10.3. Добавление своих команд (алиасы) 19](#_Toc96181115)

[2.10.4. Работа с ветками 19](#_Toc96181116)

[2.10.5. Сохранение фрагментов кода и создание своих Gist 20](#_Toc96181117)

[2.10.6. Работа с панелью «Система управления версиями» 22](#_Toc96181118)

[2.10.7. Хостинг 22](#_Toc96181119)

[2.10.8. Удаление репозитория 24](#_Toc96181120)

[2.10.9. Создание файла .gitignore 24](#_Toc96181121)

[Глава 3. Разработка базы данных 26](#_Toc96181122)

[3.1. Моделирование данных. ERD 26](#_Toc96181123)

[3.2. Создание базы данных посредством запроса 27](#_Toc96181124)

[3.3. Создание аналитических запросов 28](#_Toc96181125)

[Глава 4. Разработка приложения для работы в день С-1 (руководство пользователя) 29](#_Toc96181126)

[4.1. База данных 29](#_Toc96181127)

[4.2. Разработка форм авторизации и поиска 29](#_Toc96181128)

[4.3. Разработка форм фильтрации и сортировки 30](#_Toc96181129)

[Глава 5. Разработка desktop-приложений 31](#_Toc96181130)

[5.2. Разработка ERD диаграмм 31](#_Toc96181131)

[5.2.1. Работа в MS Visio 31](#_Toc96181132)

[5.2.2. Подготовка данных для импорта 32](#_Toc96181133)

[5.2.3. Разработка базы данных «Мебельная фабрика» 33](#_Toc96181134)

[5.3. Создание приложения 35](#_Toc96181135)

[5.4. Разработка формы авторизации 37](#_Toc96181136)

[5.5. Регистрация заказчиков 37](#_Toc96181137)

[5.6. Учет фурнитуры и материалов 38](#_Toc96181138)

[Глава 6. Разработка документации 40](#_Toc96181139)

[6.1. Разработка руководства пользователя 40](#_Toc96181140)

[6.2. Разработка диаграммы 41](#_Toc96181141)

[6.2.1. ERD-диаграмма 41](#_Toc96181142)

[6.2.2. Диаграмма прецедентов 42](#_Toc96181143)

[6.2.3. Диаграмма деятельности 43](#_Toc96181144)

[6.2.4. Диаграмма взаимодействия 44](#_Toc96181145)

[6.2.5. Диаграмма классов 45](#_Toc96181146)

# Глава 1. Работа в MS Excel

## Разработка простейшей базы данных в программе MS Excel

Была спроектирована и релизована база данных «Страховая компания», состоящая из одного списка.



Рис.1. Таблица «Страховая компания» с двадцатью уникальными записями

Были сформулированы и реализованы пять запросов на поиск и отбор информации из созданного списка.

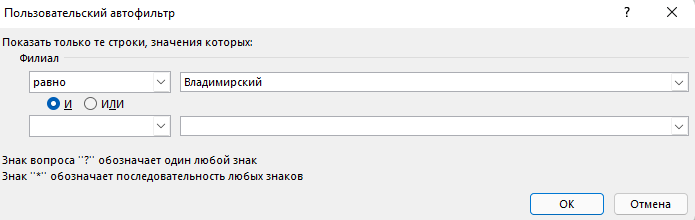


Рис. 2. Критерий для выборки клиентов владимирского филиала

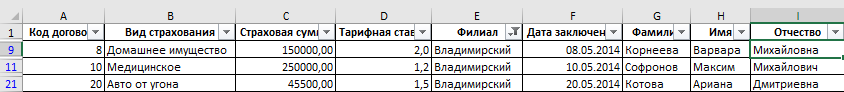


Рис. 3. Выборка клиентов владимирского филиала

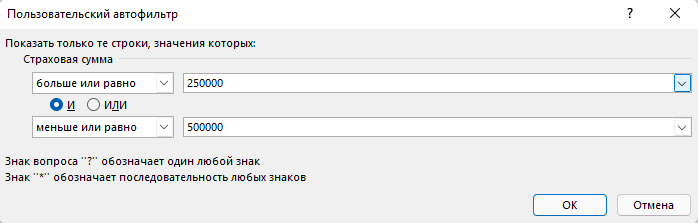


Рис. 4. Критерий для выборки клиентов, страховая сумма которых находится в диапазоне от 250000 до 500000

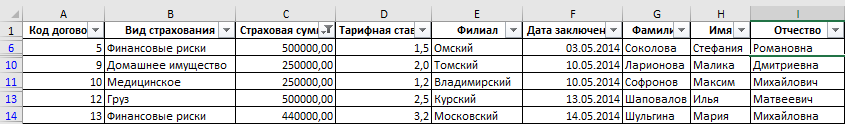


Рис. 5. Выборка клиентов, страховая сумма которых находится в диапазоне от 250000 до 500000

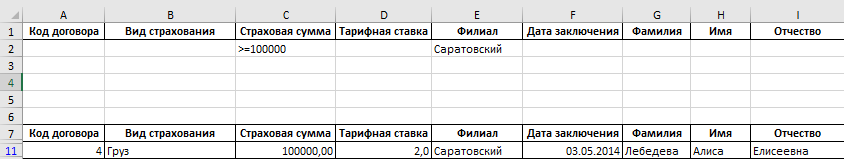


Рис. 6. Выборка клиентов саратовского филиала, страховая сумма которых больше или равна 100000

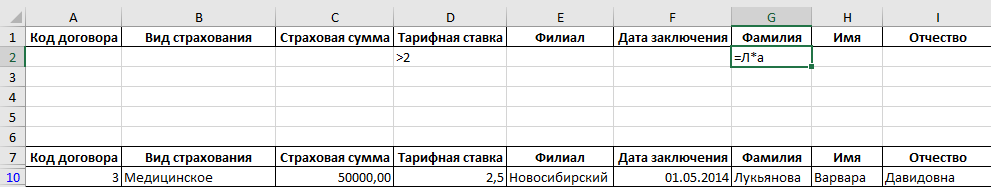


Рис. 7. Выборка клиентов, фамилия которых начинается на «Л» и заканчивается на «а», а тарифная ставка составляет свыше 2

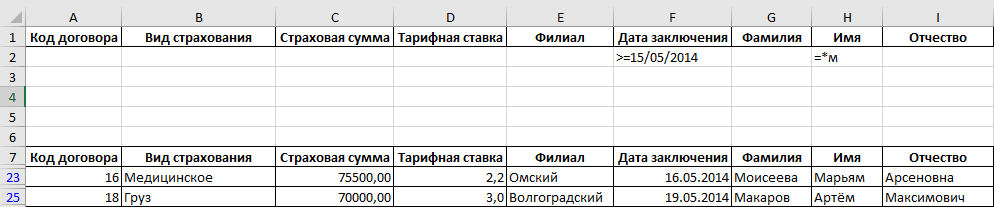


Рис. 8. Выборка клиентов, имя которых заканчивается на «м», а дата заключения договора позже 15 мая 2014

## Функции ВПР и ГПР, операции с диапазонами

Задания были выполнены с использованием всех нужных функций и операций. Результат отображен на скриншотах в таблицах.

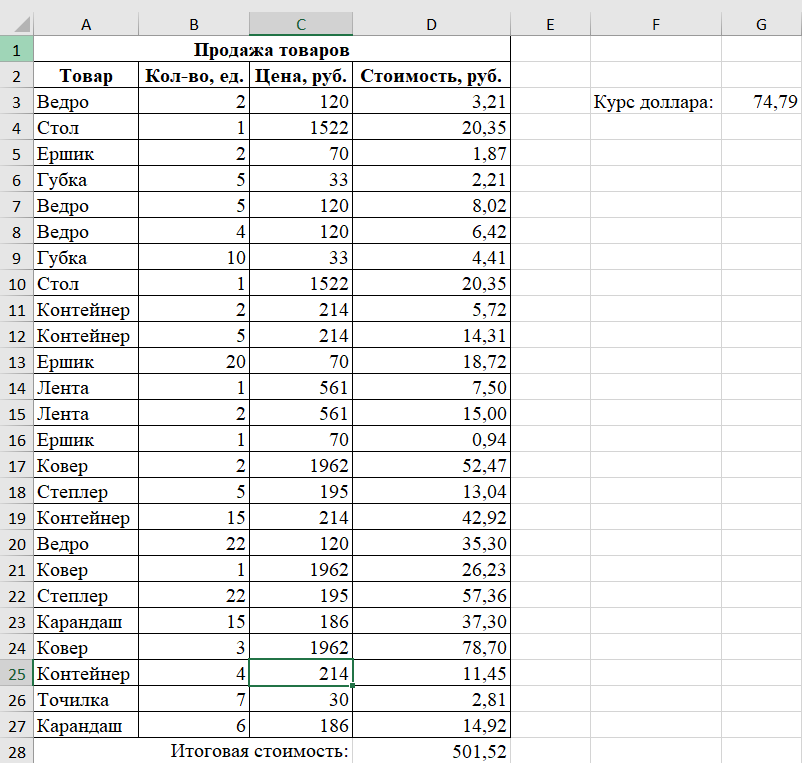


Рис. 9. Продажа

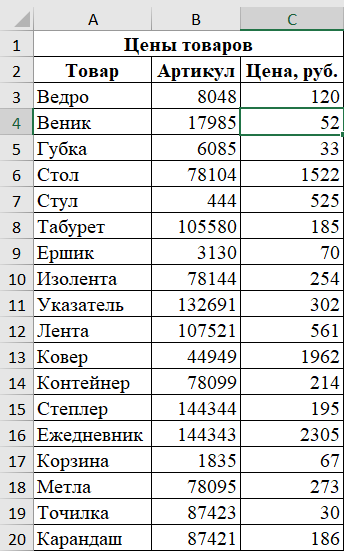


Рис. 10. Цены

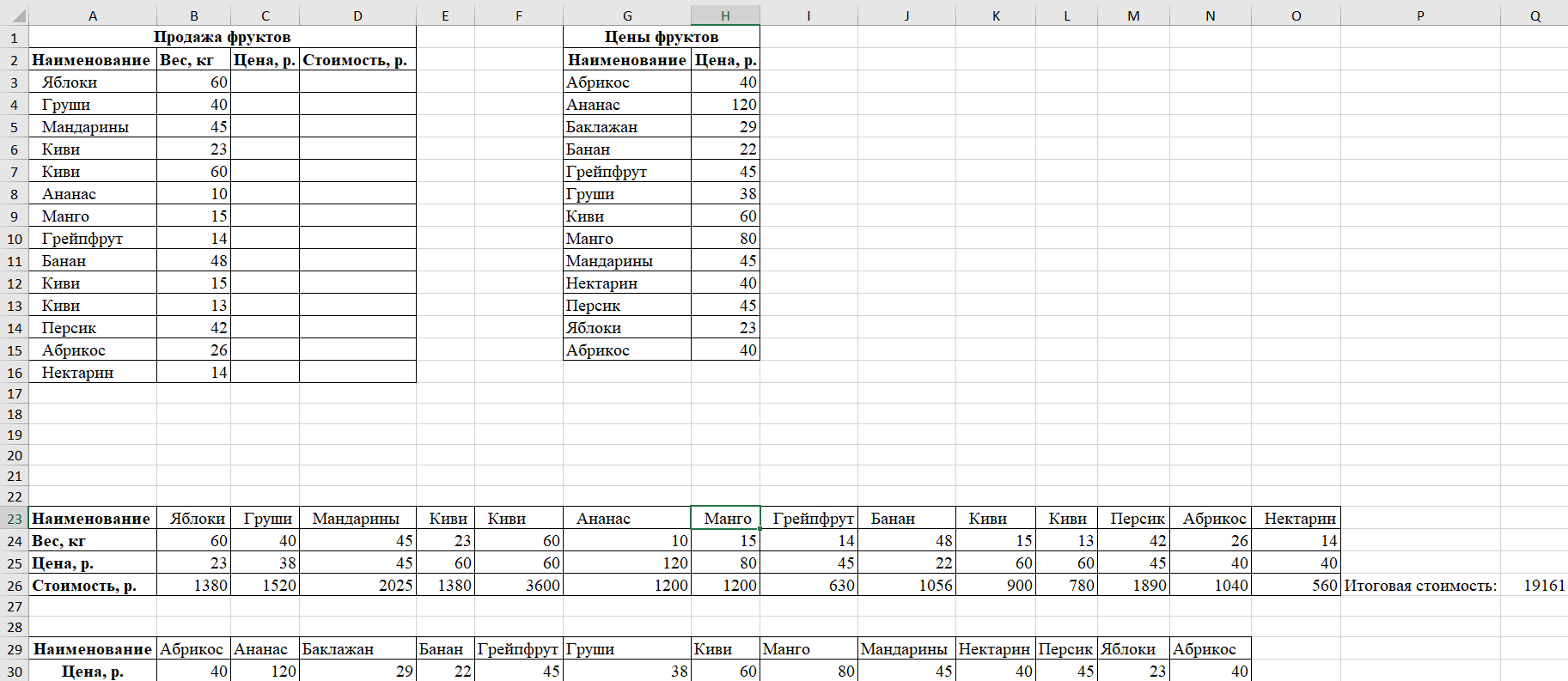


Рис. 11. ГПР

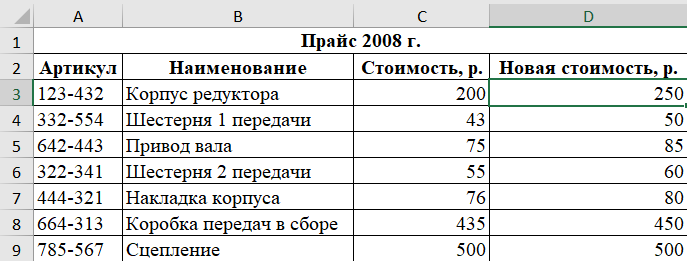


Рис. 12. Прайс 2008г.

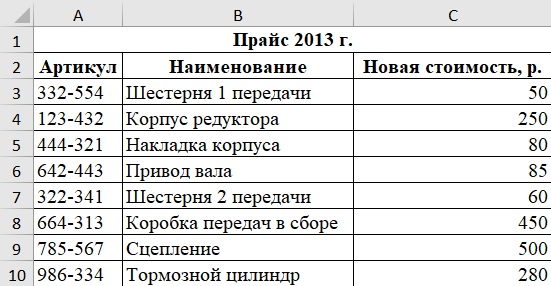


Рис. 13. Прайс 2013г.

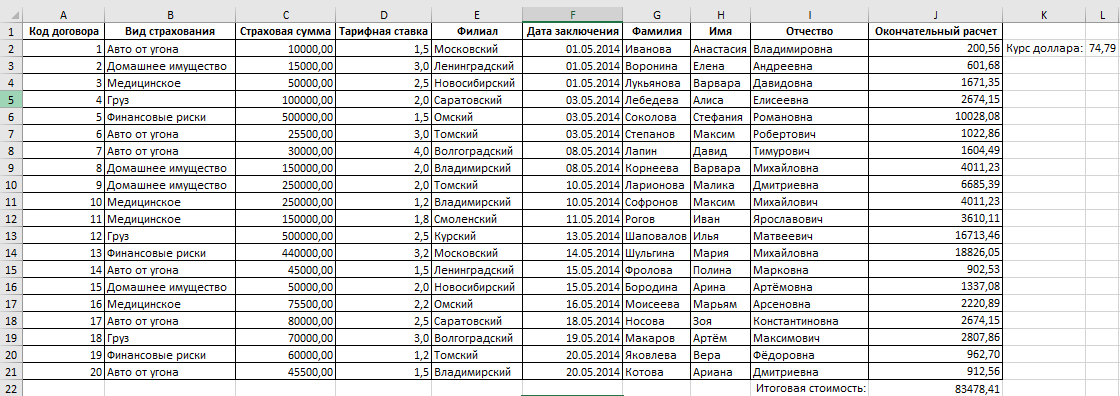


Рис. 14. Страховая компания

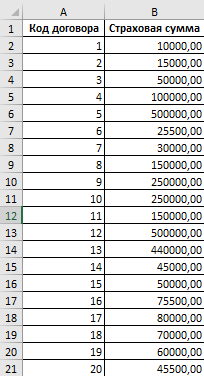


Рис. 15. Страховые суммы

**Глава 2. Руководство пользователя по работе с GitHub**

## Область применения

Система управления версиями — программное обеспечение для облегчения работы с изменяющейся информацией.

## Краткое описание возможностей

Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое. Такие системы наиболее широко используются при разработке программного обеспечения для хранения исходных кодов разрабатываемой программы.

## Уровень подготовки пользователя

Для облегчения работы с системами рекомендуется:

Иметь общие сведения о Visual Studio Code.

## Перечень эксплуатационной документации

1. Инструкция по установке Visual Studio Code;
2. Руководство пользователя (настоящий документ).

## Список обозначений и сокращений

ОС — Операционная система.

## Назначение и условия применения

## Git+GitHub — назначение и возможности

Git **—** распределённая система управления версиями.

Назначение Git+GitHub **—** размещение git*-*репозиториев и совместной разработки проектов.

Возможности Git+GitHub:

* Сохранять все изменения, внесенные в код;
* Даже самое минимальное изменение фиксируется в Git отдельно, а не перезаписывается;
* Можно создавать приватные репозитории, которые будут видны только вам и выбранным вами людям.

## Условия пользования

GitHub не несёт ответственности за то, что вы публикуете.

GitHub не проводит премодерацию контента, однако оставляет за собой право удалять контент, который, по мнению сервиса, нарушает данные правила.

Вы даёте сервису право воспроизводить (reproduce) свой контент (например, в целях бэкапа); отображать (display) его; модифицировать (например, для поисковых нужд); распространять (чтобы у других пользователей GitHub был доступ к вашему контенту) или воспроизводить (perform) — это уже на случай, если вы загружаете музыку или видео. GitHub не имеет права продавать ваше произведение или распространять его за пределами сервиса.

Если вы публикуете контент с пометкой о том, что он распространяется под определённой лицензией, то вы согласны с тем, что он распространяется под этой лицензией. Если вы делаете публикации (contributions) в репозиторий, связанный с какой-либо лицензией, то ваши правки будут связаны с той же лицензией.

Сотрудники GitHub получают доступ к приватным репозиториям только в случае необходимости и с вашего согласия.

## Подготовка к работе

### Установка необходимого набора инструментов на Windows

* Зайдите на сайт git-scm.com;
* Нажмите на раздел Downloads;
* Автоматически начнется загрузка, либо предоставится выбор версии в зависимости от ОС;
* После того, как загрузка завершилась, требуется нажать на этот файл и запустить его. После запуска оставлять все параметры по умолчанию и нажимать Next до начала установки;
* После установки уберите галочку с View Release Notes и нажмите Finish.

### Порядок проверки работоспособности

* Открыть командную строку с помощью Win+R, прописав cmd;
* Ввести «git --version» без кавычек;
* Должна выводиться версия git (например: git version 2.135.1.windows.2). Если ошибка — ничего выводиться не будет.

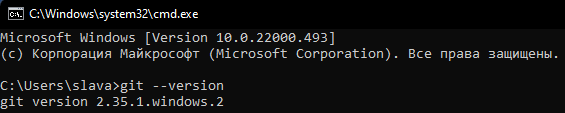


Рис. 16. Вывод версии git

* Открыть Visual Studio Code. (Для проверки работоспособности в самой системе);
* Создать папку;
* В редакторе Visual Studio Code нажать открыть папку в панели быстрого доступа и выбрать ранее созданную папку;
* Нажать на правую кнопку мыши и создать файл (например: index html);
* Сделать стандартную разметку;
* В строке title прописать (например: Работа с GIT);
* В теле кода (body) прописать (например: <p>Git\_GitHub</p>);
* Сохранить

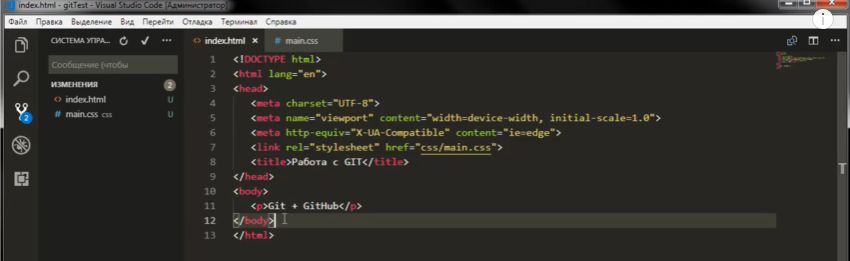


Рис. 17. Создание файлов и папок

## Описание операций

### Регистрация на GitHub и онлайн репозиторий

Для регистрации на GitHub и создание репозитория требуется:

* Зайти на сайт github.com;
* Ввести свои данные для регистрации;

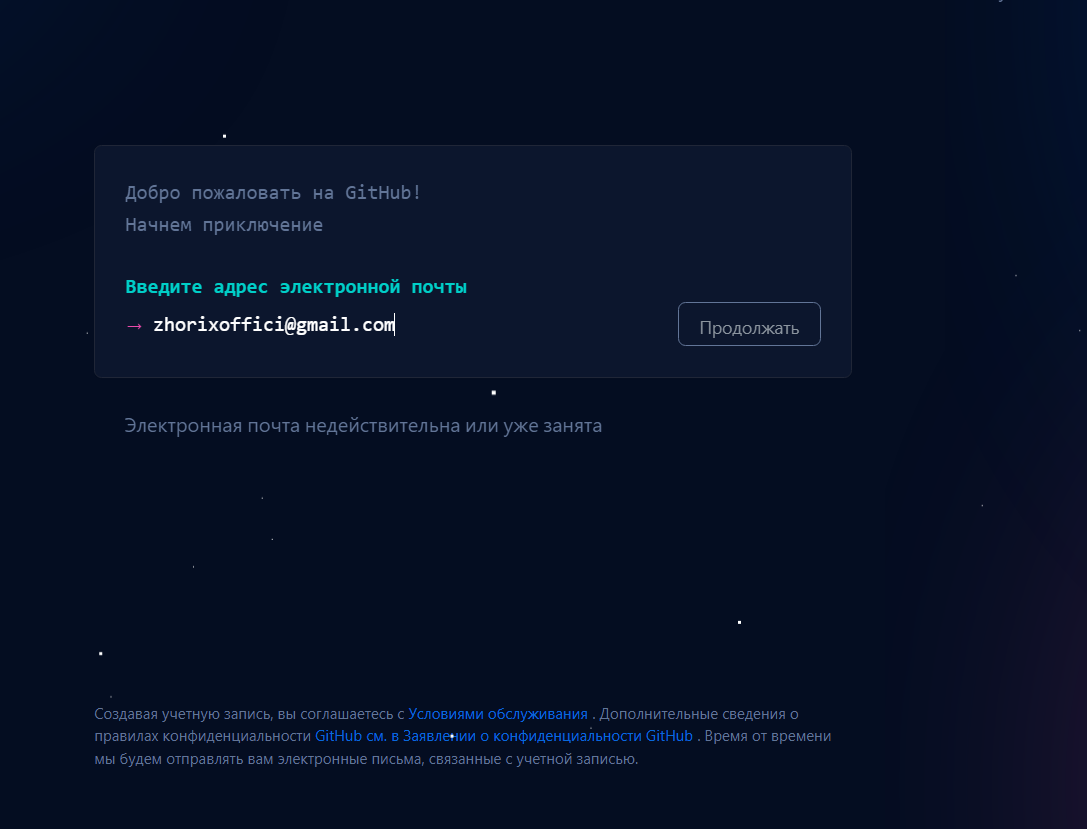


Рис. 18. Регистрация на github.com

* После подтверждения своего аккаунта нажмите на кнопку «Create repository»;

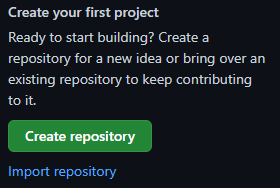


Рис. 19. Создание репозитория

* Создать репозиторий, вписав название и описание, выбрав приватный или общедоступный;

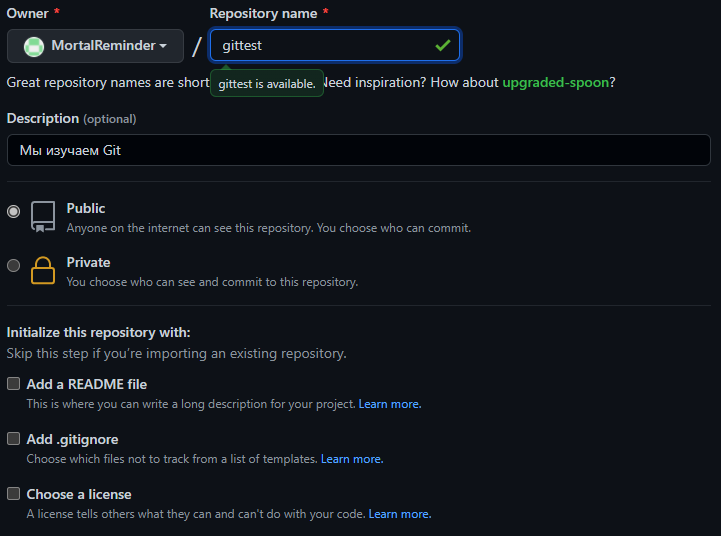


Рис. 20. Выбор параметров репозитория

Для работы с репозиторием требуется:

* Скопировать ссылку;



Рис. 21. Копирование ссылки

* Открыть редактор и прописать в терминале построчно git init, git add, git commit -m “first commit” (ваш коммит);

Коммиты — основные конструктивные элементы временной шкалы проекта Git.

* Далее прописать git remote add origin и вставить ссылку https://github.com/MortalReminder/gittest.git;
* Затем прописать git push -u origin master;

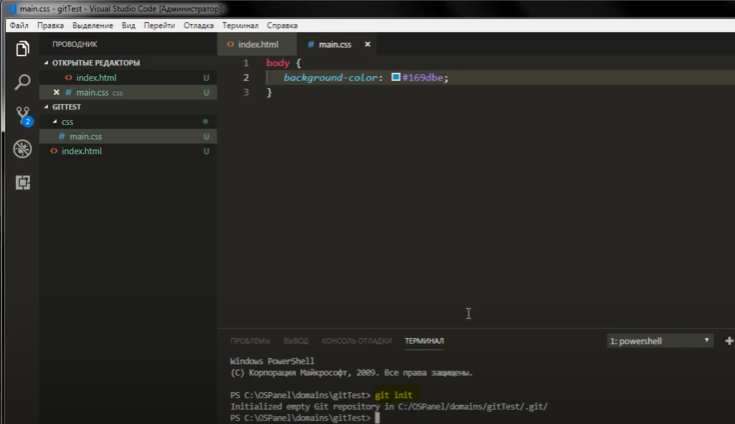


Рис. 22. Прописывание нужных данных в терминале

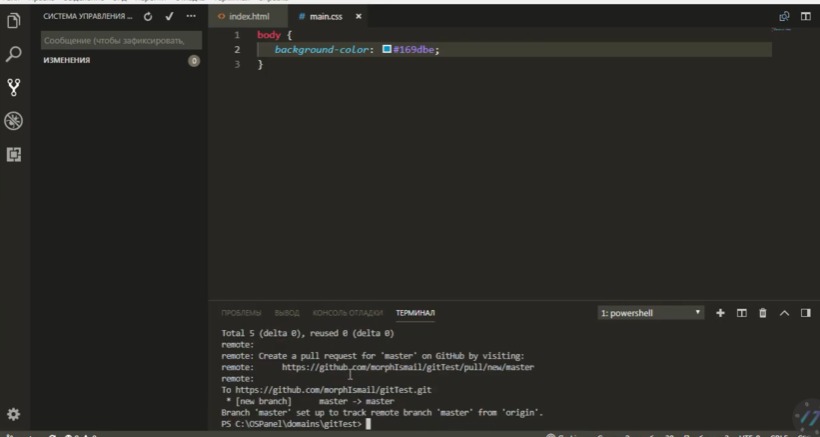


Рис. 23. Прописанные данные

При обновлении на сайте увидим созданные ранее папки и файлы

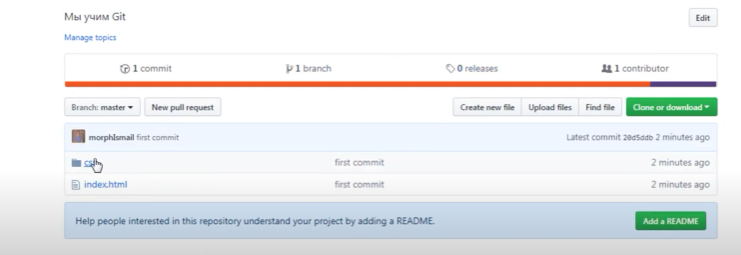


Рис. 24. Папки и файлы на сайте

Можно использовать звездочку в качестве избранного:



Рис. 25. Избранный репозиторий

В настройках в разделе Collaborators можно добавить пользователя:

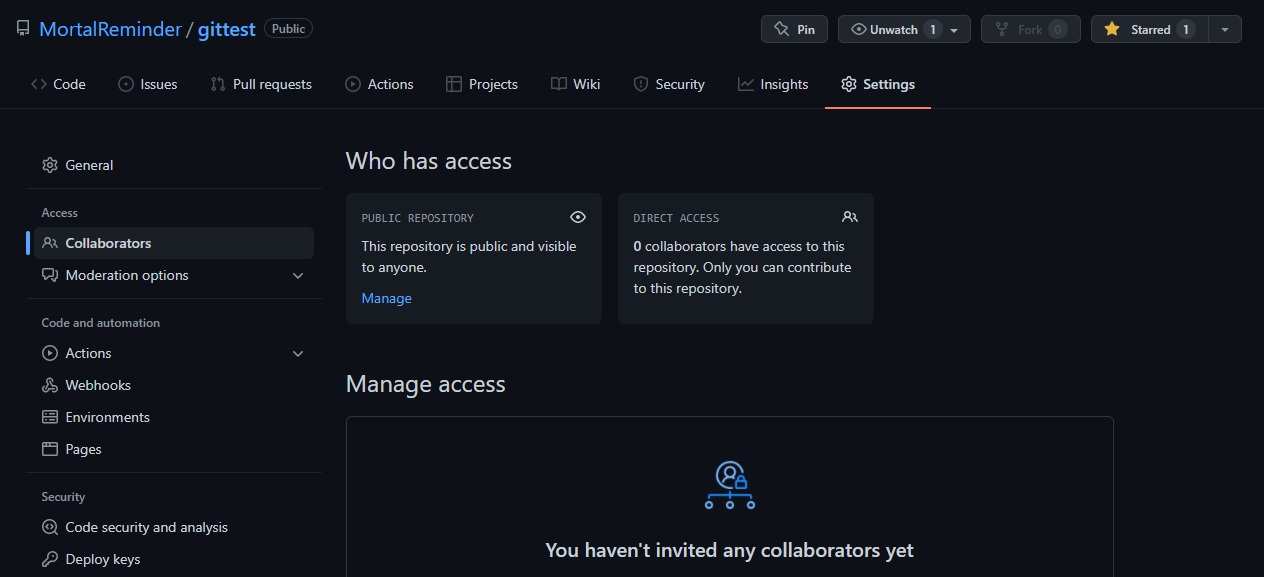


Рис. 26. Добавление нового пользователя репозитория

### Использование команд

git clone — это утилита командной строки Git для выбора существующего репозитория и создания его клона, т. е. копии.

Команда git status отображает состояние рабочего каталога и раздела проиндексированных файлов.

Команда git add добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита.

git commit — это команда для записи индексированных изменений в репозиторий Git.

Команда git push используется для выгрузки содержимого локального репозитория в удаленный репозиторий.

Если вы удалили папку, ее можно скачать с сайта или же скопировать ссылку:

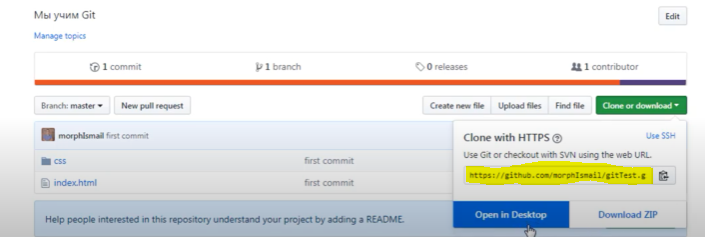


Рис. 27. Копирование ссылки

А потом прописать в редакторе git clone и данную ссылку:

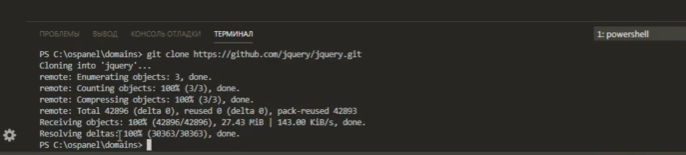


Рис. 28. Команда git clone

Для добавление содержимого рабочего каталога в индекс прописываем в терминале git add и индекс:



Рис. 29. Команда git add.

С помощью gid status можно узнать какие файлы были модифицированы (выделены красным цветом):

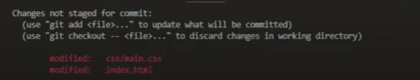


Рис. 30. Команда gid status

Далее, прописав git add. (с точкой!) для добавления файлов, а после снова gid status мы узнаем, что файлы были добавлены и готовы к отправке. (выделены зеленым цветом)

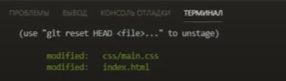


Рис. 31. Команда (git add.)

Далее для записи индексированных изменений в репозиторий Git прописываем git commit -m «Название»:

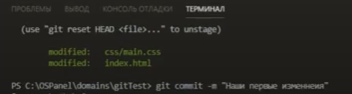


Рис. 32. Команда git commit

Прописав снова git status, заметим, что файлы в терминале отсутствуют:

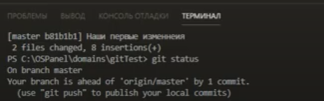


Рис. 33. Команда git status

Можно использовать git push для выгрузки репозитория на сервер:

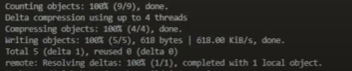


Рис. 34. Команда git push

Команда git pull используется для извлечения и загрузки содержимого из удаленного репозитория и немедленного обновления локального репозитория этим содержимым.

При вводе команды git pull содержимое выгружается на сайт и показываются изменения:

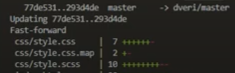


Рис. 35. Команда git pull

Для просмотра изменений можно использовать нижнюю кнопку в панели слева:

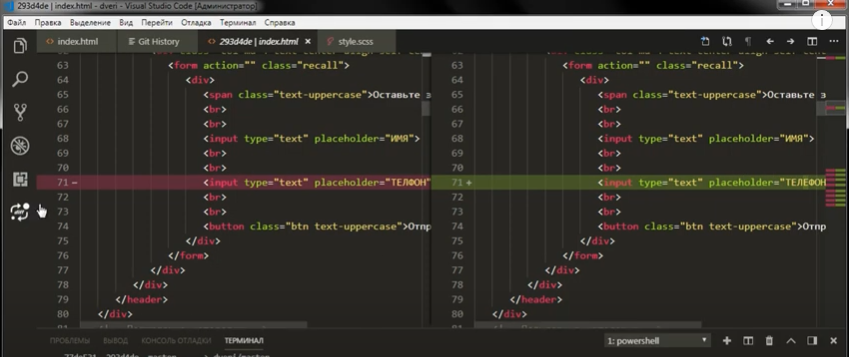


Рис. 36. Просмотр изменений

Так же можно использовать команды с помощью системы управления

Команда get commit – галочка



Рис. 37. Система управления

Команду git push можно осуществить нажатием на троеточие, а далее на «отправить».

Отменить все изменения можно нажатием стрелочки в системе управления:

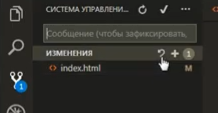


Рис. 38. Отмена

Отменить изменения можно и при помощи терминала прописав команду git checkout – индекс:

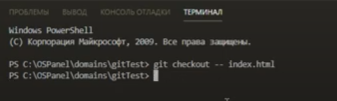


Рис. 39. Команда git checkout

### Добавление своих команд (алиасы)

Для просмотра и добавления команд следует:

* Открыть документ .gitconfig и в разделе alias прописать желаемые сокращения для команд.

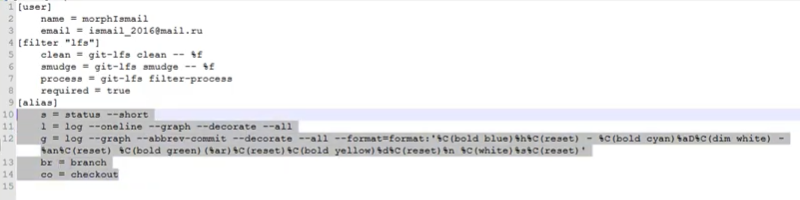


Рис. 40. Алиасы

Прописав в терминале алиасы, они будут работать так же, как и обычные команды.



Рис. 41. Команда git status (алиас)

### Работа с ветками

Команда git branch показывает все ветки. Звезда около ветки означает, в какой ветке мы сейчас находимся:

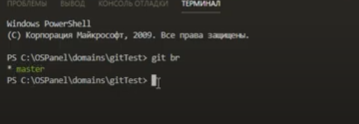


Рис. 42. Команда git branch (алиас)

Если к команде добавить -v, мы увидим какой последний коммит был в ней сделан:

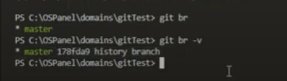


Рис. 43. Команда git branch -v

Если добавить l - увидим подробности о ветке:

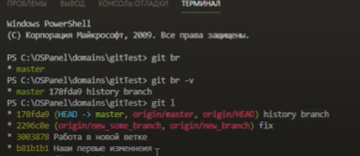


Рис. 44. Команда git branch l

Для создания новой ветки достаточно прописать команду git branch и ввести ее название.

Чтобы в нее перейти нужно прописать git checkout и ее название.

Для быстрого создания и перехода нужно прописать git checkout -b и название новой ветки.

Для того, чтобы переименовать ветку - прописать git branch -m и новое название.

Для добавления ветки на сайт нужно прописать git push –set-upstream origin и название ветки.

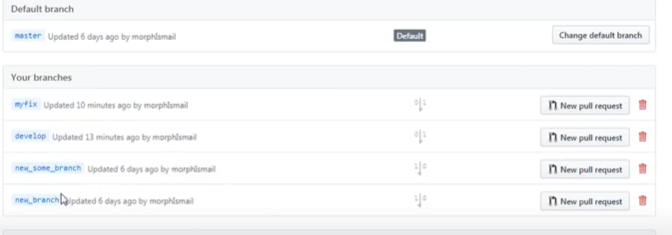


Рис. 45. Ветки, выгруженные на сайт

### Сохранение фрагментов кода и создание своих Gist

Для создания своего Gist потребуется:

* На сайте найти в верхнем правом кнопку New gist



Рис. 46. Кнопка New gist

* Далее вводим название, описание и сам код в соответствующие поля и выбираем сохранить как скрытый gist или публичный.

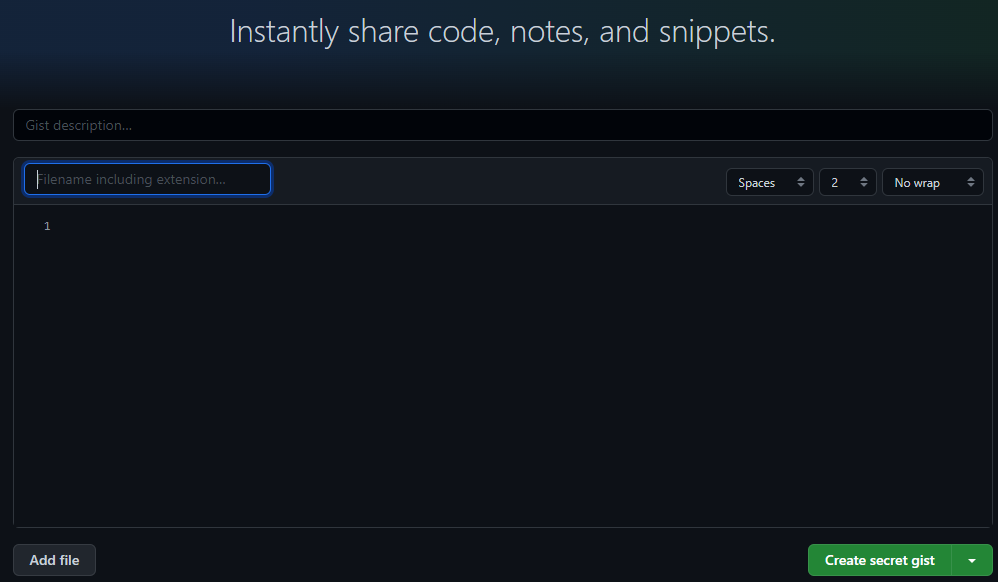


Рис. 47. Создание gist

Так же можно создать gist, установив расширение, которое можно найти в панели слева в самом редакторе:

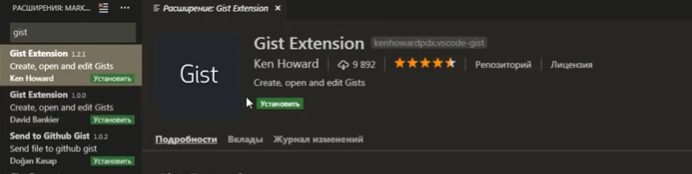


Рис. 48. Установка расширения

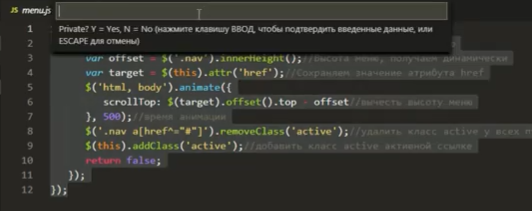


Рис. 49. Создание gist

### Работа с панелью «Система управления версиями»

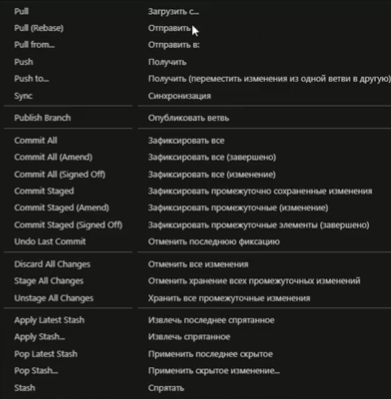


Рис. 50. Список команд

### Хостинг

Хостинг — это размещение интернет-проектов на физических и виртуальных серверах. Для этого требуется:

* В настройках на сайте github.com найти пункт GitHub Pages

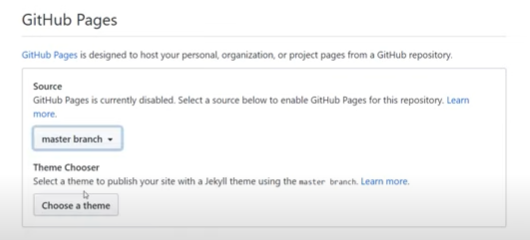


Рис. 51. GitHub Pages

* В этом пункте найти подпункт Source, и выбрать master branch

После этого наша страница доступна в виде сайта.

Перейдя в раздел environment, можно с помощью View deployment посмотреть нашу страницу:

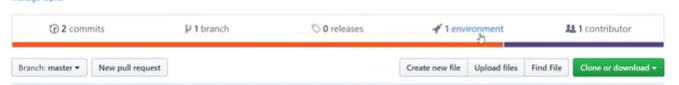


Рис. 52. Раздел environment

### Удаление репозитория

Чтобы удалить репозиторий, требуется:

* В настройках на сайте github.com найти пункт Danger Zone.

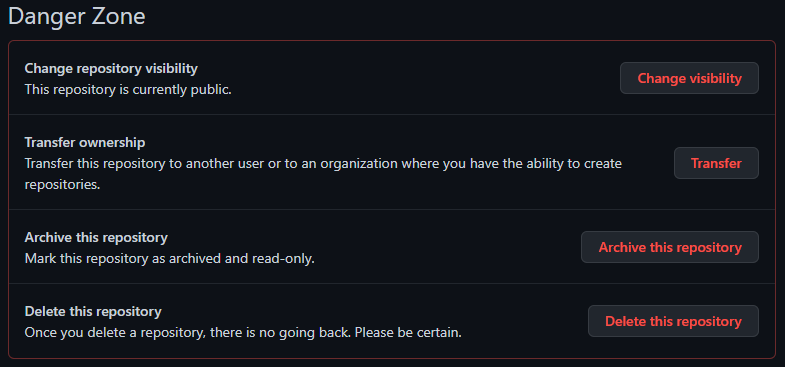


Рис. 53. Danger Zone

* Затем нажать на последний пункт и появляется окно с просьбой повторить написанный текст, после чего репозиторий успешно будет удален.

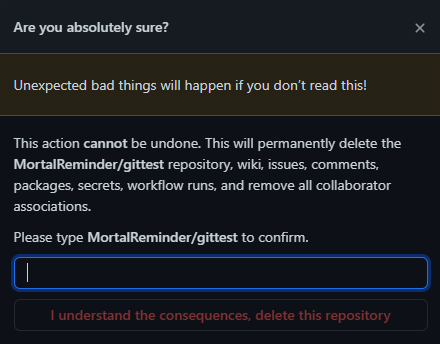


Рис. 54. Удаление репозитория

### Создание файла .gitignore

С помощью файла .gitignore можно указать какие каталоги нужно исключить из репозитория, т. е. они не будут передаваться на сервер.

Для его создания нужно прописать следующий код:

> .gitignore

git add .gitignore

git commit -m "message" .gitignore

А далее вписывать игнорируемые каталоги:

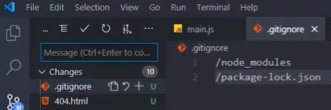


Рис. 55. Gitignore

# Глава 3. Разработка базы данных

## Моделирование данных. ERD

Была разработана база данных под названием «Компьютерные курсы».

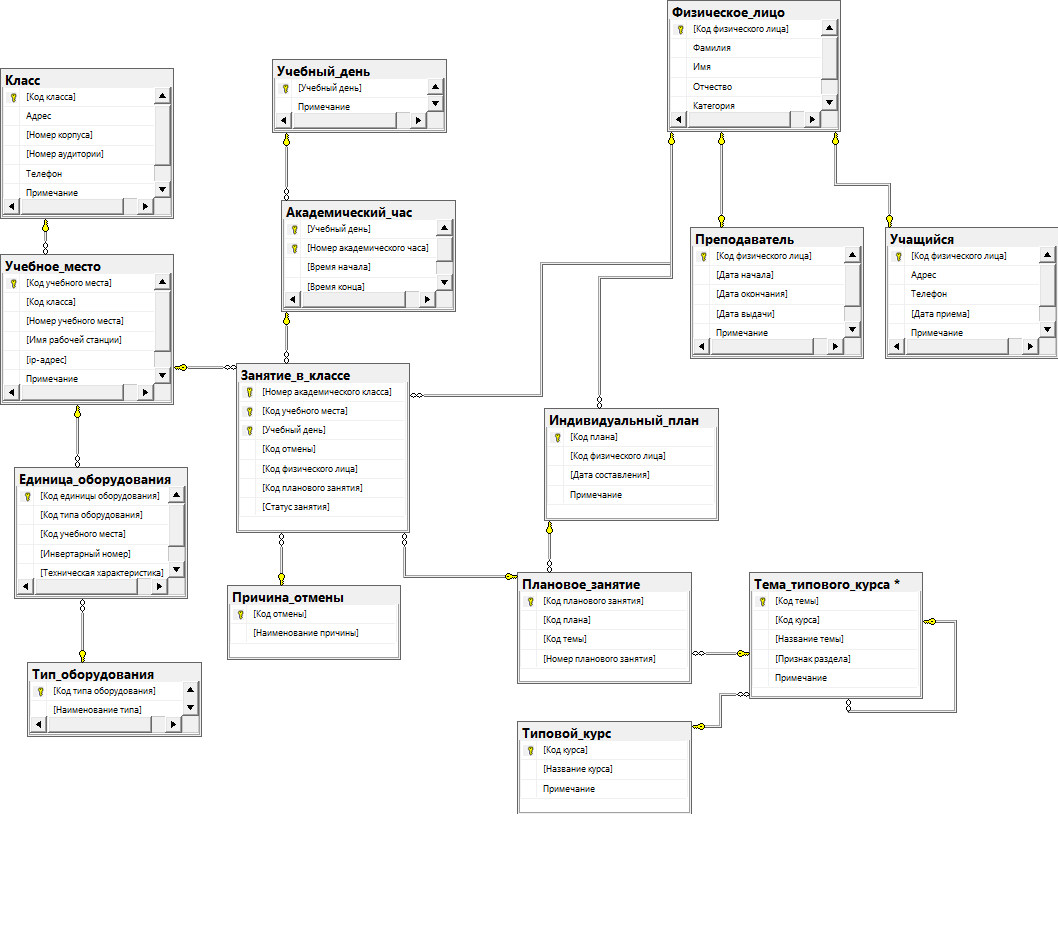


Рис. 56. Схема данных в SQL



Рис. 57. ERD-диаграмма в Visio

## Создание базы данных посредством запроса

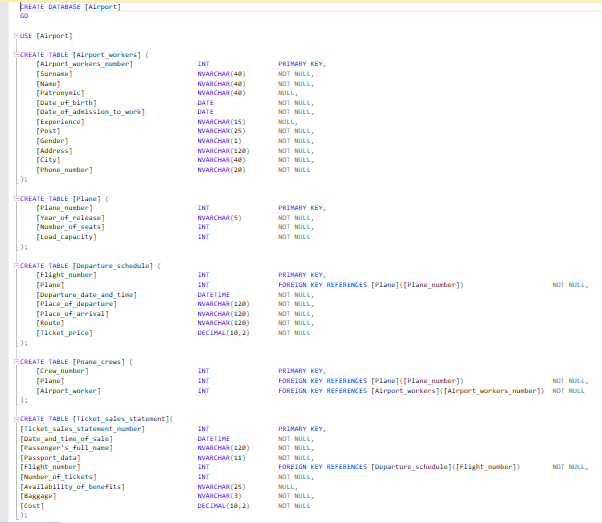


Рис. 58. Запрос на создание базы данных «Аэропорт»

## Создание аналитических запросов

Создан аналитический запрос, который отвечает на вопрос: сколько минимально курсов надо удалить, чтобы объём очистки был не менее 5 Гб.

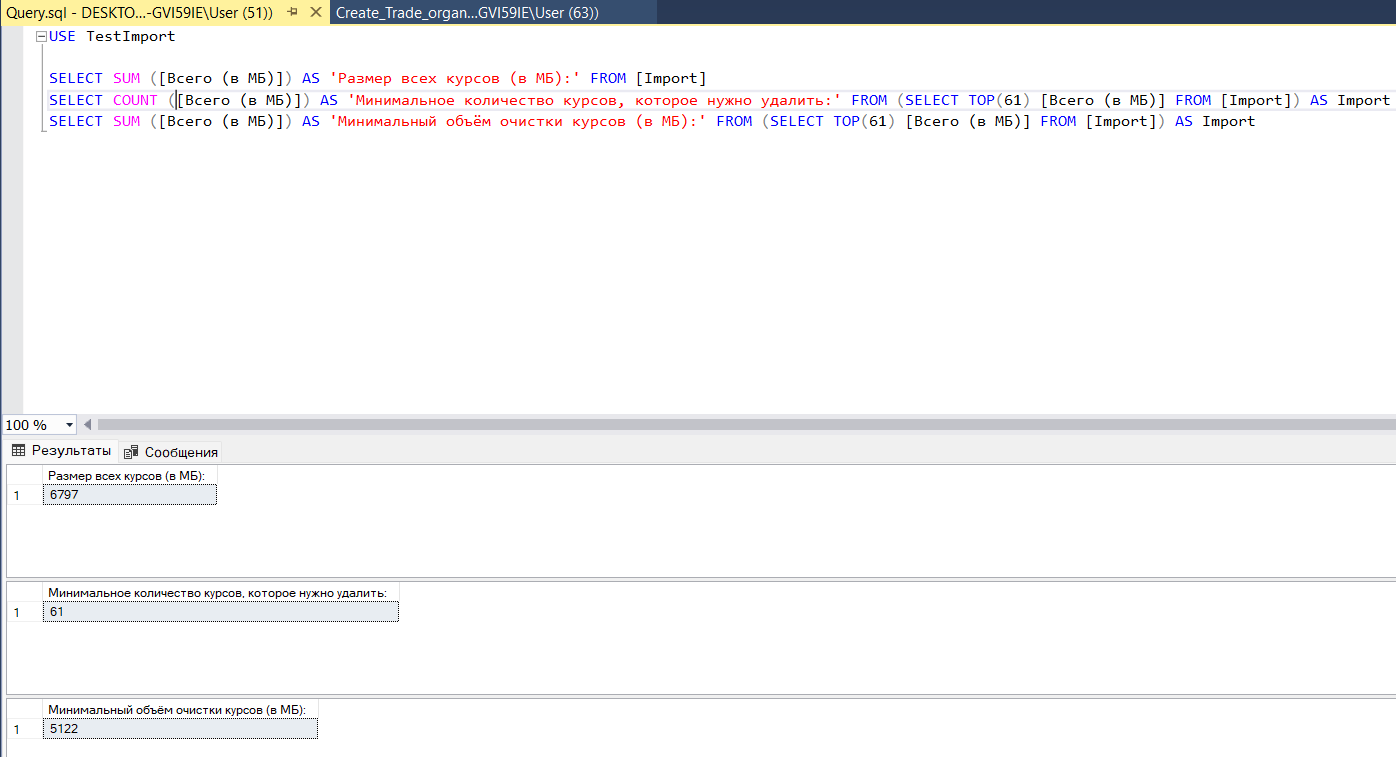


Рис. 59. Аналитический запрос

# Глава 4. Разработка приложения для работы в день С-1 (руководство пользователя)

## База данных

Была разработана база данных с тремя таблицами и связями.



Рис. 60. Меню

## Разработка форм авторизации и поиска

Авторизация разделена на обычного пользователя и администратора. В зависимости от введенных данных программа будет направлять либо в меню администратора, либо в меню пользователя.

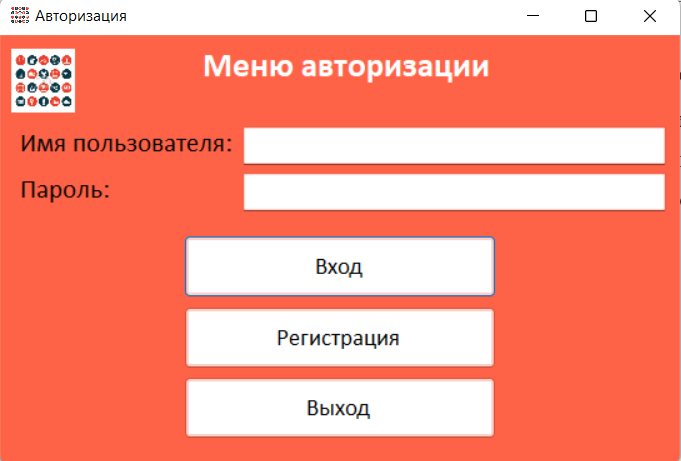


Рис. 61. Авторизация

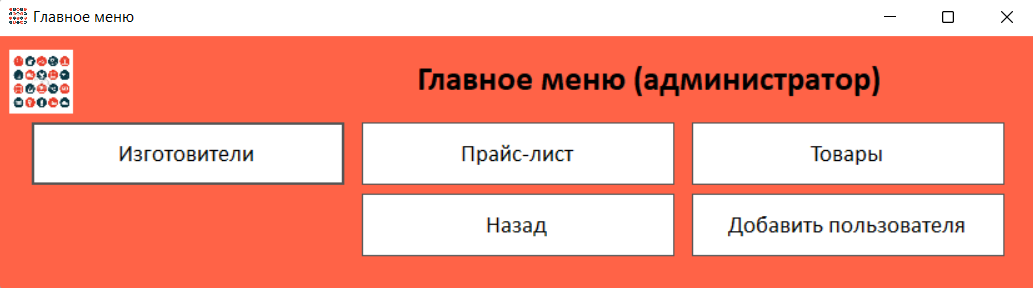


Рис. 62. Меню администратора

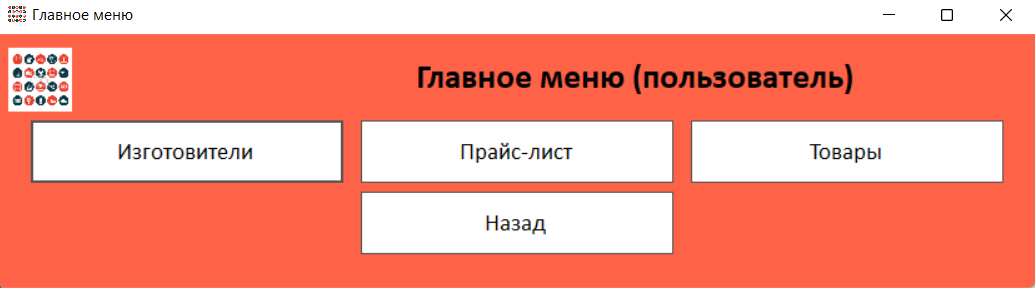


Рис. 63. Меню пользователя

## Разработка форм фильтрации и сортировки

В моём проекте Visual Studio были разработаны формы фильтрации и сортировки, как для администратора, так и для пользователя.

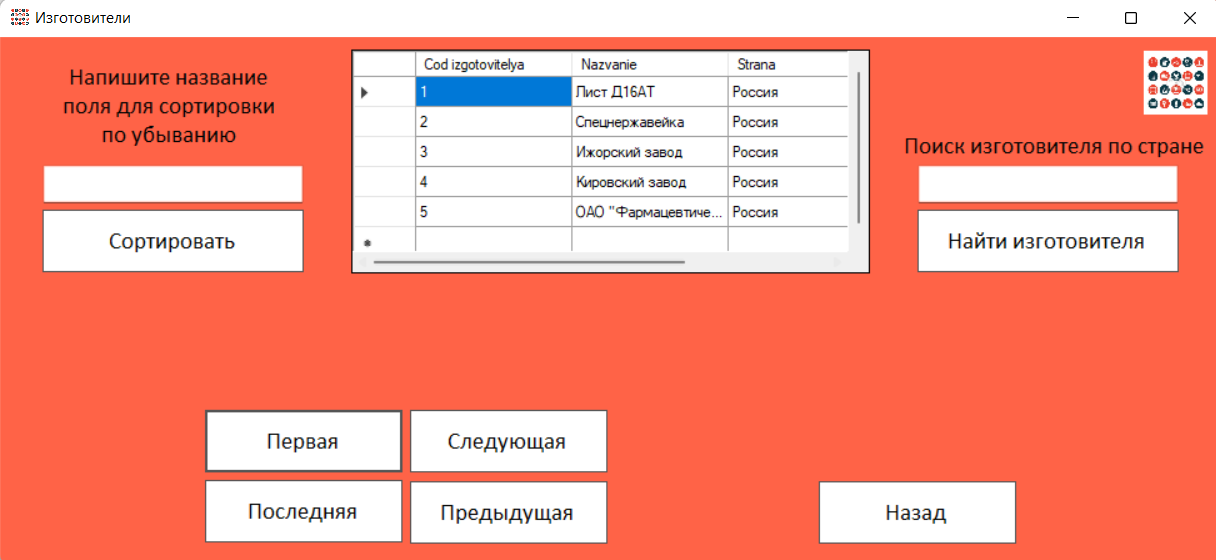


Рис. 64. Сортировка и фильтрация

# Глава 5. Разработка desktop-приложений

## Разработка ERD диаграмм

ERD-диаграмма отображает отношения набора сущностей, хранящиеся в базе данных. Другими словами, мы можем сказать, что ERD-диаграммы помогут вам объяснить логическую структуру баз данных. На первый взгляд диаграмма ER выглядит очень похоже на блок-схему. Однако ERD-диаграмма включает в себя множество специализированных символов, и ее значения делают эту модель уникальной.

* + 1. Работа в MS Visio

Для разработки ERD диаграммы была использована программа Visio. Были определены внешние и первичные ключи, все таблицы логически связаны между собой.



Рис. 65. ERD-диаграмма

### Подготовка данных для импорта

Таблицы выданы в программе Excel. Все таблицы были успешно подготовлены к импорту в программу SQL.

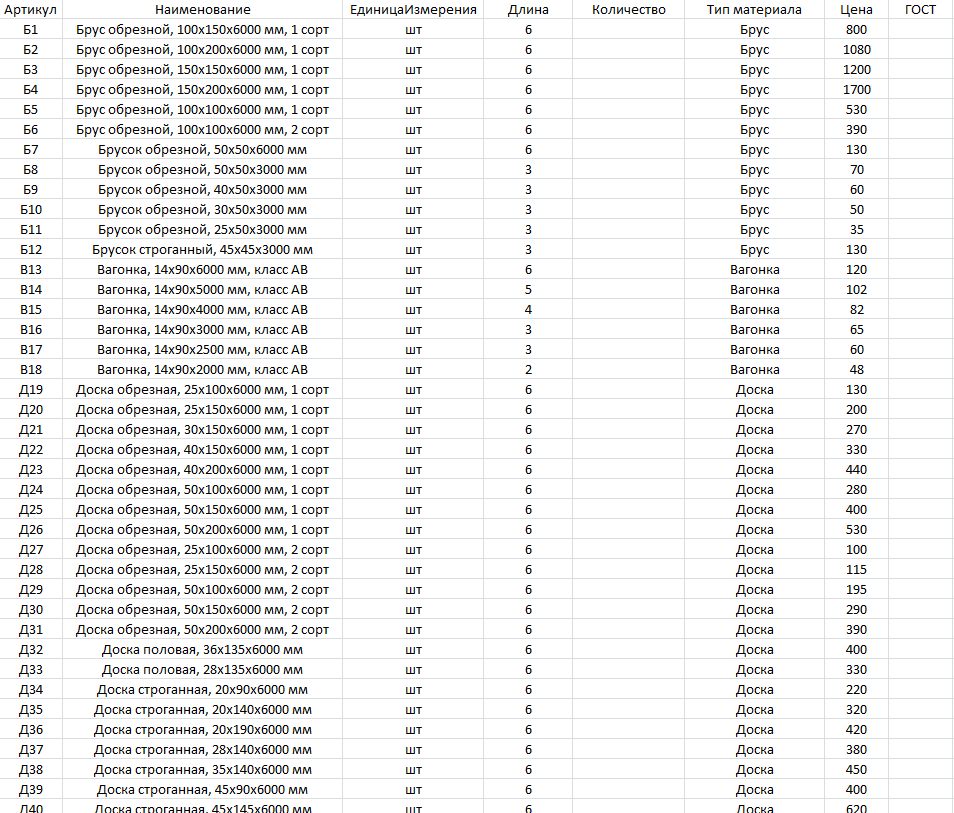


Рис. 66. Таблица «Материал»



Рис. 67. Таблица «Фильтрация»

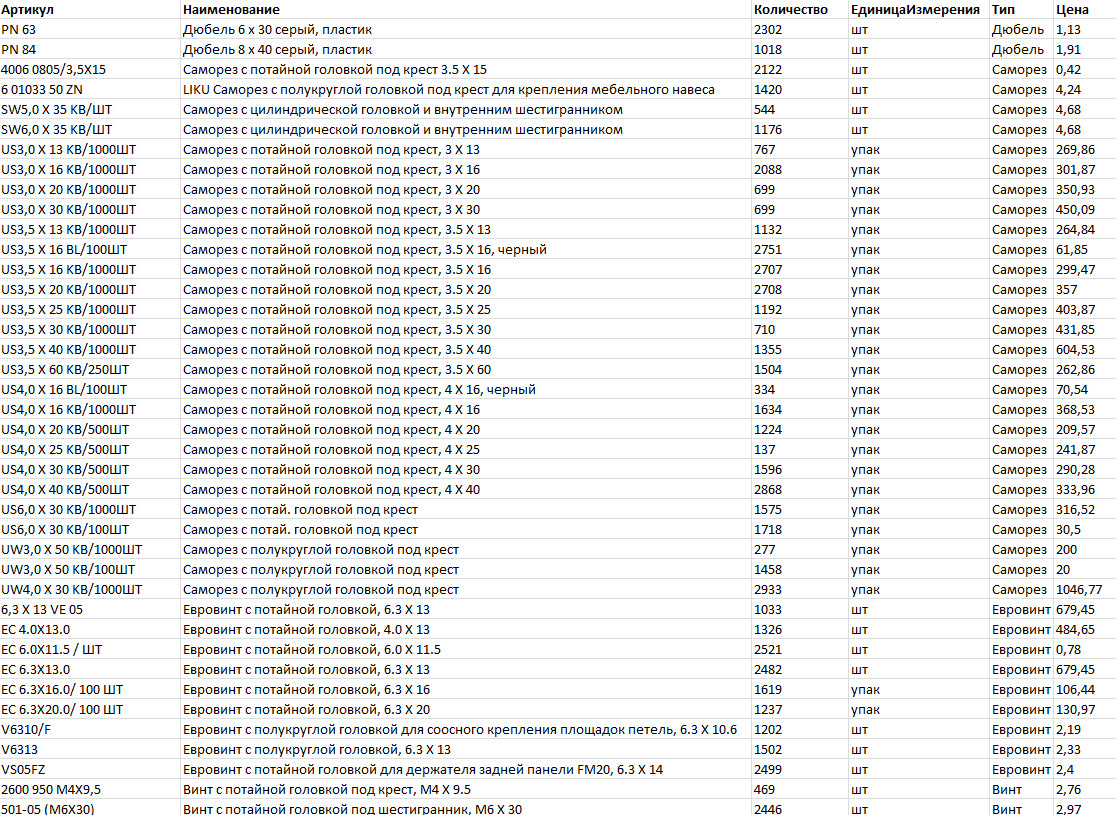


Рис. 68. Таблица «Фурнитура»

### **Разработка базы данных «Мебельная фабрика»**

Были созданы все таблицы БД «Мебельная фабрика». В таблицы успешно осуществлён импорт из Excel.

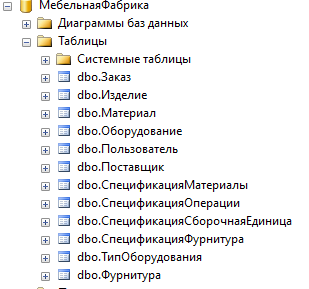
****

Рис. 69. База данных

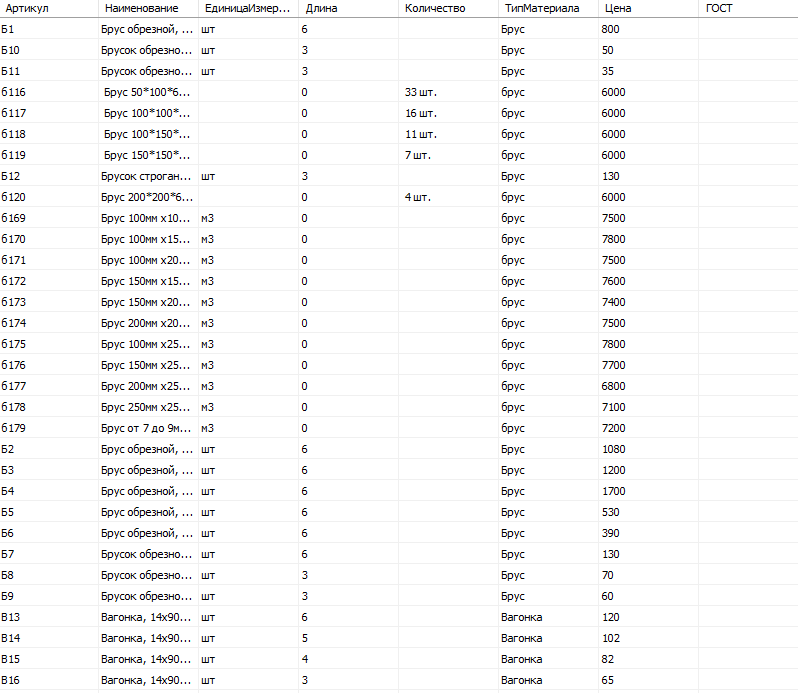
****

Рис. 70. Таблица «Материал» в SQL

****

Рис. 71. Таблица «Фурнитура» в SQL

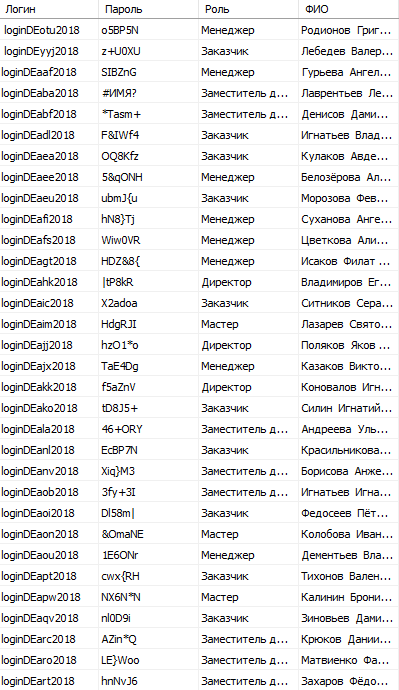
****

Рис. 72. Таблица «Пользователь» в SQL

Созданная схема базы данных в SQL

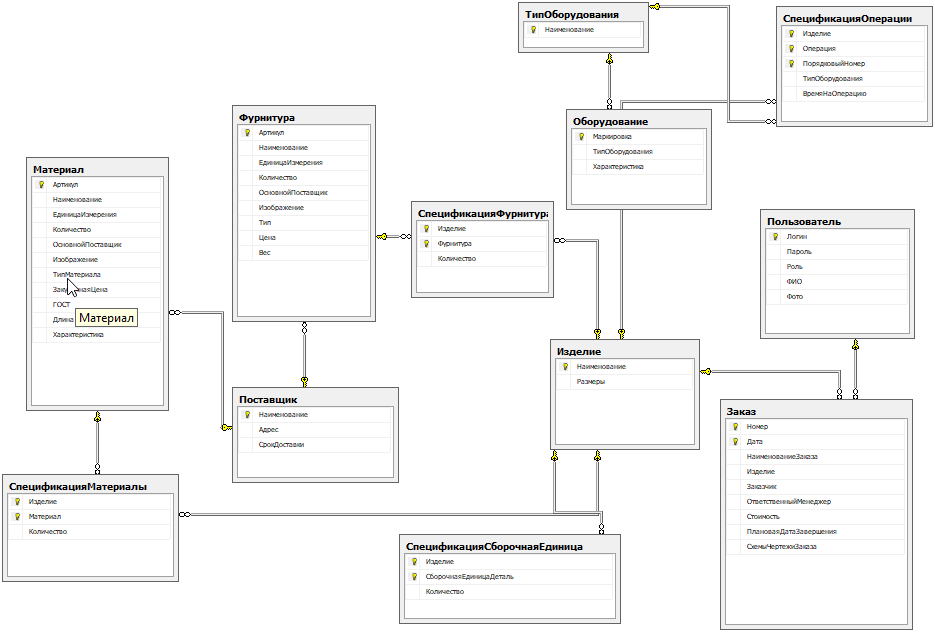


Рис. 73. Схема базы данных в SQL

## **Создание приложения**

Для создания проекта нужно:

1. Запустить Visual studio;
2. Далее выбрать «Приложение Windows Forms»;

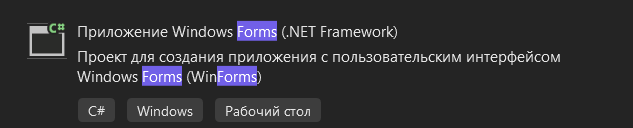


Рис. 74. Выбор типа проекта

1. Задать название проекта;

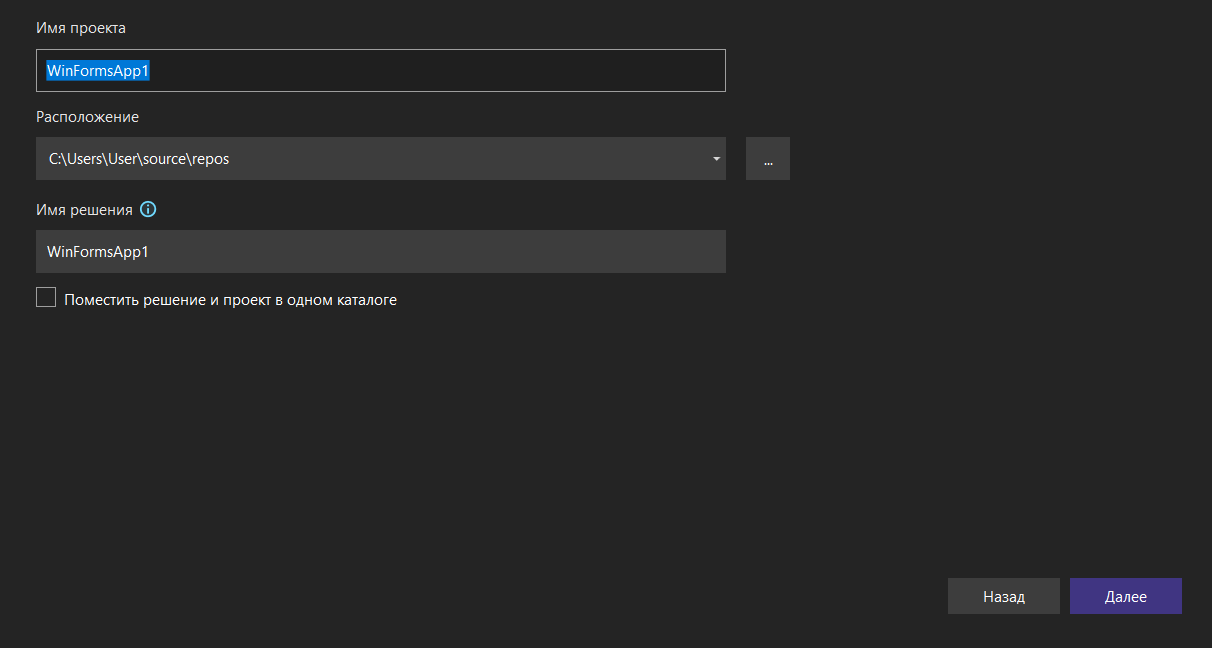


Рис. 75. Имя проекта

1. Платформу оставить неизменной и создать проект;

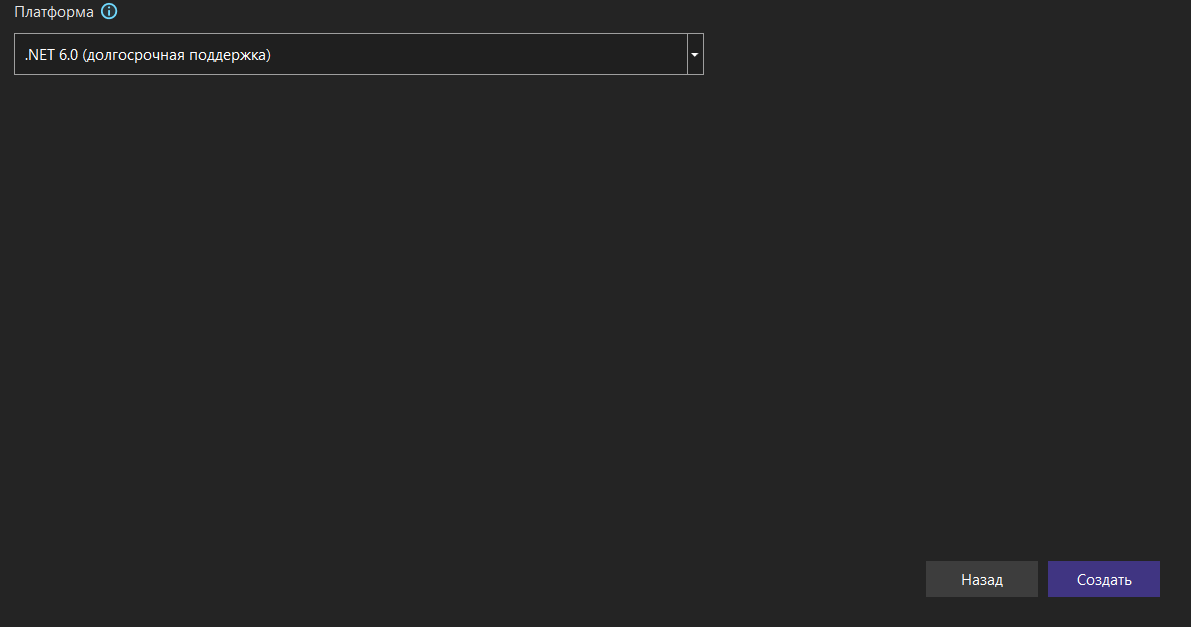


Рис. 76. Платформа

## **Разработка формы авторизации**

В базе данных «Мебельная фабрика» была создана форма авторизации с разделением на пять ролей (заказчик, менеджер, мастер, заместитель директора и директор). В зависимости от введенных данных пользователь будет пользоваться различным функционалом. Для просмотра списка ролей можно нажать на кнопку «Список ролей» в правой нижней части формы авторизации.

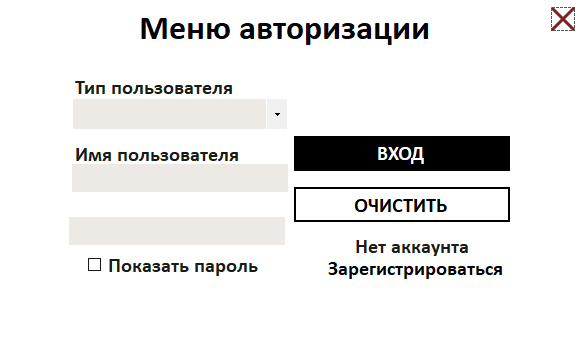


Рис. 77. Авторизация

## Регистрация заказчиков

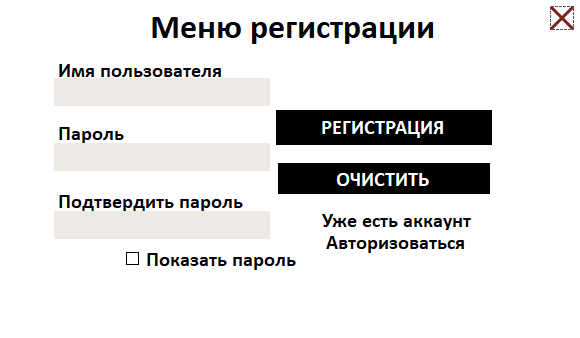


Рис. 78. Регистрация пользователей

## Учет фурнитуры и материалов

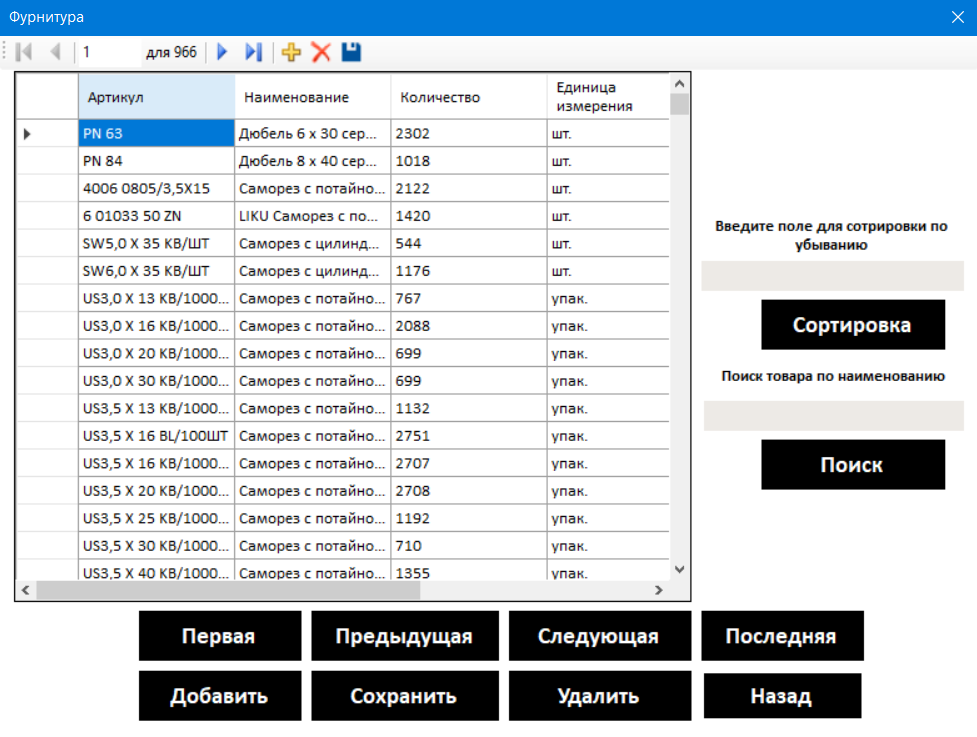


Рис. 79. Учет фурнитуры

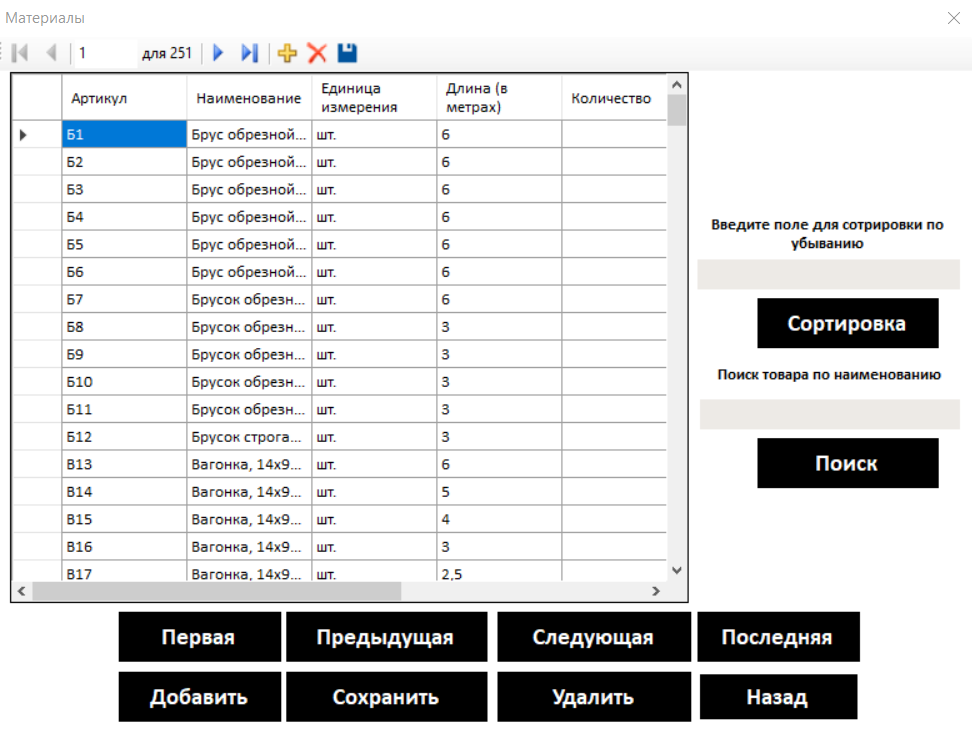


Рис. 80. Учет материалов

# Глава 6. Разработка документации

## Разработка руководства пользователя

**Структура**

Все пользовательские интерфейсы должны иметь заголовок и названием (кроме простых диалогов). Убедитесь, что во всех элементах системы пра­вильно используются выравнивание и пробелы. Также следите за правильной логической группировкой элементов, чтобы систему было легко использовать. Ко всем интерфейсам, которые вы разрабатываете для этой системы, должны применяться одинаковые конст­руктивные решения. Интерфейсами считаются все точки, в которых пользователь взаимо­действует с системой, включая формы, отчеты, письма и т. д.

**Руководство по графике**

**Шрифт:**

Calibri

Допустимые варианты: нормальный, курсив, полужирный.

Размер: 12–20.

В заголовках и выделениях в тексте используйте основные цвета.

Если фон одного из темных основных цветов, используйте белый шрифт.

**Цвета:**

RGB: 255, 255, 255

RGB: 58, 171,74

RGB: 3; 2; 11

RGB: 197; 161; 115

## Разработка диаграммы

### ERD-диаграмма

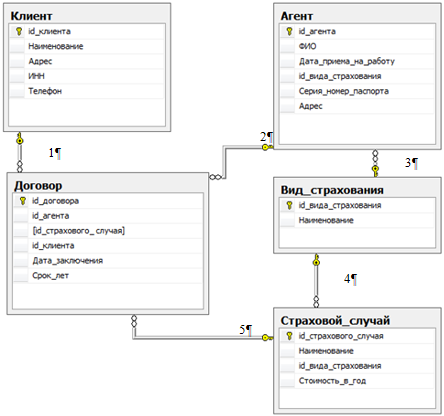


Рис. 81. ERD-диаграмма

### Диаграмма прецедентов

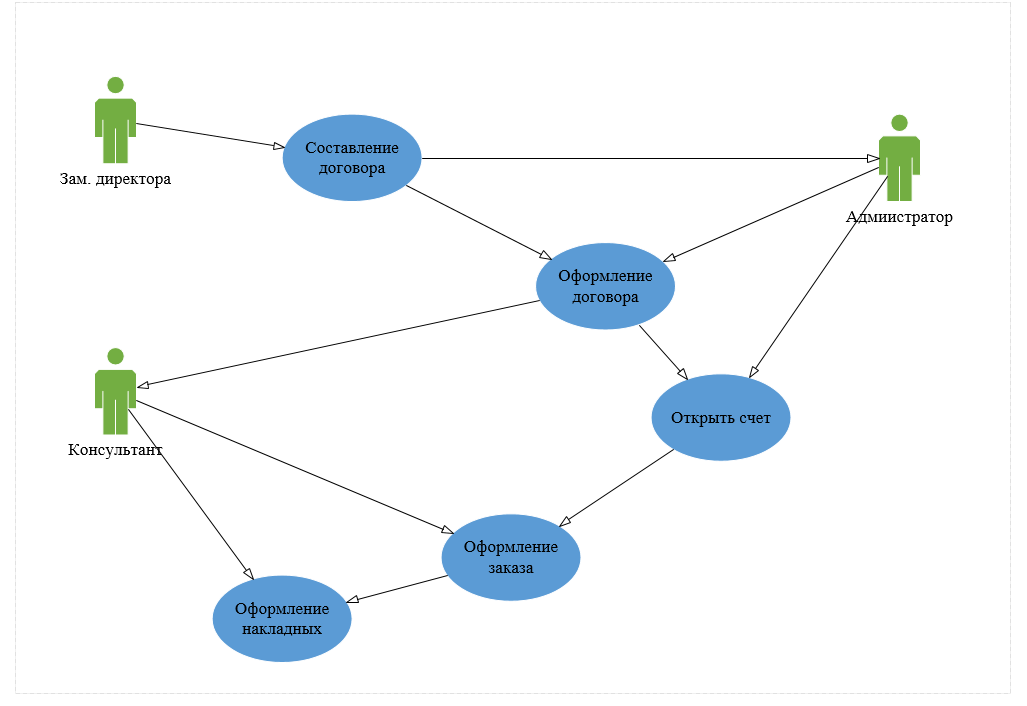


Рис. 82. Диаграмма прецедентов

### Диаграмма деятельности

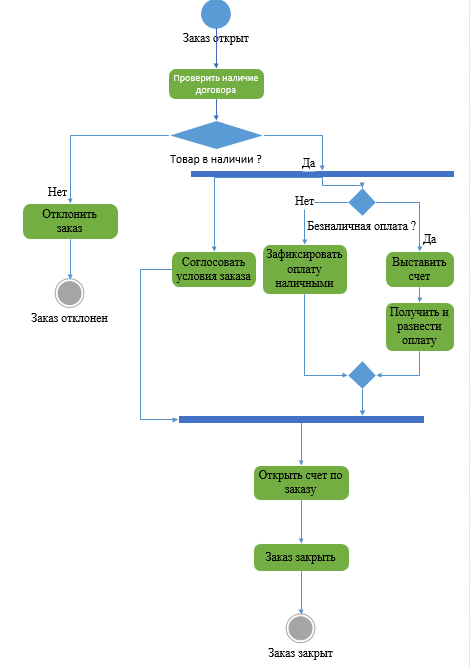


Рис. 83. Диаграмма деятельности

### Диаграмма взаимодействия

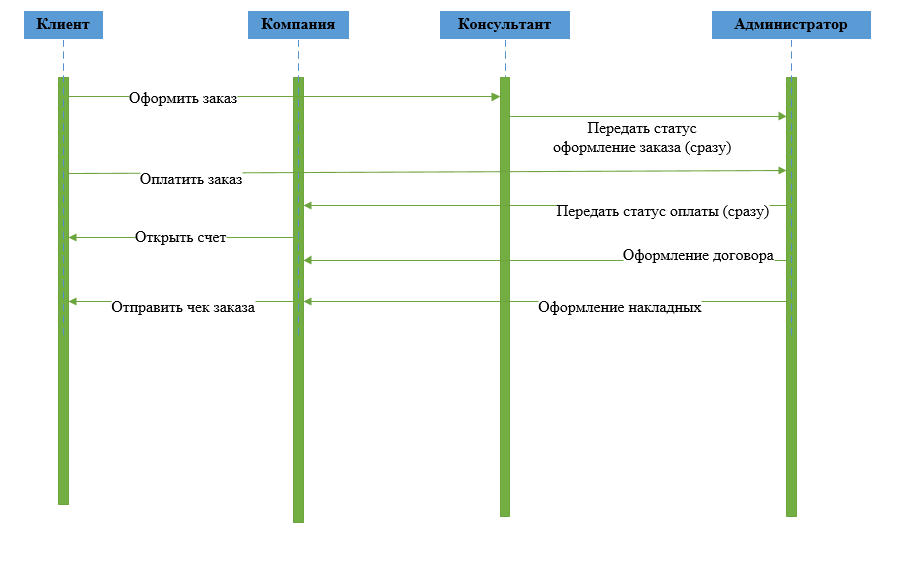


Рис. 84. Диаграмма взаимодействия

### Диаграмма классов

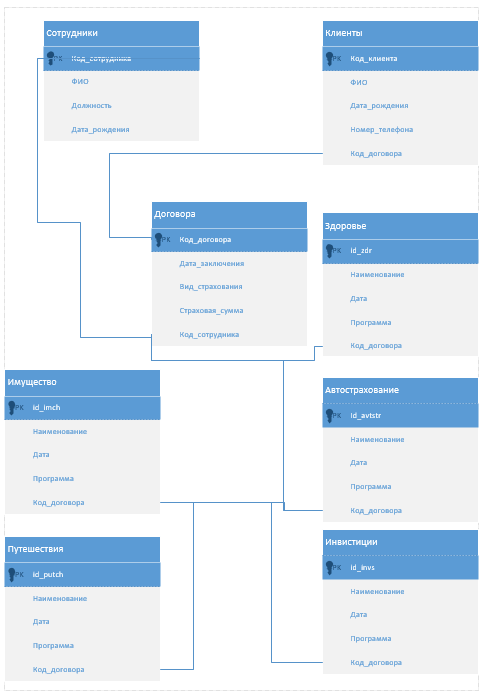


Рис. 85. Диаграмма классов