**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ**

**ТВЕРСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение**

**«Вышневолоцкий колледж»**

**ОТЧЕТ**

По специальности: 09.02.07 «Информационные системы и программирование»

По ПМ.02 «Осуществление интеграции программных модулей»

Выполнил студент группы П-46-18

Невский Данила Александрович

Руководитель проекта:

Яковлева Жанна Сергеевна

г. Вышний Волочек

2022 г.

**Оглавление**

[**ГЛАВА 1. РАБОТА В MS EXCEL КАК В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ** 3](#_Toc96582226)

[**1.1 Разработка простейшей базы данных в программе MS Excel** 3](#_Toc96582227)

[**1.2. Функции ВПР и ГПР** 3](#_Toc96582228)

[**1.3 Операции с диапазонами** 5](#_Toc96582229)

[**ГЛАВА 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО РАБОТЕ С GITHUB** 6](#_Toc96582230)

[**2. Создание репозитория** 14](#_Toc96582231)

[**3. Создание онлайн репозитория на Github.com** 15](#_Toc96582232)

[4**. Добавление репозитория из Git в GitHub** 16](#_Toc96582233)

[**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ** 18](#_Toc96582234)

[**3.1. Моделирование данных. ERD** 18](#_Toc96582235)

[**3.2. Создание базы данных по средствам запросов** 18](#_Toc96582236)

[**3.3. Создание аналитических запросов** 18](#_Toc96582237)

[**ГЛАВА 4. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ В ДЕНЬ С-1** 19](#_Toc96582238)

[**4.1. База данных** 19](#_Toc96582239)

[**4.2. Разработка форм авторизации и поиска** 19](#_Toc96582240)

[**4.3. Разработка форм фильтрации и сортировки** 20](#_Toc96582241)

[**ГЛАВА 5. РАЗРАБОТКА DESKTOP ПРИЛОЖЕНИЙ** 21](#_Toc96582242)

[**5.2. Разработка ERD диаграмм** 21](#_Toc96582243)

[**5.2.1. Работа в MS Visio** 22](#_Toc96582244)

[**5.2.2. Подготовка данных для импорта** 22](#_Toc96582245)

[**5.2.3. Разработка базы данных «Мебельная фабрика»** 22](#_Toc96582246)

[**5.3 Создание приложений** 22](#_Toc96582247)

[**5.4 Разработка формы авторизации** 22](#_Toc96582248)

[**5.5 Регистрация заказчиков** 23](#_Toc96582249)

[**5.6 Учет оборудования** 24](#_Toc96582250)

[**5.7 Учет фурнитуры и материалов** 24](#_Toc96582251)

[**5.8 Список заказов** 25](#_Toc96582252)

[**6.1 Разработка прототипа** 25](#_Toc96582253)

[**6.2 Работа с диаграммами** 26](#_Toc96582254)

[**6.2.1 ERD-Диаграмма** 27](#_Toc96582255)

[**6.2.2 UseCase-Диаграмма** 27](#_Toc96582256)

[**6.2.4 Диаграмма вариантов использования** 28](#_Toc96582257)

[**6.2.5 Диаграмма прецедентов** 28](#_Toc96582258)

[**6.2.5 Диаграмма деятельности** 28](#_Toc96582259)

# **ГЛАВА 1. РАБОТА В MS EXCEL КАК В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ**

# **1.1 Разработка простейшей базы данных в программе MS Excel**

Была разработана база данных на тему «Успеваемость студентов»:

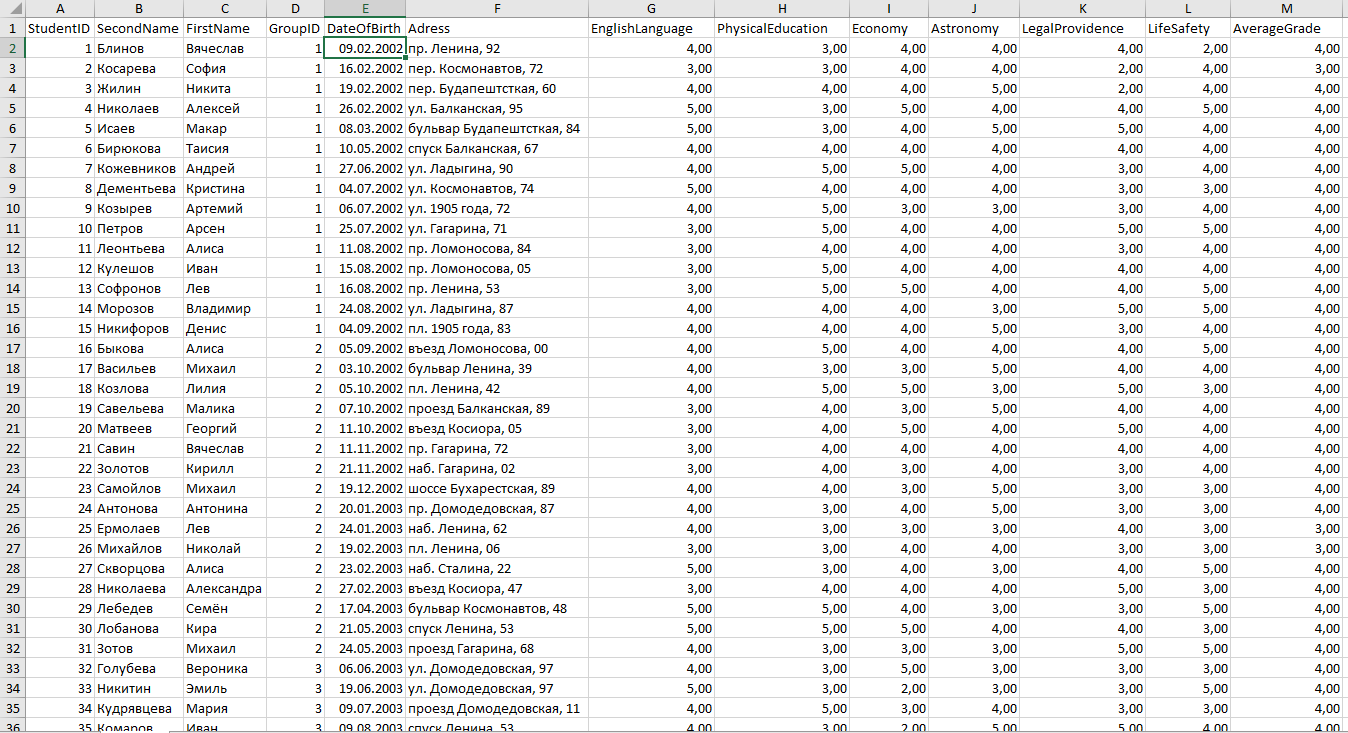


Рис 1. База данных «Успеваемость студентов» в MS EXCEL

# **1.2. Функции ВПР и ГПР**

Для выполнения функций ВПР было создано две таблицы: «Продажа товаров» и «Цены товаров». С их помощью была выполнена функция ВПР – из таблицы «Цены товаров» переносилась информация из поля «Цена, руб» на одноименное поле в таблице «Продажа товаров». Перенос осуществлялся с помощью команде «ВПР» в разделе «Ссылки и массивы»

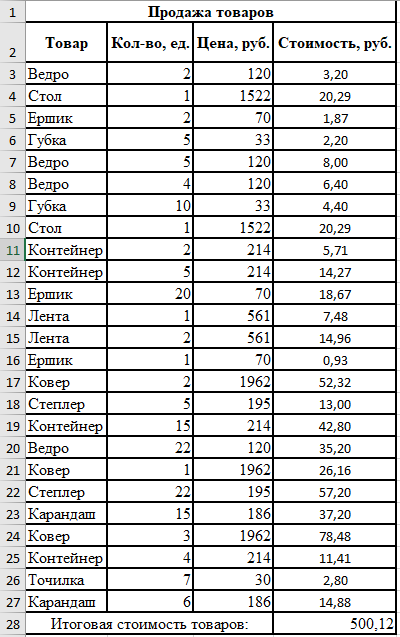


Рис 2. Таблица «Продажа товаров»

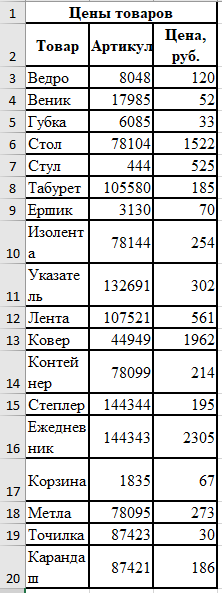


Рис 3. Таблица «Цены товаров»

Для реализации функции ГПР были использованы таблицы «Продажа фруктов» и «Цены фруктов», после транспонирования появилась новая таблица, объединявшая значения из обеих таблиц.

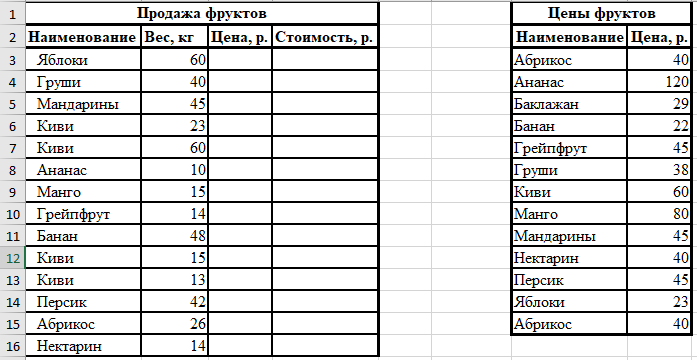


Рис 4. Таблицы «Продажа фруктов» и «Цены фруктов»

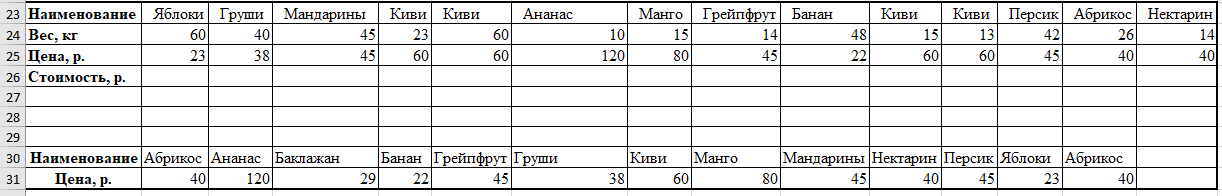


Рис 5. Результат функции «ГПР»

# **1.3 Операции с диапазонами**

Для выполнения операций с диапазонами необходимо было рассчитать итоговую стоимость всех товаров из таблицы «Прайс 2008 г.».



Рис 6. Результат работы с диапазонами.

# **ГЛАВА 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО РАБОТЕ С GITHUB**

**2.1 Разработка руководства пользователя**

После запуска установочного файла «Git-2.34.1-32-bit» перед нами появляется окно установки, где нужно выбрать путь, в котором будет хранится устанавливаемое приложение.

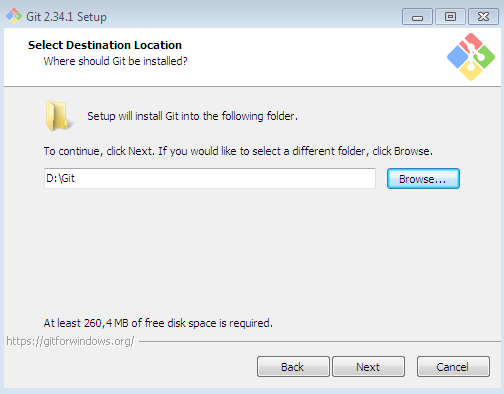


Рис. 7. Окно выбора места установки программы «GIT»

После выбора местоположения и нажатия клавиши «Next» перед нами открывается следующее окно, в котором нужно выбрать дополнительные компоненты при установке:

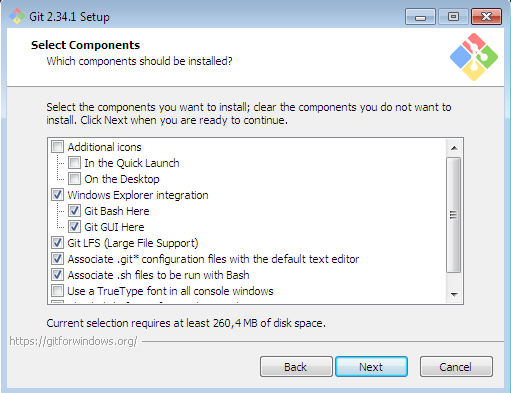


Рис 8. Окно выбора дополнительных компонентов при установке

Следующее меню представляет из себя окно выбора – добавлять ли программу в папку автозапуска или нет. Мы нажимаем клавишу «Don’t create a Start Menu folder», что можно перевести дословно как «Не создавать папку в меню Пуск».

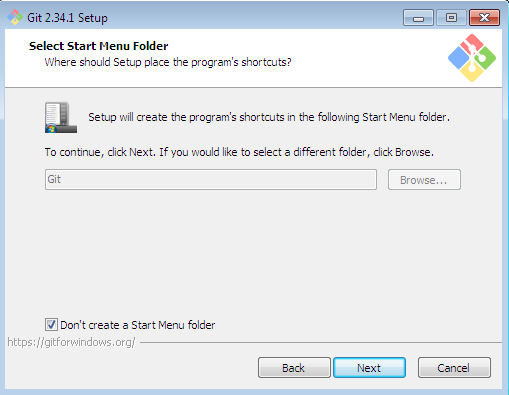


Рис 9. Окно выбора создания папки в меню «Пуск»

После выбора создания папки в меню пуск, программа предлагает выбрать редактор для программы по умолчанию. Мы останавливаемся на предложенном редакторе «Vim».

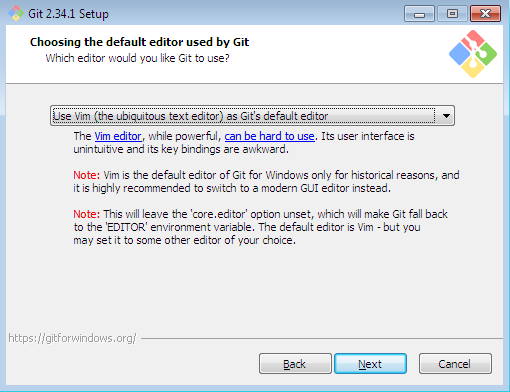


Рис 10. Окно выбора редактора для GIT

Следующую окно предлагает выбрать способ корректировки имени начальной ветки в новых репозиториях. Можно позволить программе самой решать, нажав «Let Git decide» или же выбрать значение по умолчанию, нажав «Override the default branch name for new repositories» и ввести textbox название начальной ветки в репозиториях. Мы выбираем первый вариант, чтобы программа сама выбирала имя начальной ветки в новых репозиториях.

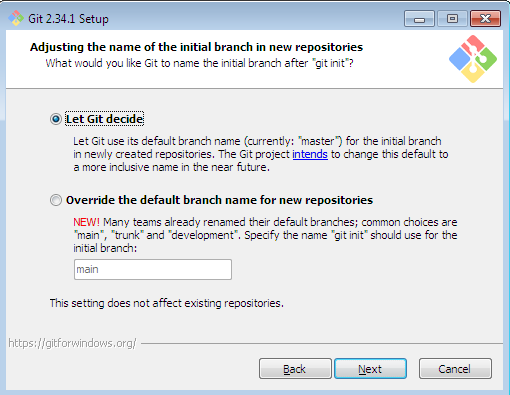


Рис 11. Окно выбора корректировки имени начальной ветки в новых репозиториях

Далее идет окно выбора способа использования Git:

Use Git from Git Bash only

Переменная PATH не модифицируется и работа с Git возможна только через специализированную оболочку, которая называется Git Bash.

Use Git from the Windows Command Prompt

В этом случае происходит минимальная модификация переменной окружения PATH, которая позволит работать с Git через командную строку Windows. Работа через Git Bash также возможна.

Use Git and optional Unix tools from the Windows Command Prompt

В переменную PATH вносится значительное количество модификаций, которые позволят, в рамках командной строки Windows, использовать как Git, так и утилиты Unix, которые поставляются вместе с дистрибутивом Git.

Мы выбираем рекомендованный пункт, а именно второй

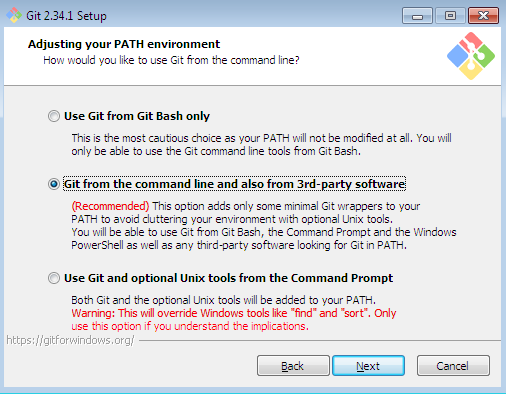


Рис 12. Окно выбора способа использования Git

Следующее окно предлагает выбрать библиотеку, которая будет использоваться при подключении по протоколу HTTPS.

OpenSSL - сертификаты сервера будут проверяться с использованием Unix-файла ca-bundle.crt.

Windows Secure Channel - сертификаты сервера будут проверяться с использованием стандартной библиотеки Windows.

Мы выбираем «OpenSSL», т.к. собираемся использовать эти протоколы для работы с «GitHub».

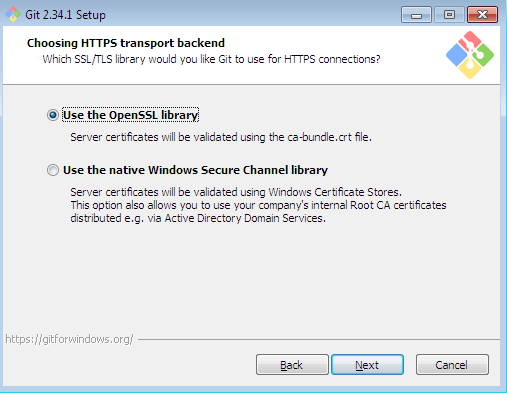


Рис 13. Окно выбора библиотеки протокола HTTPS

В следующем слайде пользователь выбирает способ обработки окончания строк. Тут важно выбрать первый и по совместительству стандартный вариант, а именно «Checkout Windows-style, commit Unix-style line endings». Это значение гарантирует, что Git преобразует LF в CRLF при проверке текстовых файлов. При выполнении текстовых файлов CRLF также преобразуется в LF.

Это мера совместимости для защиты новых строк в текстовых файлах, что позволяет легко работать с текстовыми файлами в Windows и на платформах Unix.

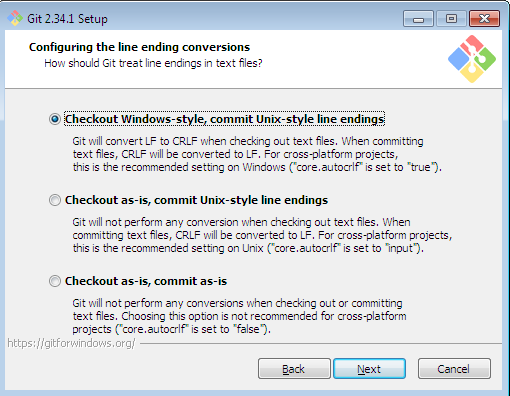


Рис 14. Окно выбора способа обработки окончания строк

Дальше появляется окно конфигурации используемого терминала, где:

MinTTY - терминал Unix, который мы и будем выбирать.

Windows - стандартный терминал Windows.

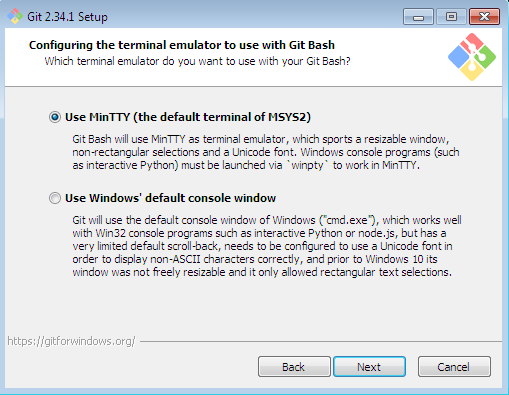


Рис 15. Окно выбора конфигурации терминала

В этом окне выбирается поведение по умолчанию функции «Get Pull». Оставляет значение по умолчанию и продвигаемся далее.

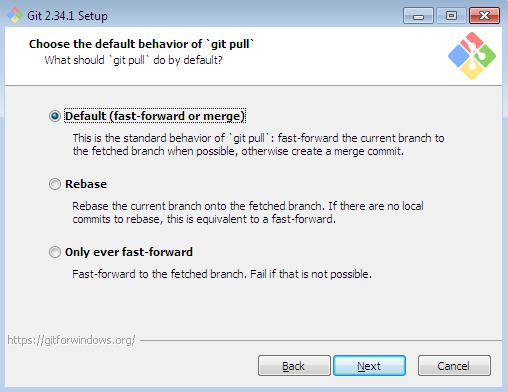


Рис 16. Выбор поведения по умолчанию функции «Get Pull»

Следующее окно предлагает нам выбрать помощника по учетным данным. Соглашаемся и выбираем представленный вариант «Git Credential Manager».

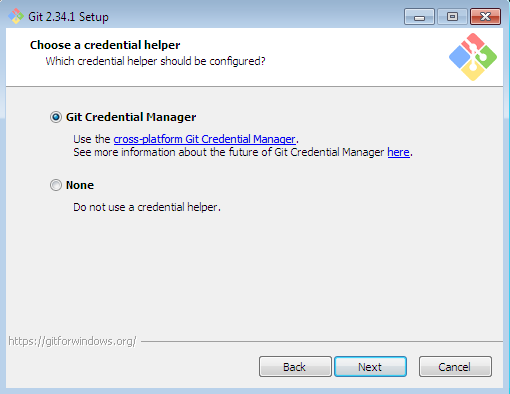


Рис 17. Выбор помощника по учетным данным

Далее нам предстоит выбирать дополнительные настройки программного обеспечения. Мы выбираем «Enable file system caching», т.к. она улучшает производительность.

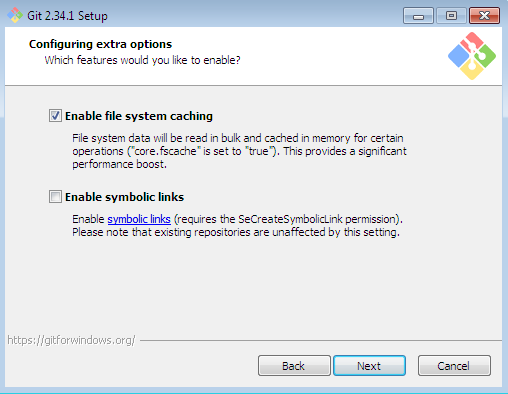


Рис 18. Выбор дополнительных настроек в программном обеспечение

Далее можно выбрать экспериментальные функции программы, мы в свою очередь пропускаем данный пункт и ничего не выбираем.

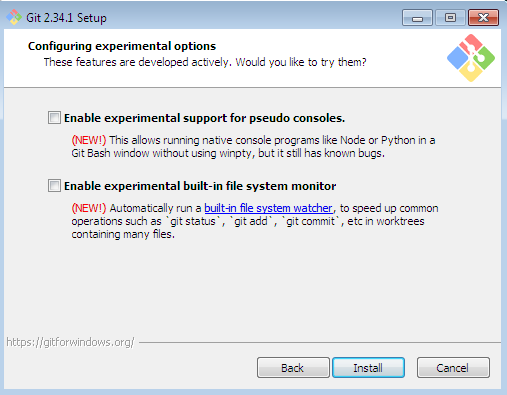


Рис 19. Выбор экспериментальных настроек программного обеспечения

На последней форме происходит установка приложения, после чего высвечивается данное окно, которое свидетельствует о том, что программа успешно установлена и готова к работе.

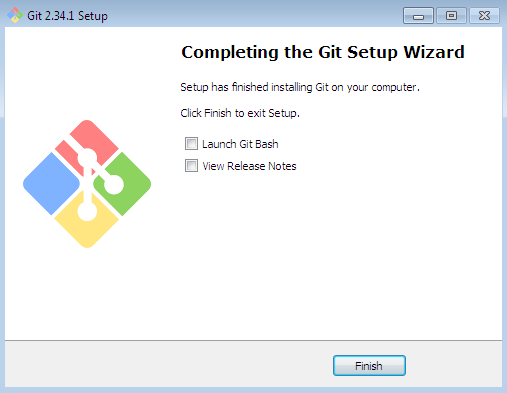


Рис 20. Окно завершения установки Git

# **2. Создание репозитория**

Открываем Visual Studio Code и открываем там созданную заранее папку в удобном месте с названием «GitText».

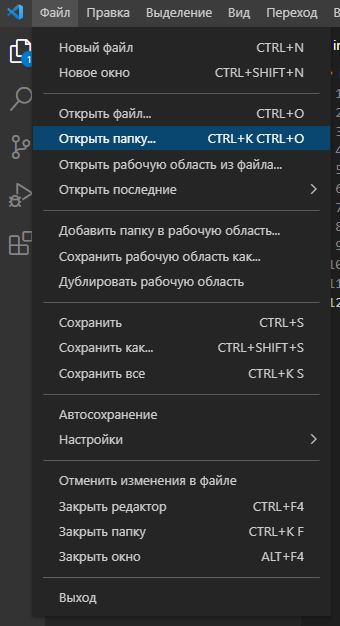


Рис 21. Окно выбора папки для репозитория

После чего создаем необходимые для репозитория файлы index.html и main.css при помощи клавиши «Создать файл»

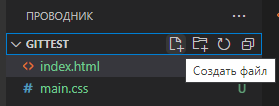


Рис 22. Окно создания файлов для репозитория

После создания файлов мы наконец-то можем создать репозиторий. Для этого нам необходимо нажать на клавишу «Инициализировать репозиторий» и выбрать папку с нашими файлами.

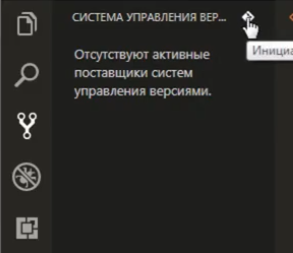


Рис 23. Инициализация репозитория

После этого в нашей папке с файлами появится скрытая папка .git, она и будет являться нашим репозиторием изменений в проекте.



Рисунок 24. Скрытая папка .git

# **3. Создание онлайн репозитория на Github.com**

Для того чтобы начать создания репозитория нужно зайти на сайт [**https://github.com/**](https://github.com/), авторизоваться или создать новую учетную запись, а затем нажать зеленую клавишу «New» в окне «Repositories», после чего откроется окно создания нового репозитория.

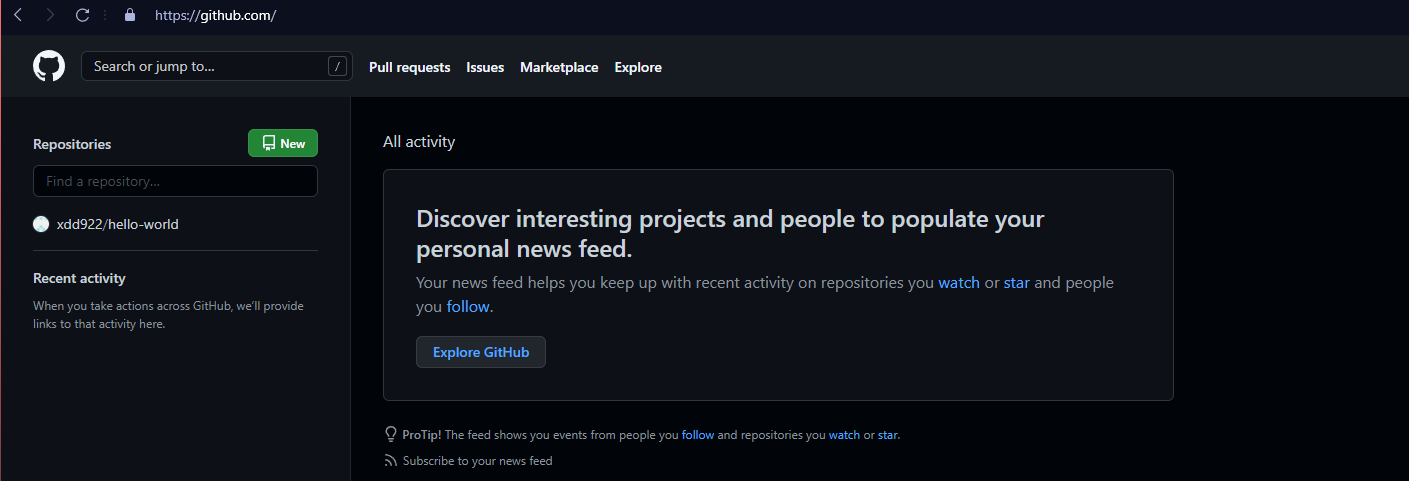


Рис 25. Сайт <https://github.com/>

После чего открывается окно создания репозитория, в котором можно выбрать следующие параметры:

* + Название репозитория
  + Описание репозитория
  + Доступ к репозиторию для пользователей (Public – для всех; Private – вы представляете доступ к репозиторию)
  + Добавить README файл (Файл с описанием вашего проекта)
  + Выбрать лицензию
  + Добавить .gitignore (Выбрать какие файлы не будут отслеживаться в списке шаблонов)

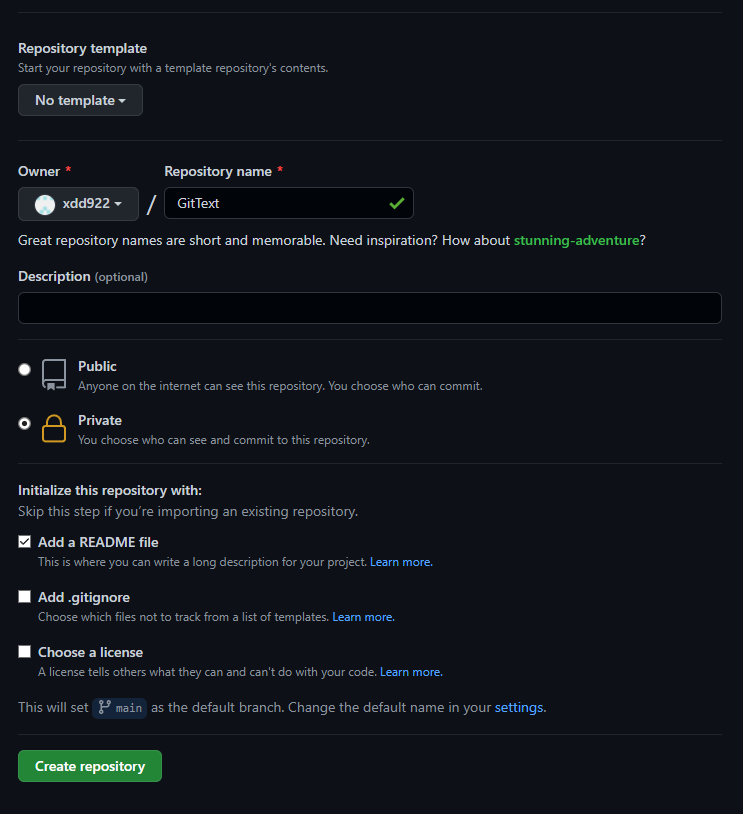


Рис 26. Окно выборов параметров репозитория

После выбора всех необходимых параметров необходимо нажать кнопку «Create repository», после чего создания можно считать завершенным.

# 4**. Добавление репозитория из Git в GitHub**

Для начала выполняем простую команду «git init» - она инициализирует наш репозиторий на компьютере и показывает его расположение.



Рис 27. Выполнение команды «git init»

После чего нужно произвести команду «git add .» - она начинает отслеживание изменений. После этого пишем «git commit -m "first commit"» - выполняет роль сбора информации об измененных файлах с комментарием «first commit». После этого программа говорит что не определила автора запроса, из-за чего нам необходимо ввести свой e-mail адрес и имя пользователя на GitHub при помощи «git config --global user.email ["Ваша почта"»](mailto:you@example.com) и «git config --global user.name "Ваше имя"» соответственно.

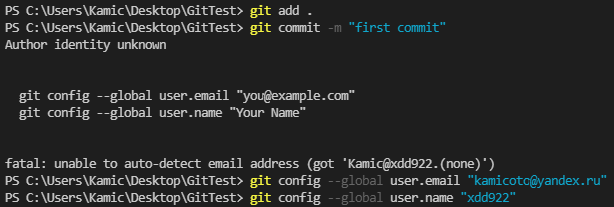


Рис 28. Процесс отправки коммита и авторизации пользователя

Для создания репозитория онлайн вводим «git remote add origin [https://github.com/xdd922/GitText»](https://github.com/xdd922/GitText) - создает репозиторий на GitHub.

Последним штрихом будет написание команды «git push -u origin master», которая отправляет наши изменения на GitHub, перед тем проведя авторизацию в вашем браузере.

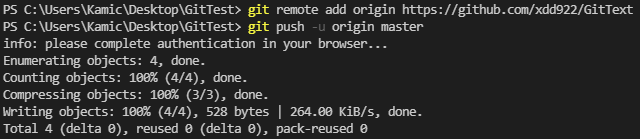


Рис 29. Синхронизация с репозиторием и отправка данных

# **ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

# **3.1. Моделирование данных. ERD**

Для моделирования данных была разработана база данных «Туристическая фирма». На основе этой базы была создана ERD-Диаграмма

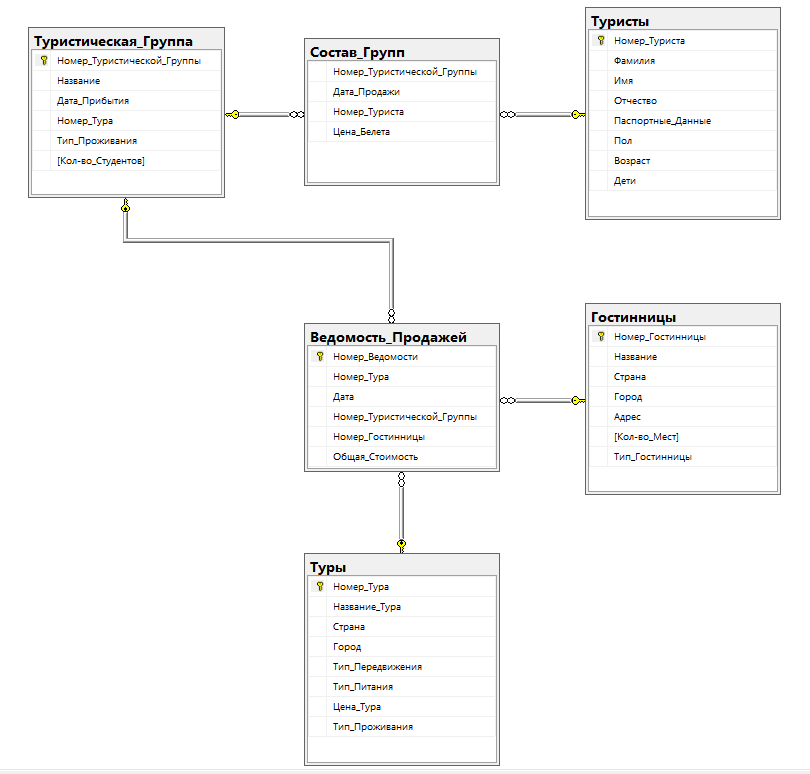


Рис 30. ERD-Диаграмма б.д «Туристическая фирма»

# **3.2. Создание базы данных по средствам запросов**

# **3.3. Создание аналитических запросов**

Был представлен аналитический вопрос: «Сколько минимально курсов необходимо удалить , чтобы объем очистки был не менее 5000 Мб данных. Для решения был разработан запрос, выводящий минимальное количество курсов для удаления.

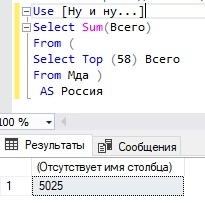


Рис 31. Результат аналитического запроса

# **ГЛАВА 4. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ В ДЕНЬ С-1**

# **4.1. База данных**

Была разработана база данных «Стоматологическая клиника».

Рисунок 32. ERD-Диаграмма б.д. «Мебельная фабрика»



Рис 32. ERD-Диаграмма б.д «Стоматологическая клиника»

# **4.2. Разработка форм авторизации и поиска**

Форма авторизации была представлена в минималистичном окне, в котором есть только кнопка входа и выхода, а так же поля ввода для пароля и логина пользователя.

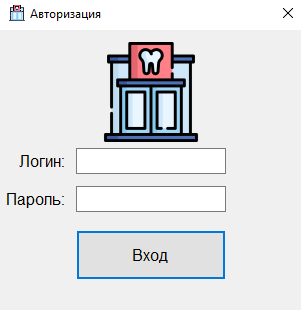


Рис 33. Форма авторизации пользователя

Функция поиска представлена в основной форме табличного вида. Необходимо ввести необходимую для поиска фразу, после чего нажать клавишу «Найти», после чего нужный результат подсветиться синим цветом, как указано на примере:

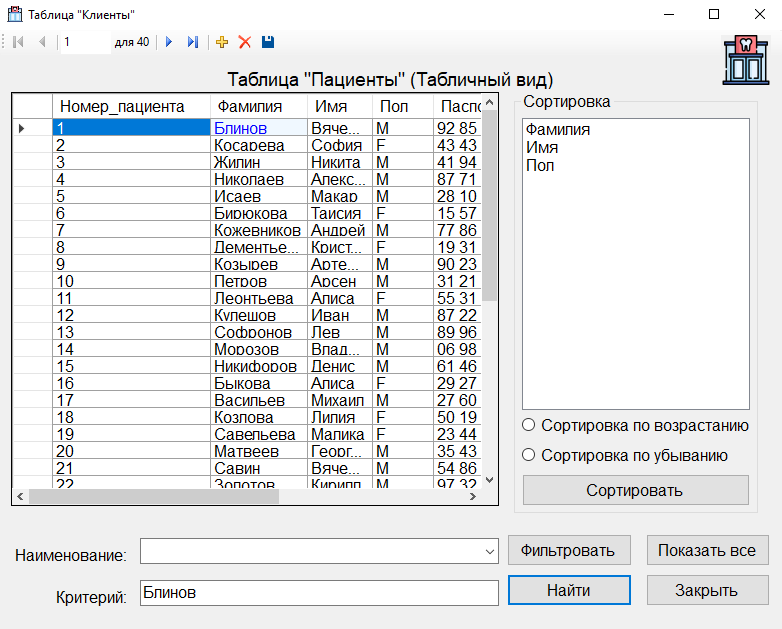


Рис 34. Пример функционала поиска по таблице

# **4.3. Разработка форм фильтрации и сортировки**

Фильтрация и сортировка представлены небольшой панелью в правой части окна табличного вида. Для сортировки необходимо выбрать какой стоблец необходимо сортировать и выбрать тип сортировки с помощью RadioButton (Сортировка по возрастанию или убыванию). Для фильтрации необходимо выбрать определенное наименование и ввести критерий поиска, после чего нажать «Фильтровать». Сразу же покажется результат фильтрации, который можно сбросить, нажав на клавишу «Показать все».

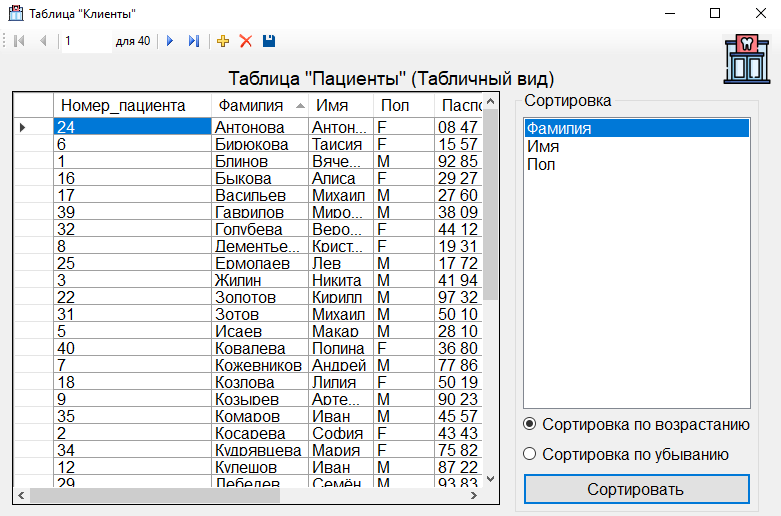


Рис 35. Результат сортировки по возрастанию столбца «Фамилия»

# **ГЛАВА 5. РАЗРАБОТКА DESKTOP ПРИЛОЖЕНИЙ**

# **5.2. Разработка ERD диаграмм**

Была разработана модель EDR-Диаграмма Мебельной фабрики в Visio

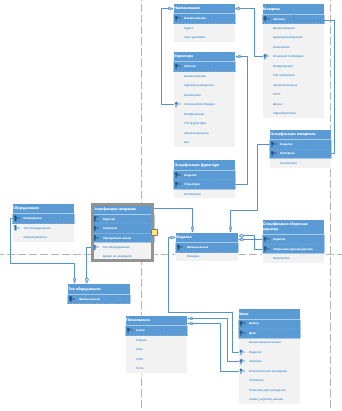


Рис 36. ERD-Диаграмма б.д «Мебельная фабрика»

# **5.2.1. Работа в MS Visio**

Для создания ERD-Диаграммы понадобилось программное обеспечение Microsoft Visio. Для начала работы необходимо нажать на клавишу «Файл», после чего создать новый файл с типом «Нотация UML для моделирования баз данных». После чего необходимо перетаскивать в рабочую область Сущности, атрибуты и создавать между ними связи. Результат работы представлен на рисунке 36, одним пунктом выше.

# **5.2.2. Подготовка данных для импорта**

Для разработки базы данных было представлено три Excel файла, в которых находились данные. Для корректного импортирования в базу данных их необходимо было откорректировать. В итоге у нас получилось 3 .csv файла (разделители – запятые), которые были готовы к импорту в Microsoft SQL.

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

Готовые .csv файлы мы импортируем в базу данных, обновляя её новыми данными. После чего можно считать разработку базы данных «Мебельная фабрика» завершенной.

# **5.3 Создание приложений**

Для начала работы по созданию приложения, необходимо определится с средой разработки нашей программы. В данном случае мы будем использовать Visual Studio 2022, на платформе WindowsForms .NET на языке программирования C#.

# **5.4 Разработка формы авторизации**

При запуске приложения программе необходимо определится с доступом к функционалу пользователя при помощи простой формы авторизации. Она состоит из двух полей для ввода логина и пароля пользователя, а также выплывающим списком значений, где пользователю нужно выбрать роль. После ввода данных и проверки их в базе данных, приложение переносит пользователя на следующую форму, которая варьируется от прав доступа к приложению.

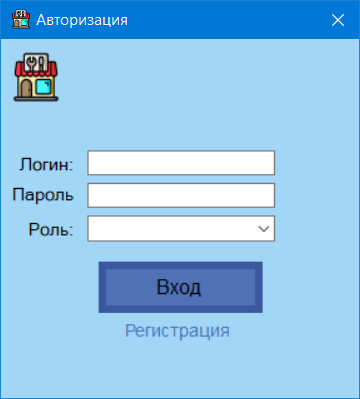


Рис 37. Внешний вид формы авторизации

# **5.5 Регистрация заказчиков**

Была разработана система регистрации пользователя. Для регистрации необходимо перейти на соответствующую форму, нажав на текст «Регистрация» в начальной форме приложения. После чего необходимо ввести ФИО пользователя, а также его логин, пароль и роль.

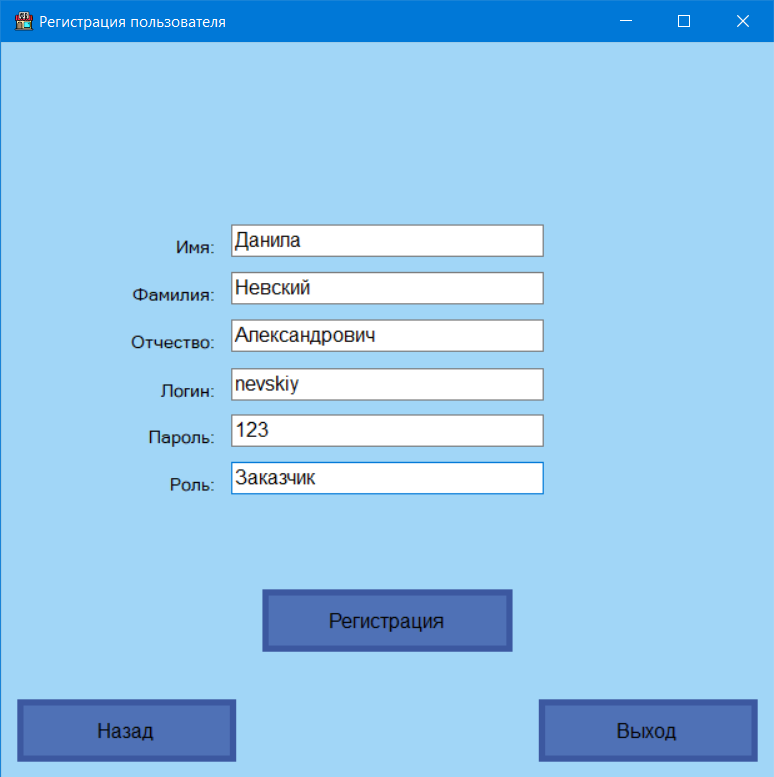


Рис 38. Внешний вид формы регистрации

# **5.6 Учет оборудования**

Для учета оборудования были разработаны две формы в программе, они обе взаимосвязаны и дополняют друг друга данными. Соответственно «Типы оборудования» дополняют таблицу «Оборудование», в которой подробно расписано оборудования для работы на мебельной фабрике.

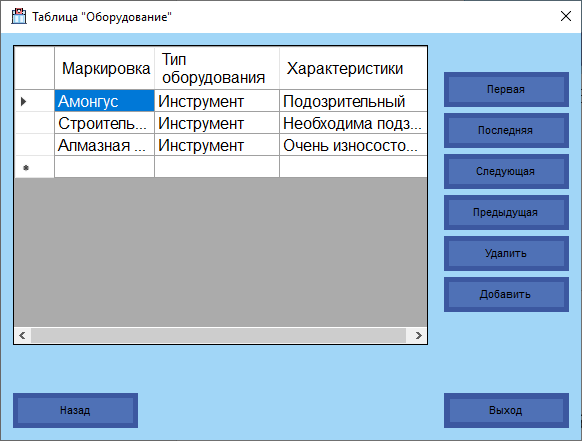


Рис 39. Таблица «Оборудование»

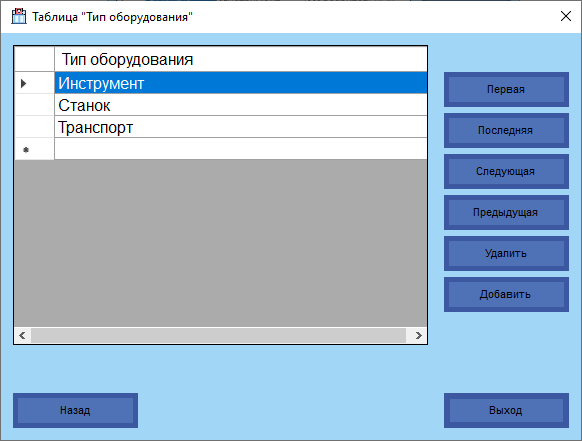


Рис 40. Таблица «Тип оборудования»

# **5.7 Учет фурнитуры и материалов**

Была разработана система учета фурнитуры и материалов для складов. В таблице «Материалы» можно отслеживать всю необходимую информацию об доступных материалах, а также дополнять её новыми данными, если у пользователя есть на это права доступа.

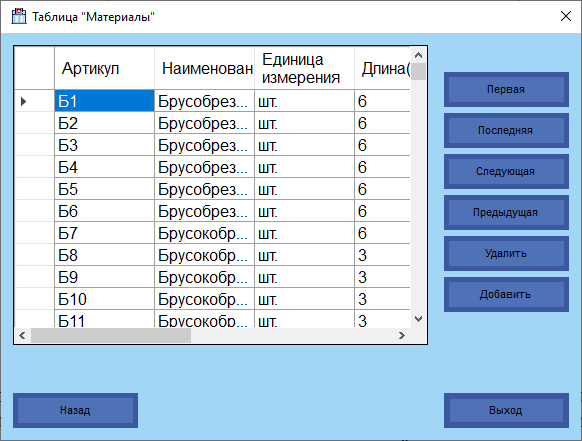


Рис 41. Таблица «Материалы»

Таблица «Фурнитура» предоставляет для пользователя данные о наличие товаров, их количестве, цене, артикуле, единицах измерения и типах товаров.

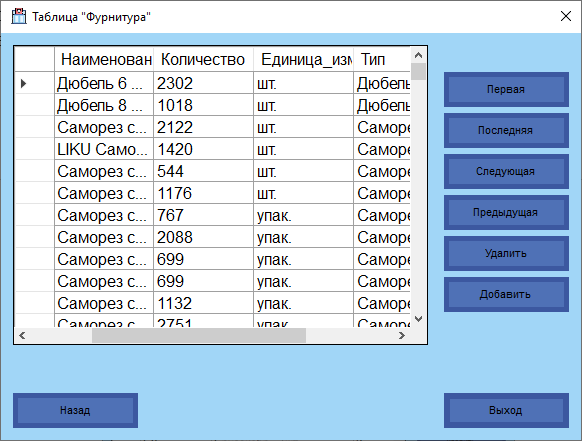


Рис 42. Таблица «Фурнитура»

# **5.8 Список заказов**

Была разработана таблица со списками заказов, в которых можно отследить информация об договорах, датах принятия и сдачи заказов, а так же минимальные данные об производимом изделии.

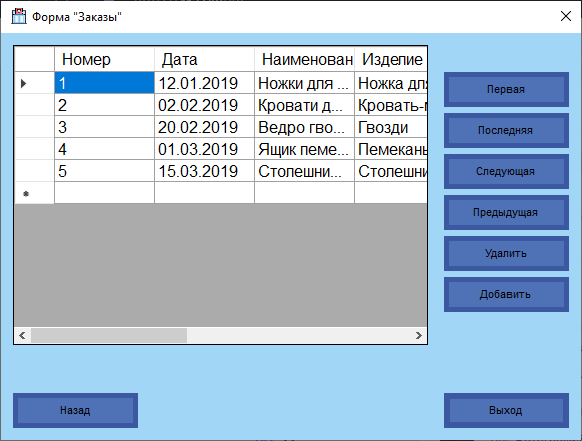


Рис 43. Таблица «Заказы»

# **ТЕМА 6. РАЗРАБОТКА ДОКУМЕНТАЦИИ**

# **6.1 Разработка прототипа**

Структура: Все пользовательские интерфейсы должны иметь заголовок с логотипом (логотип.ico) и названием. Убедитесь, что во всех элементах системы правильно используется выравнивание и пробелы.

**Руководство по графике:**

Шрифт: Arial

Допустимые варианты: обычный, курсив, полужирный. Размер: 10-18.

В заголовках и выделениях в тексте используйте основные цвета.

Применяемые цвета:

Для текста:

RGB: 255; 255; 255;

RGB: 0; 0; 0;

Основные:

RGB: 161; 214; 247;

RGB: 134; 204; 239;

RGB: 79; 113; 182;

RGB: 225; 228; 225;

# **6.2 Работа с диаграммами**

Для создания диаграмм необходимо специализированное программное обеспечение, в нашем случае это Microsoft Visio, включенное в состав Office. Для начала работы необходимо запустить программу, после чего она предложит выбрать шаблон для нового проекта.

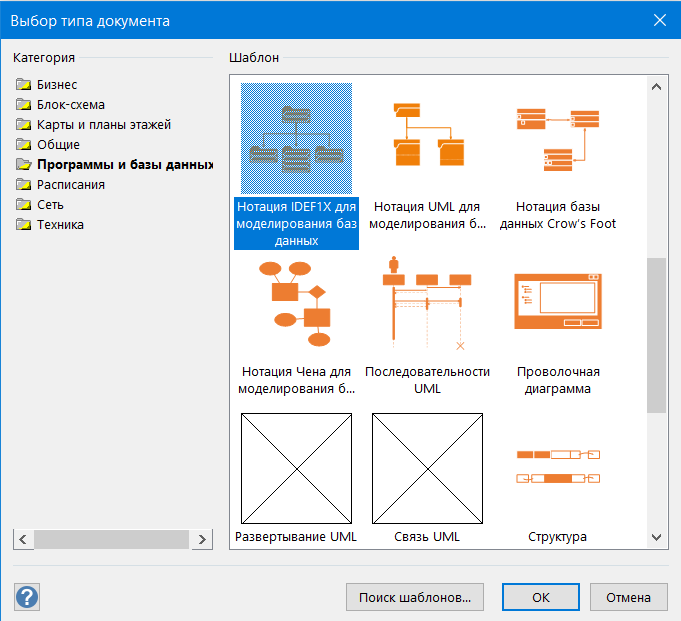


Рисунок 44. Окно выбора шаблона в Visio

После выбора в панели «Фигуры» появятся новые фигуры для создания диаграммы базы данных.

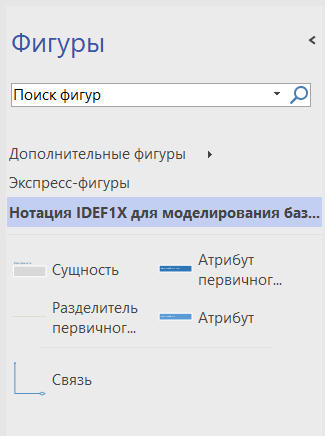


Рис 45. Окно выбора фигур в Visio

С помощью этих фигур можно приступать к созданию необходимых диаграмм, начиная задачу с ERD-Диаграммы, в создании которой используются сущности и атрибуты, а так же связи между ними.

# **6.2.1 ERD-Диаграмма**



Рис 46. ERD-Диаграмма «Успеваемость студентов»

# **6.2.2 UseCase-Диаграмма**



Рис 47. UseCase-Диаграмма

# **6.2.4 Диаграмма вариантов использования**

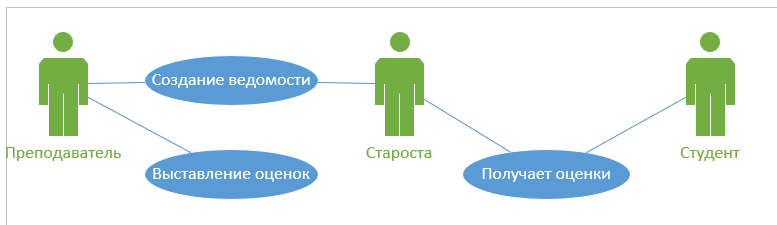


Рис 48. Диаграмма вариантов использования

# **6.2.5 Диаграмма прецедентов**

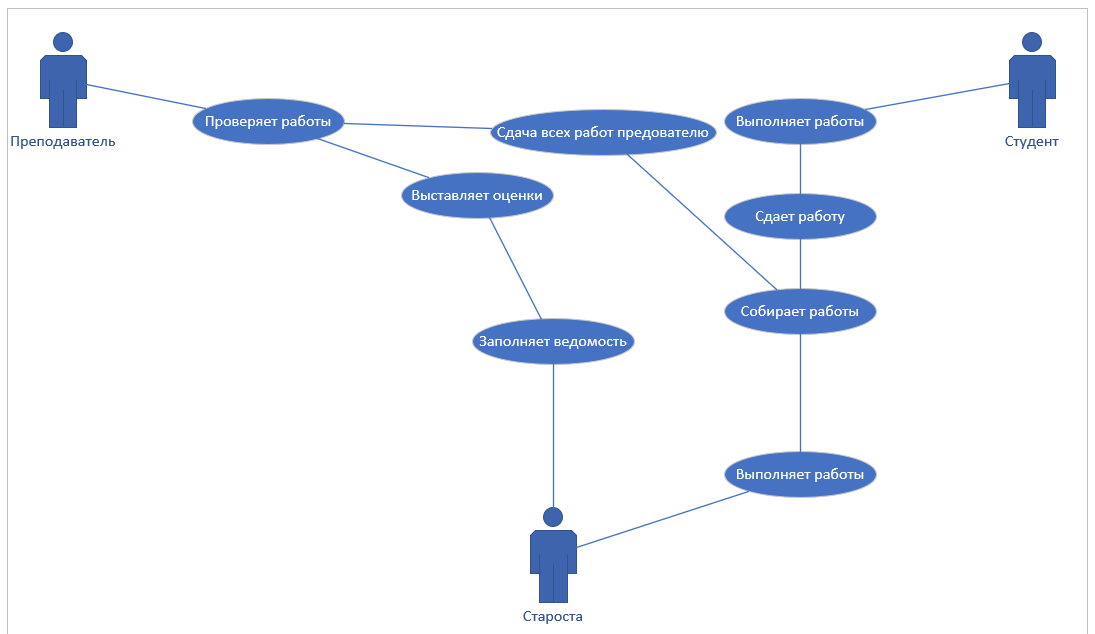


Рис 49. UML-Диаграмма прецендентов

# **6.2.5 Диаграмма деятельности**

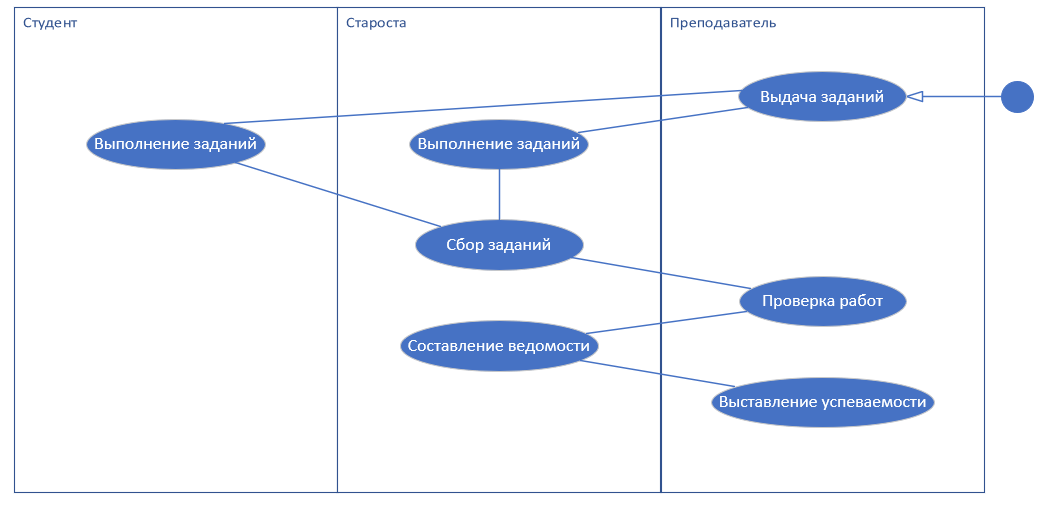


Рис 50. UML-Диаграмма деятельности