**МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ**

**ТВЕРСКОЙ ОБЛАСТИ**

**Государственное бюджетное профессиональное образовательное учреждение**

**«Вышневолоцкий колледж»**

**ОТЧЕТ**

На тему: **«Итоговая учебная практика»**

По специальности: 09.02.07 «Информационные системы и программирование»

По ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем»

Выполнил студент группы П-46-18

Невский Данила Александрович

Руководитель проекта:

Яковлева Жанна Сергеевна

г. Вышний Волочек

2022 г.

Оглавление

**Элементы оглавления не найдены.**

# **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»:

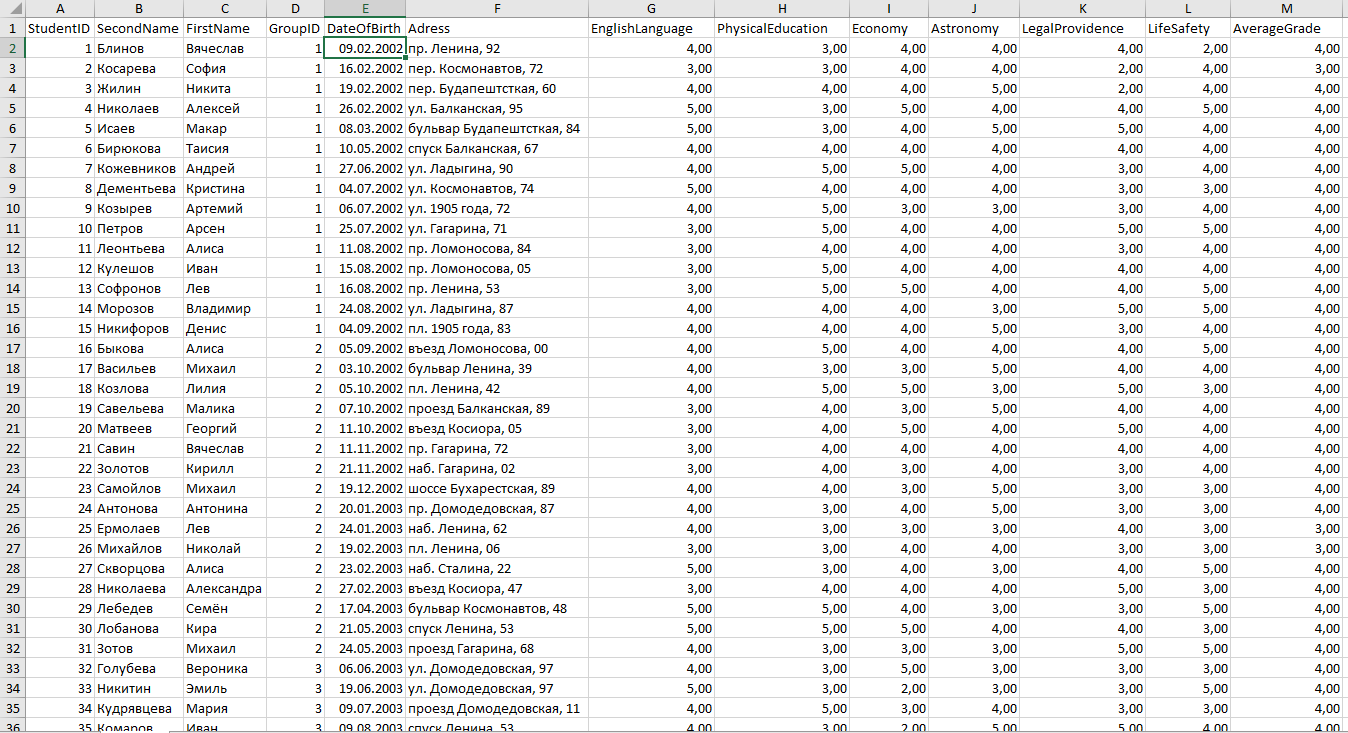


Рисунок 1. База данных «Успеваемость студентов» в MS EXCEL

# 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»

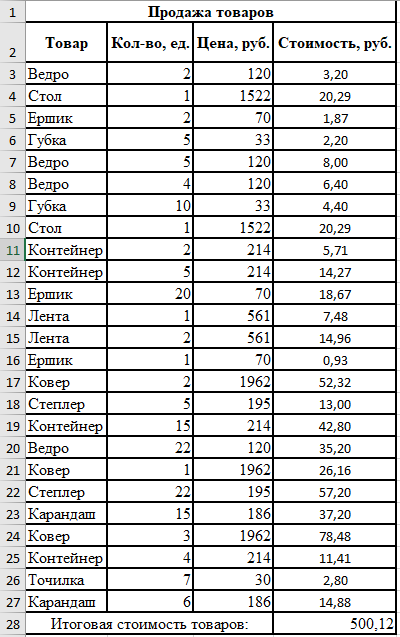


Рисунок 2. Таблица «Продажа товаров»

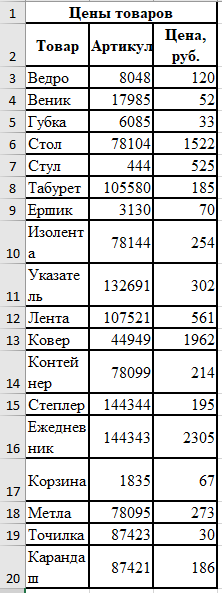


Рисунок 3. Таблица «Цены товаров»

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

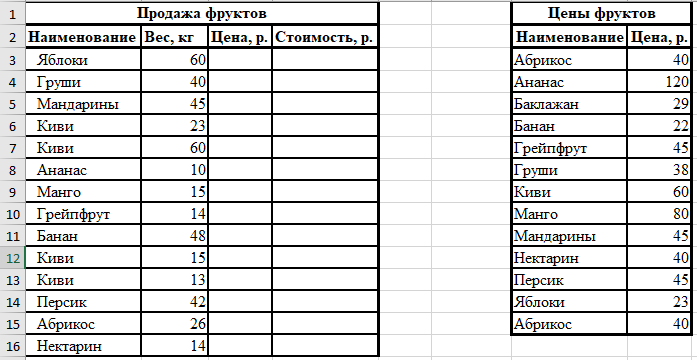


Рисунок 4. Таблицы «Продажа фруктов» и «Цены фруктов»

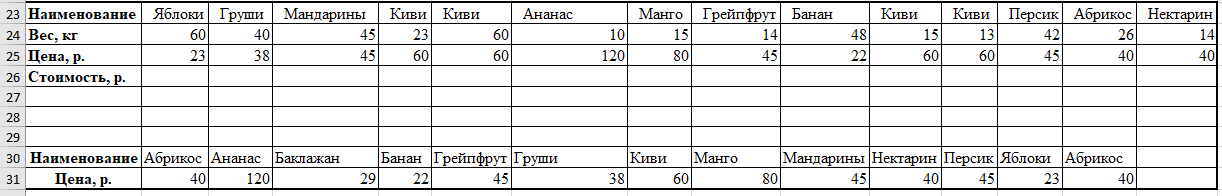


Рисунок 5. Результат функции «ГПР»

# **1.3 Операции с диапазонами**

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



Рисунок 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.

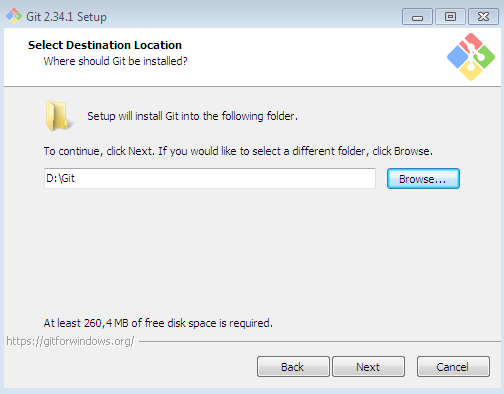


Рисунок 7. Окно выбора места установки программы «GIT»

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

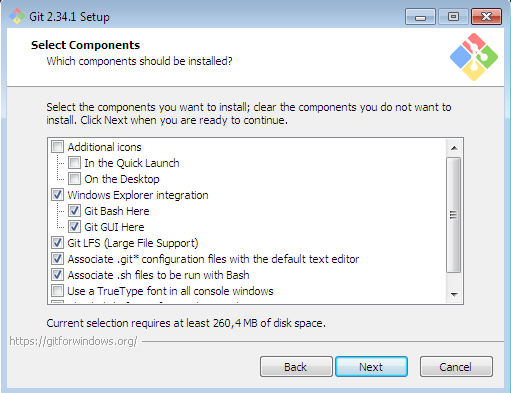


Рисунок 8. Окно выбора дополнительных компонентов при установке

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».

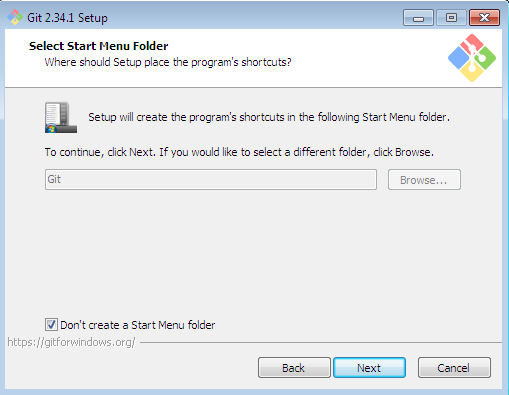


Рисунок 9. Окно выбора создания папки в меню «Пуск»

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

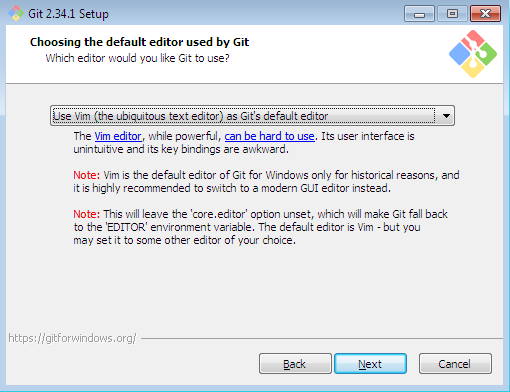


Рисунок 10. Окно выбора редактора для GIT

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.

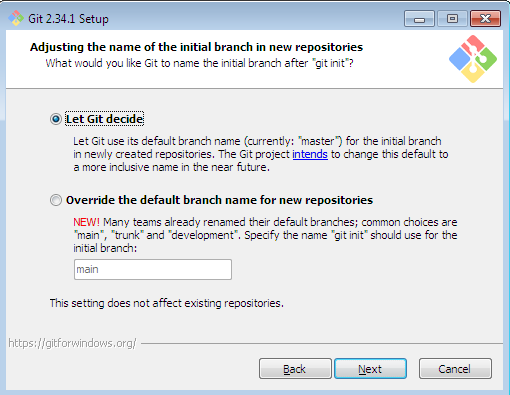


Рисунок 11. Окно выбора корректировки имени начальной ветки в новых репозиториях

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.

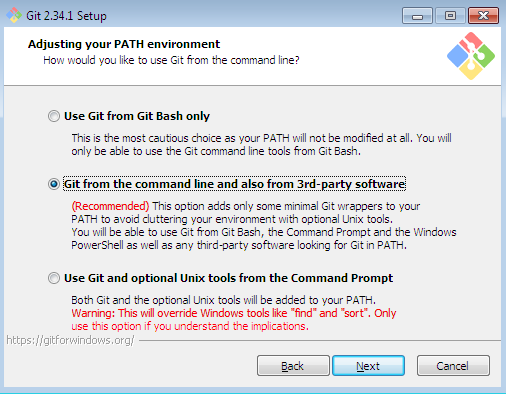


Рисунок 12. Окно выбора способа использования Git

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».

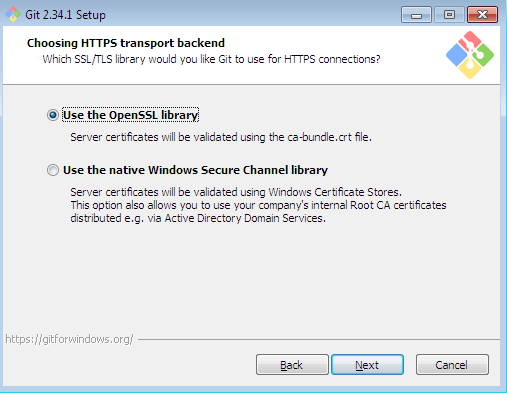


Рисунок 13. Окно выбора библиотеки протокола HTTPS

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.

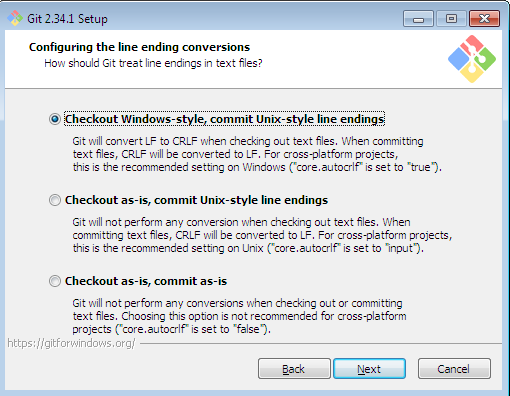


Рисунок 14. Окно выбора способа обработки окончания строк

Дальше появляется окно конфигурации используемого терминала, где:

* **MinTTY** - терминал Unix, который мы и будем выбирать.
* **Windows** - стандартный терминал Windows.

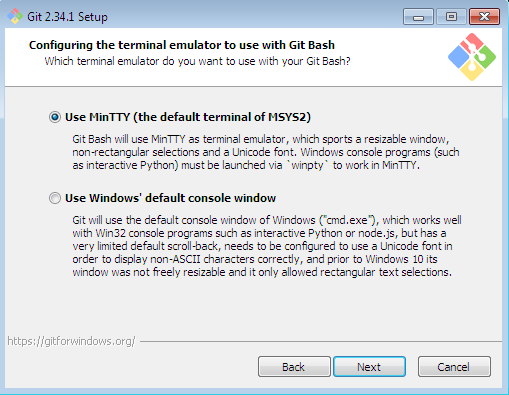


Рисунок 15. Окно выбора конфигурации терминала

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

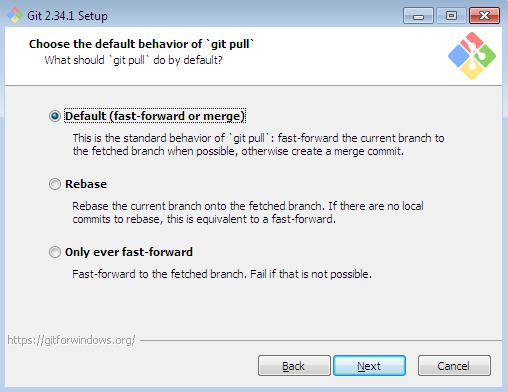


Рисунок 16. Выбор поведения по умолчанию функции «Get Pull»

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

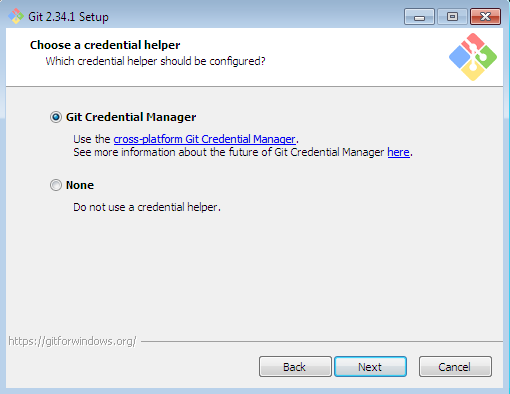


Рисунок 17. Выбор помощника по учетным данным

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

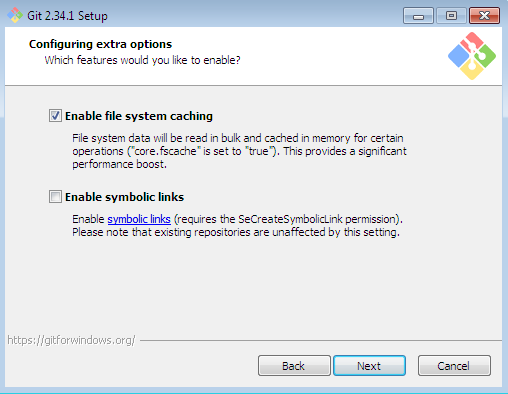


Рисунок 18. Выбор дополнительных настроек в программном обеспечение

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

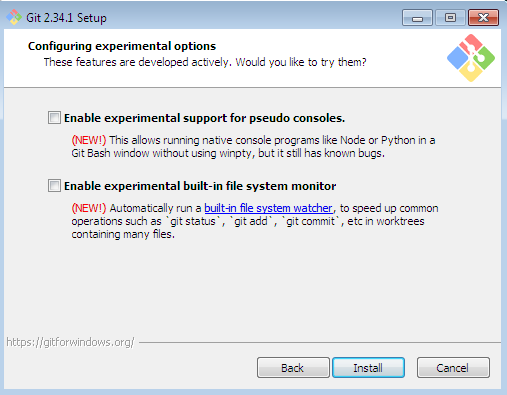


Рисунок 19. Выбор экспериментальных настроек программного обеспечения

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

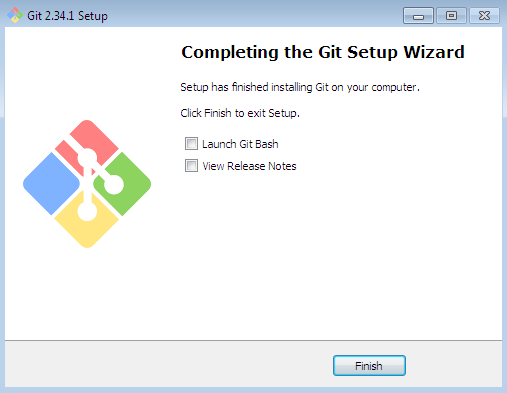


Рисунок 20. Окно завершения установки Git

# **2. Создание репозитория**

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

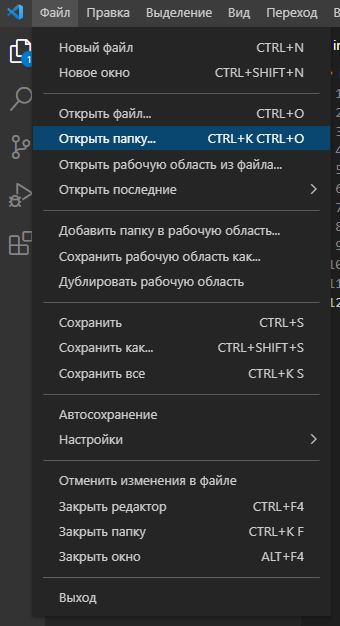


Рисунок 21. Окно выбора папки для репозитория

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

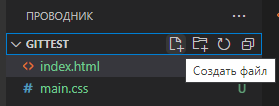


Рисунок 22. Окно создания файлов для репозитория

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

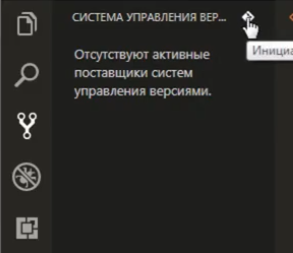


Рисунок 23. Инициализация репозитория

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.

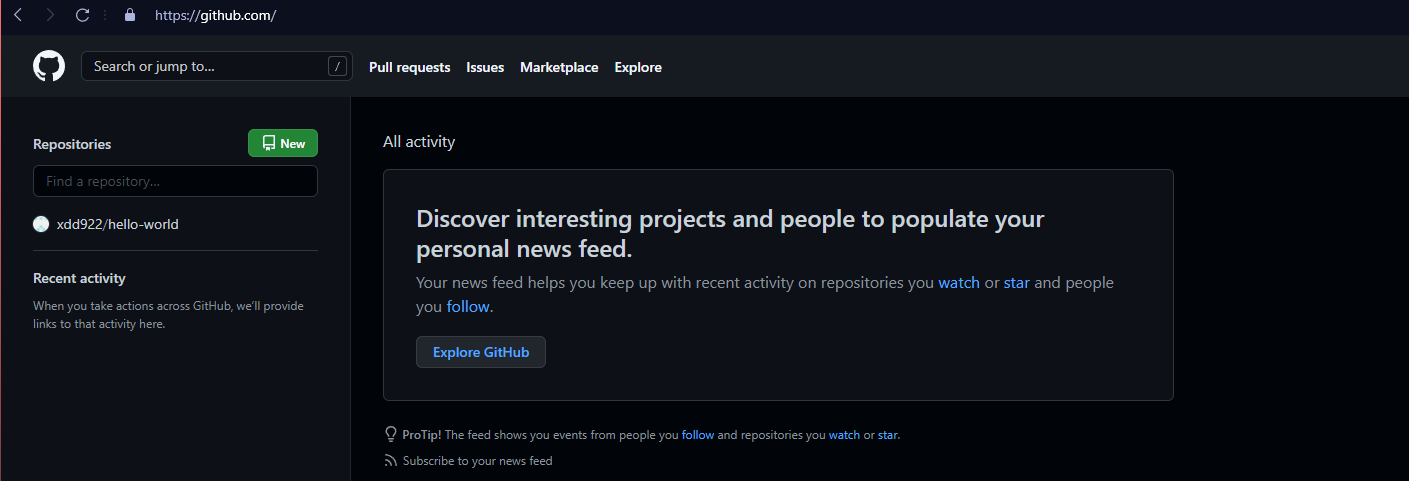


Рисунок 25. Сайт <https://github.com/>

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)

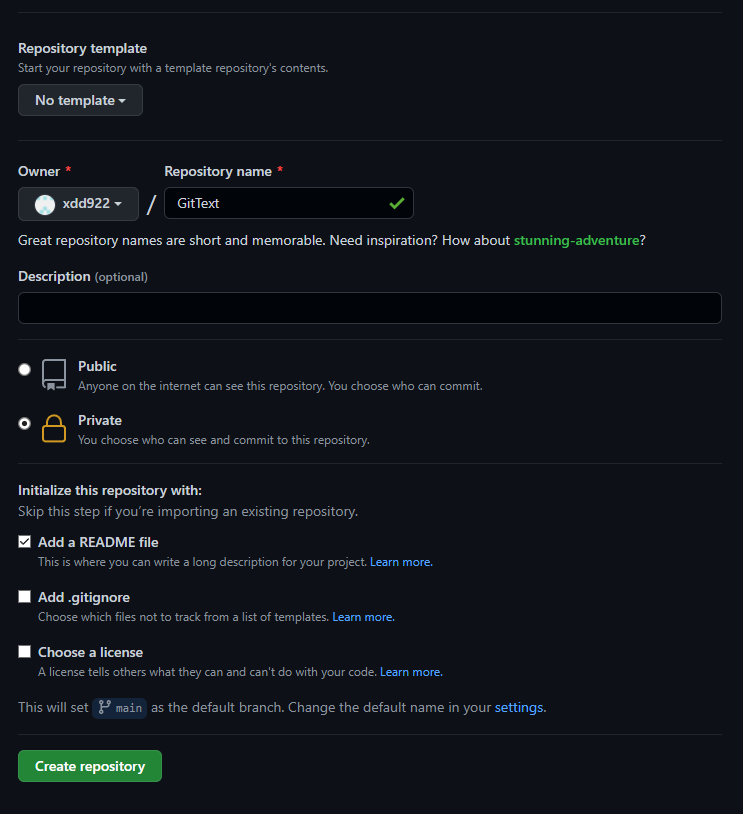


Рисунок 26. Окно выборов параметров репозитория

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.



Рисунок 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.

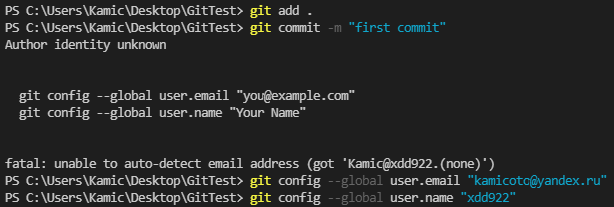


Рисунок 28. Процесс отправки коммита и авторизации пользователя

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.

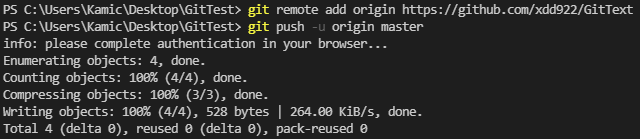


Рисунок 29. Синхронизация с репозиторием и отправка данных

# **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

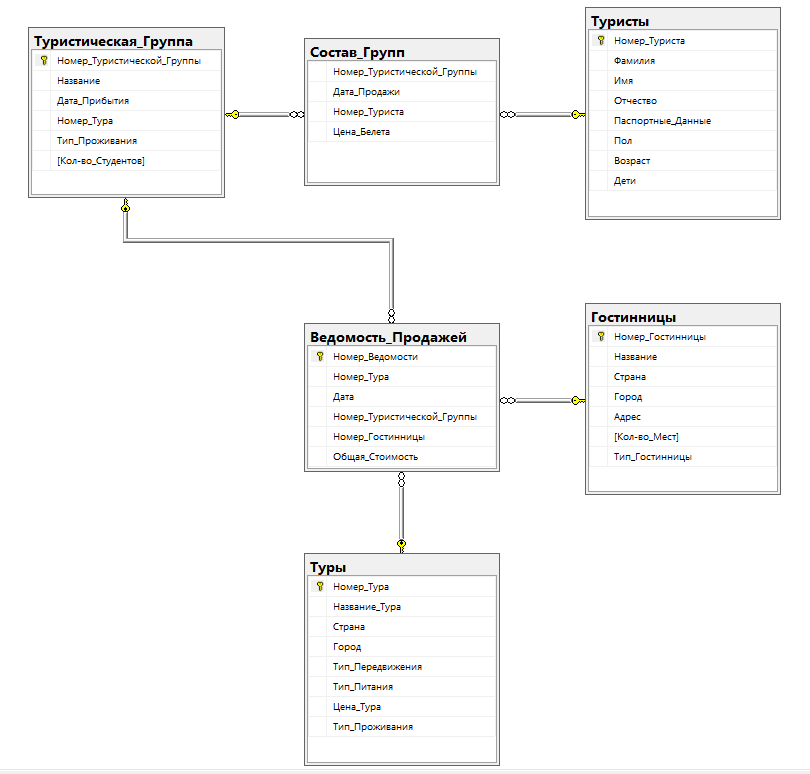


Рисунок 30. ERD-Диаграмма б.д «Туристическая фирма»

# **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.

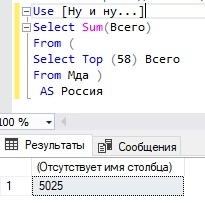


Рисунок 31. Результат аналитического запроса

# **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»



Рисунок 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.

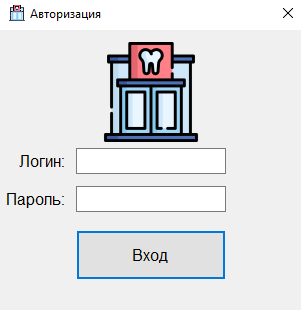


Рисунок 33. Форма авторизации пользователя

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:

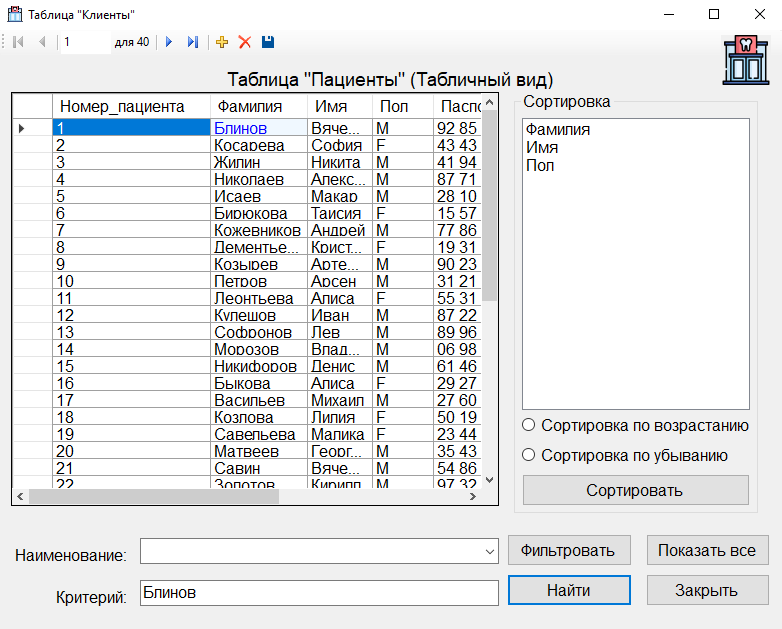


Рисунок 34. Пример функционала поиска по таблице

# **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».

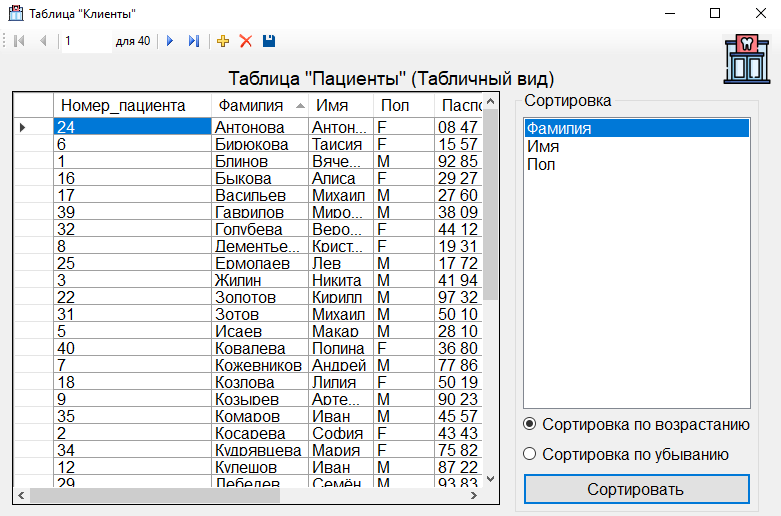


Рисунок 35. Результат сортировки по возрастанию столбца «Фамилия»

# **ГЛАВА 5. РАЗРАБОТКА DESKTOP ПРИЛОЖЕНИЙ**

# **5.2. Разработка ERD диаграмм**

Была разработана модель EDR-Диаграмма Мебельной фабрики в Visio

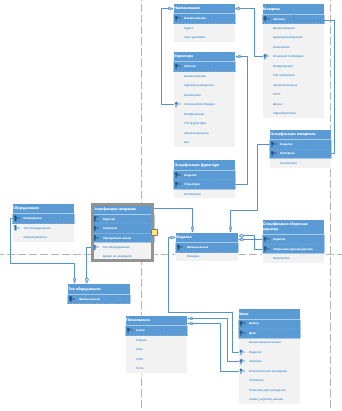


Рисунок 36. ERD-Диаграмма б.д «Мебельная фабрика»

# **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.

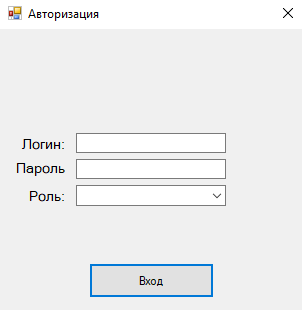


Рисунок 37. Внешний вид формы авторизации