Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

Нижегородский институт управления

Кафедра Информатики и информационных технологий

ОТЧЕТ О ПРОДЕЛАННОЙ РАБОТЕ по курсу

«Программная инженерия»

Выполнила: студентка группы ИБ1-321

Яшина Евгения Леонидовна

Нижний Новгород

2026 г.

**Оглавление**

[Цель работы: 3](#_Toc8895)

[Задачи работы: 3](#_Toc23792)

[Оборудование и ПО: 3](#_Toc21377)

[Ход работы: 4](#_Toc16212)

[Теоретическая часть: 4](#_Toc4213)

[1. Понятие системы контроля версий Git 4](#_Toc5554)

[2. Типы репозиториев Git 4](#_Toc14542)

[2.1 Локальный репозиторий 4](#_Toc8437)

[2.2 Удалённый репозиторий 4](#_Toc30269)

[3. Основные типы файловых состояний в Git 5](#_Toc1686)

[4. Основные команды Git и их назначение 5](#_Toc14504)

[5. Назначение SSH-ключа 6](#_Toc32268)

[6. Ветвление в Git 6](#_Toc19521)

[Практическая часть: 6](#_Toc13043)

[1. Регистрация на GitHub 6](#_Toc17824)

[2. Создание репозитория 8](#_Toc18751)

[3. Настройка SSH‑ключа (для безопасной аутентификации). 9](#_Toc7072)

[4. Инициализация локального репозитория 11](#_Toc7221)

[5. Подключение удалённого репозитория и отправка изменений. 12](#_Toc17661)

[6. Работа с файлами на GitHub 12](#_Toc22018)

[7. Синхронизация локального и удалённого репозиториев 14](#_Toc8200)

[8. Изучение истории изменений. 15](#_Toc11556)

[Ответы на контрольные вопросы: 17](#_Toc23798)

[Дополнительное задание: 17](#_Toc15874)

[Вывод: 20](#_Toc10119)

# Цель работы:

Освоить базовые операции работы с системой контроля версий Git и облачным сервисом GitHub: регистрацию, создание репозитория, настройку SSH‑ключей, базовые команды Git, редактирование файлов и управление изменениями.

# Задачи работы:

* ·Разобраться, что такое Git и для чего он используется.
* Изучить различия между локальным и удалённым репозиторием.
* Освоить основные команды Git и понять их назначение.
* Зарегистрироваться на платформе GitHub.
* Создать удалённый репозиторий.
* Настроить SSH-ключ для безопасного подключения.
* Создать локальный репозиторий и добавить в него файлы.
* Выполнить фиксацию изменений с помощью команд git add и git commit.
* Подключить локальный репозиторий к удалённому и отправить изменения (git push).
* Выполнить синхронизацию репозиториев с помощью команды git pull.
* Изучить работу с Git в среде разработки IntelliJ IDEA или OpenIDE.
* Сделать вывод о полученных навыках.

# Оборудование и ПО:

* Персональный компьютер с ОС Windows 11.
* Установленный Git Bush.
* Браузер Chrome.
* Командная строка.

# Ход работы:

## Теоретическая часть:

### 1. Понятие системы контроля версий Git

Git - это распределённая система контроля версий, предназначенная для отслеживания изменений в файлах и организации совместной работы над проектами. Основная задача Git заключается в сохранении истории изменений, возможности возврата к предыдущим версиям файлов и управлении параллельной разработкой.

В отличие от централизованных систем контроля версий, Git не требует постоянного подключения к серверу. Каждый разработчик имеет локальную копию репозитория со всей историей изменений. Это повышает надёжность хранения данных и удобство работы.

### 2. Типы репозиториев Git

#### 2.1 Локальный репозиторий

Локальный репозиторий создаётся на компьютере пользователя с помощью команды git init.

Он содержит:

* рабочую директорию (файлы проекта),
* индекс (область подготовки изменений),
* историю коммитов.

Локальный репозиторий позволяет выполнять основные операции: добавление файлов, фиксацию изменений, просмотр истории и создание веток.

#### 2.2 Удалённый репозиторий

Удалённый репозиторий размещается на сервере. В данной работе используется платформа GitHub.

Удалённый репозиторий используется для:

* хранения проекта в облаке,
* совместной работы,
* резервного копирования,
* обмена изменениями между разработчиками.

Связь между локальным и удалённым репозиториями осуществляется через команды git push, git pull, git fetch.

### 3. Основные типы файловых состояний в Git

Git отслеживает состояние файлов в трёх основных областях:

**Рабочая директория (Working Directory)** - файлы проекта, с которыми работает пользователь.

**Индекс (Staging Area)** - область подготовки файлов к коммиту.

**Репозиторий (Repository)** - база данных зафиксированных изменений.

Файл может находиться в одном из состояний:

* неотслеживаемый (untracked),
* изменённый (modified),
* подготовленный к фиксации (staged),
* зафиксированный (committed).

### 4. Основные команды Git и их назначение

**Инициализация и настройка**

git init - создание нового локального репозитория.

git remote add origin - подключение удалённого репозитория.

ssh-keygen - генерация SSH-ключа для безопасной аутентификации.

**Работа с файлами**

git status - просмотр состояния файлов.

git add <имя\_файла> - добавление файла в индекс.

git commit -m "сообщение" - фиксация изменений в истории.

git log - просмотр истории коммитов.

git log --oneline - краткий вывод истории.

**Синхронизация с удалённым репозиторием**

git push - отправка локальных изменений на сервер.

git pull - получение изменений из удалённого репозитория.

git fetch - загрузка изменений без автоматического слияния.

### 5. Назначение SSH-ключа

SSH-ключ используется для безопасной аутентификации пользователя при подключении к удалённому репозиторию без ввода пароля.

При работе с GitHub SSH-ключ:

* подтверждает личность пользователя,
* обеспечивает защищённое соединение,
* упрощает процесс отправки изменений.

Ключ состоит из двух частей:

* приватный ключ (хранится на компьютере пользователя),
* публичный ключ (добавляется в настройки аккаунта GitHub).

### 6. Ветвление в Git

Ветвление - это механизм создания независимой линии разработки.

Основная ветка проекта обычно называется main (ранее - master).

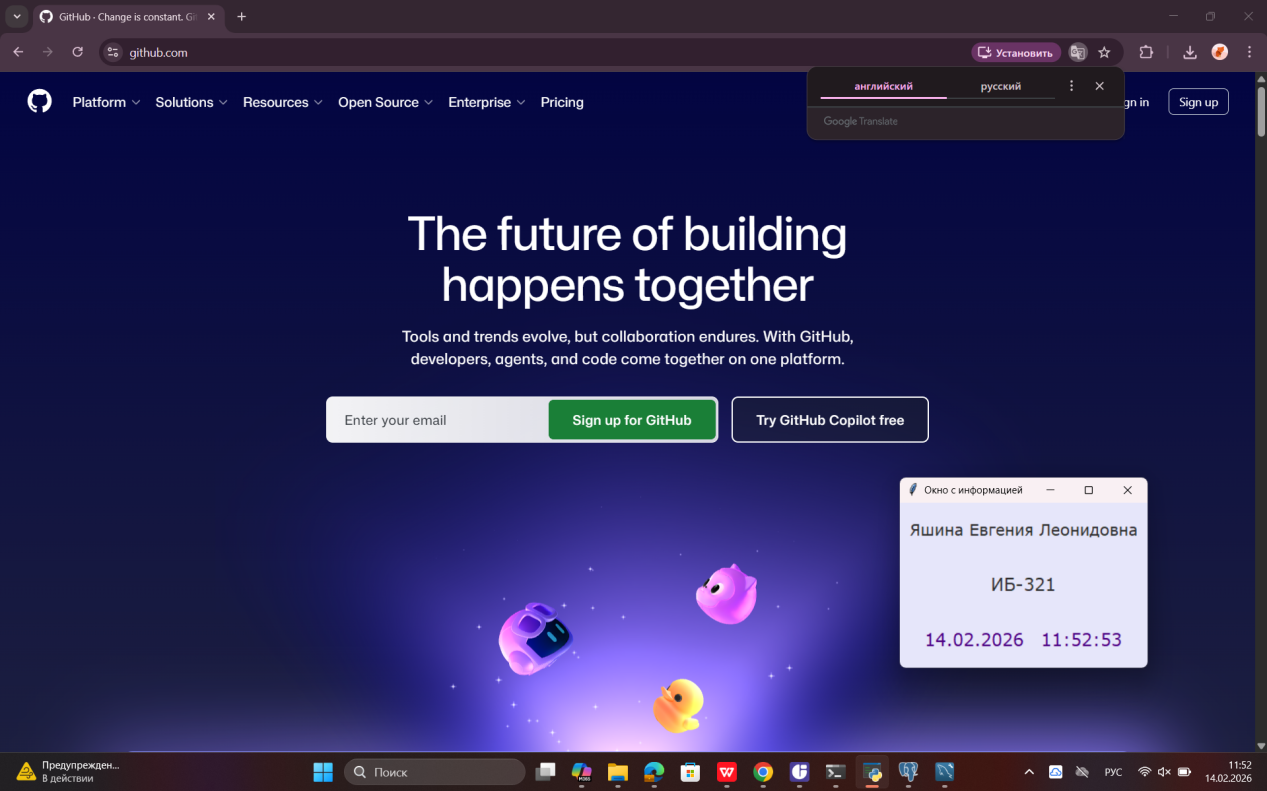
Основные команды:

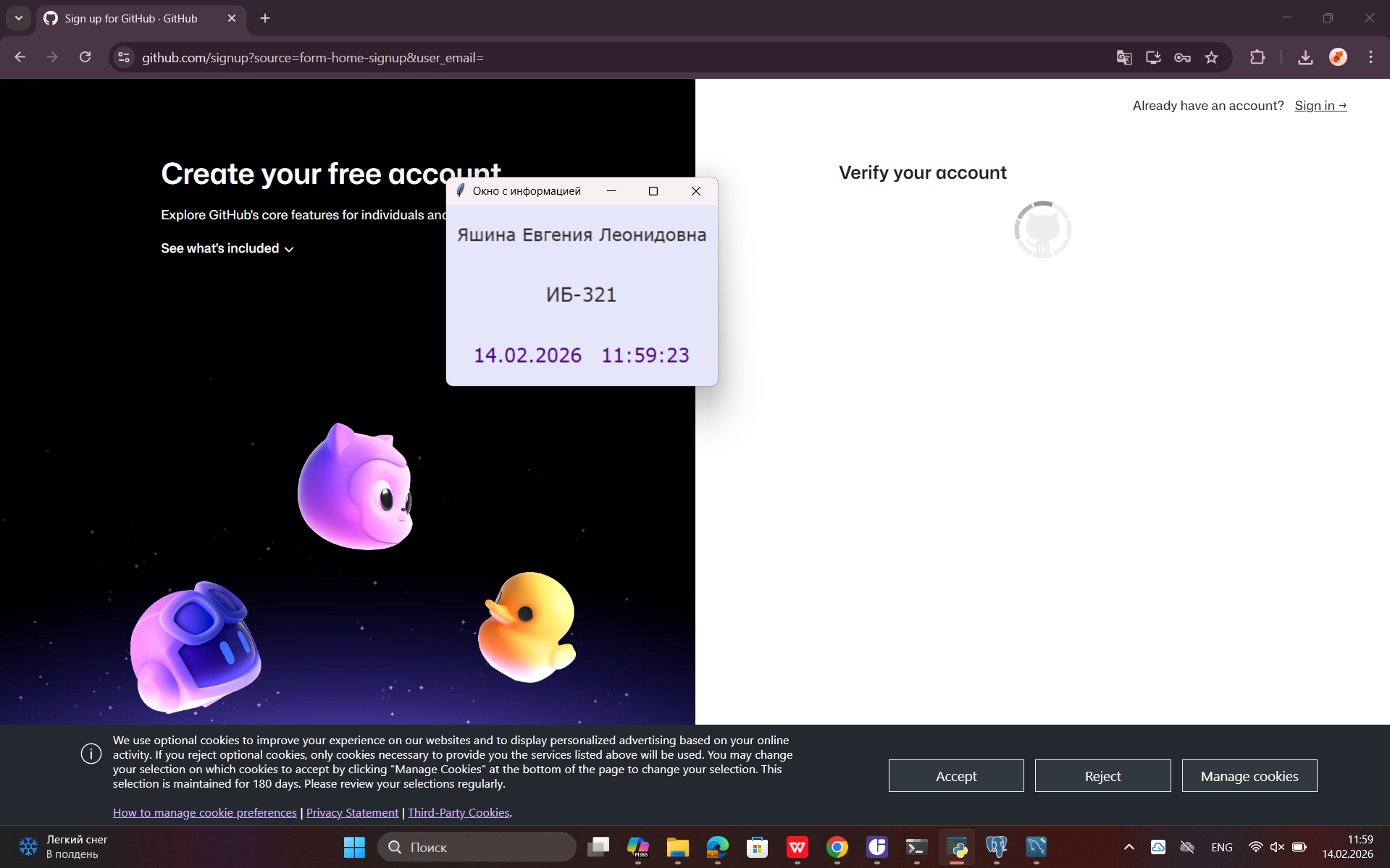
* git branch <имя> - создание ветки;
* git checkout <имя> - переключение на ветку;
* git merge - объединение веток.

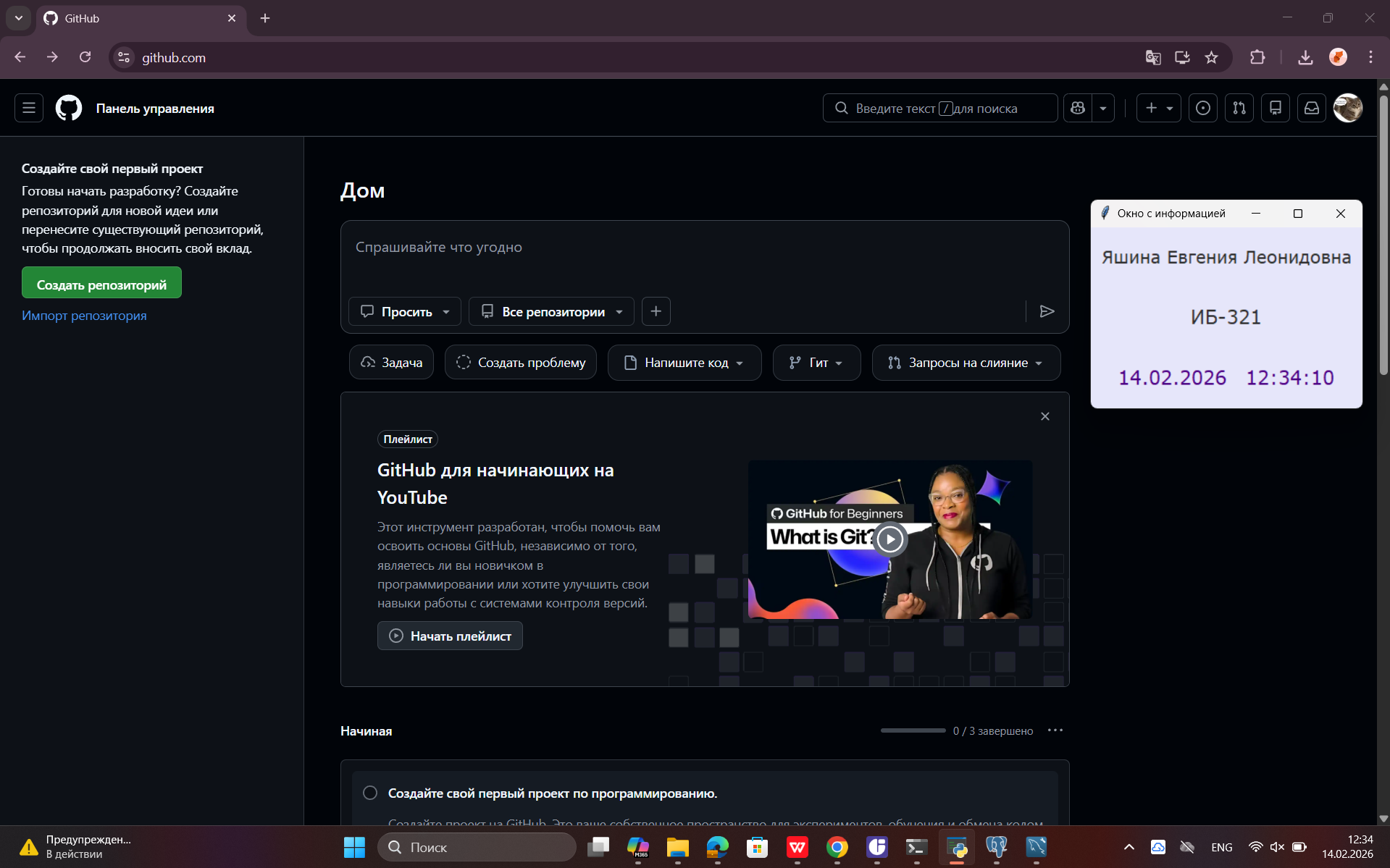
Ветвление позволяет разрабатывать новые функции без риска повлиять на основную версию проекта.

## Практическая часть:

### Регистрация на GitHub

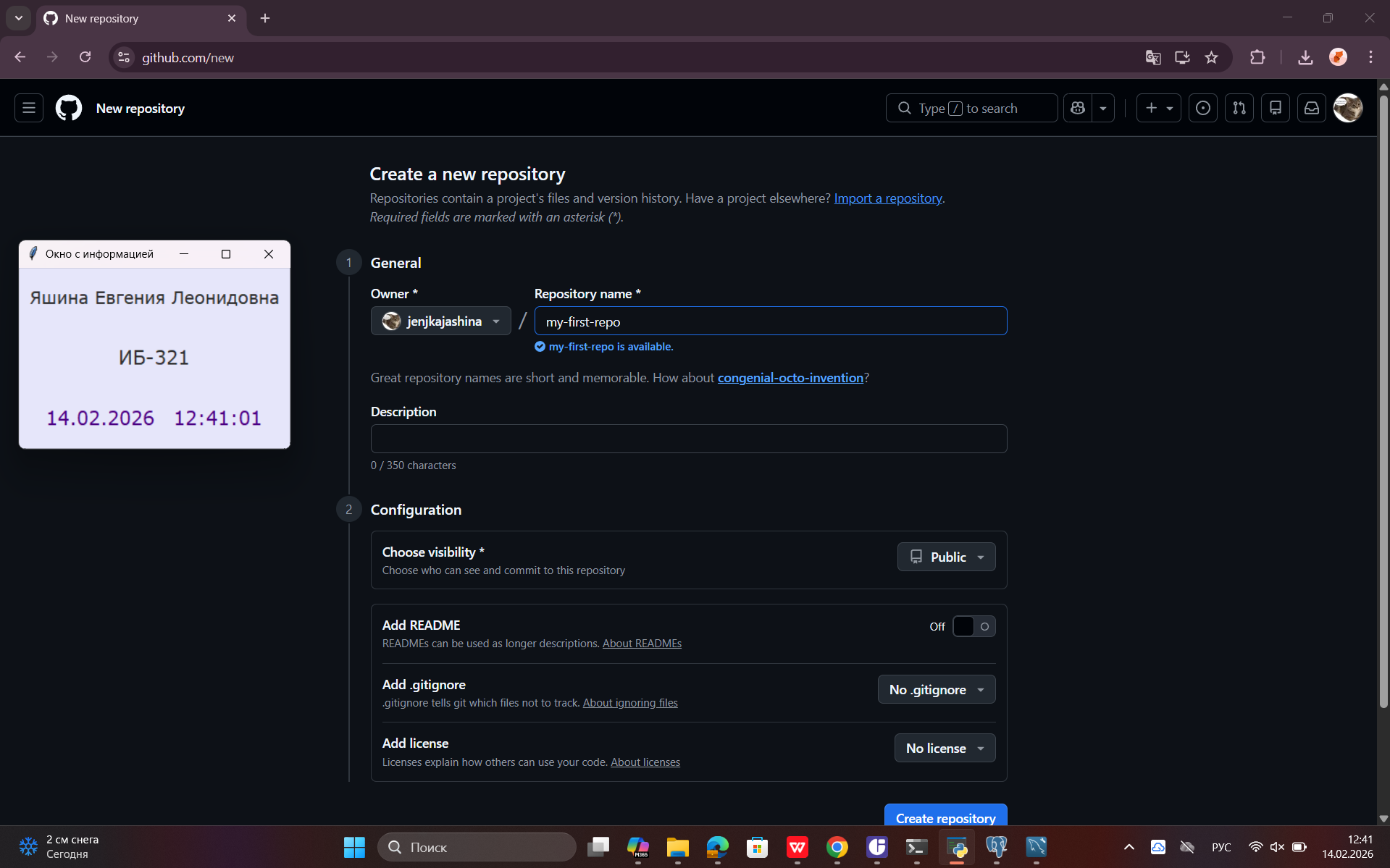


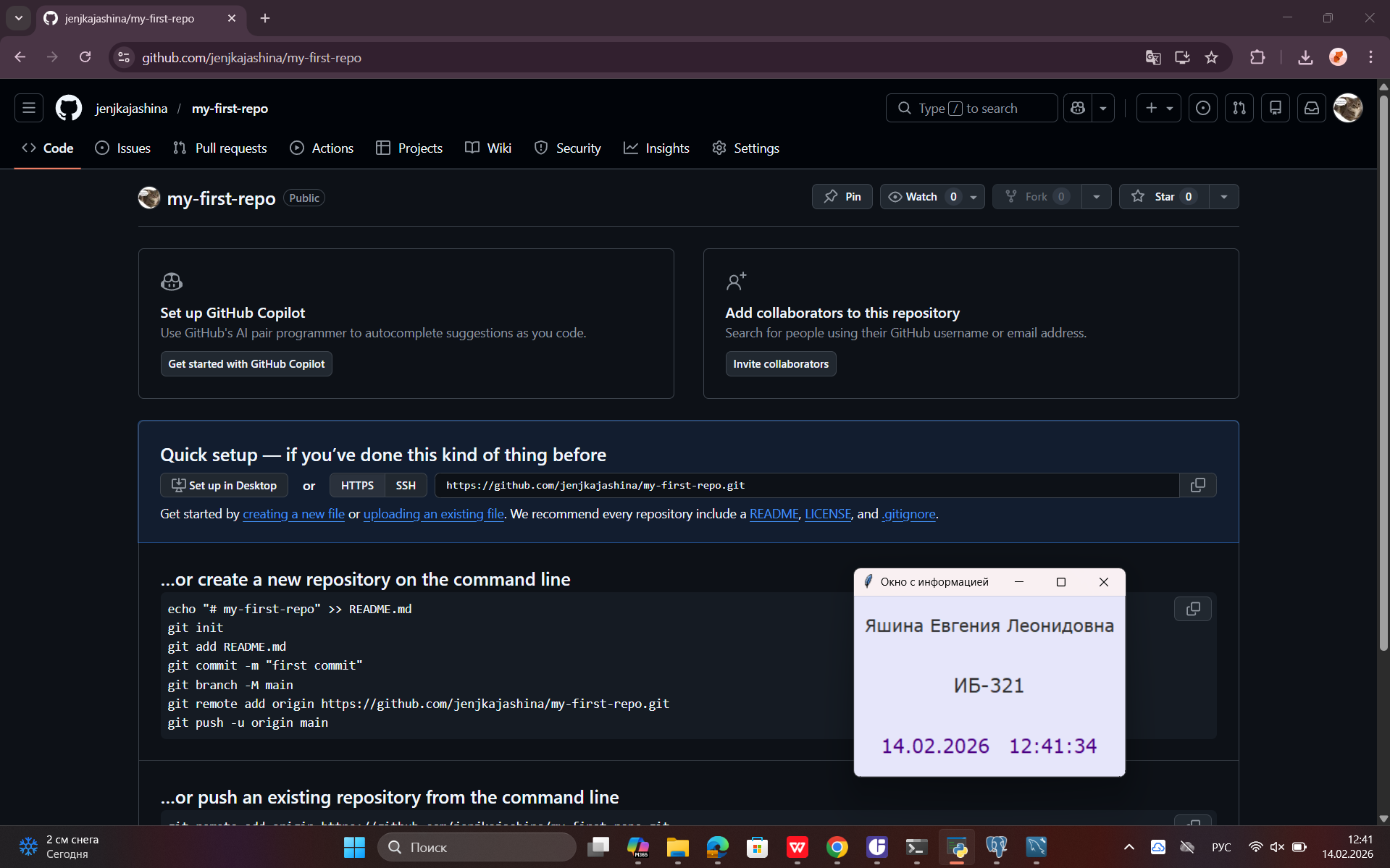




Успешно зарегистрировалась, завершила настройку профиля.

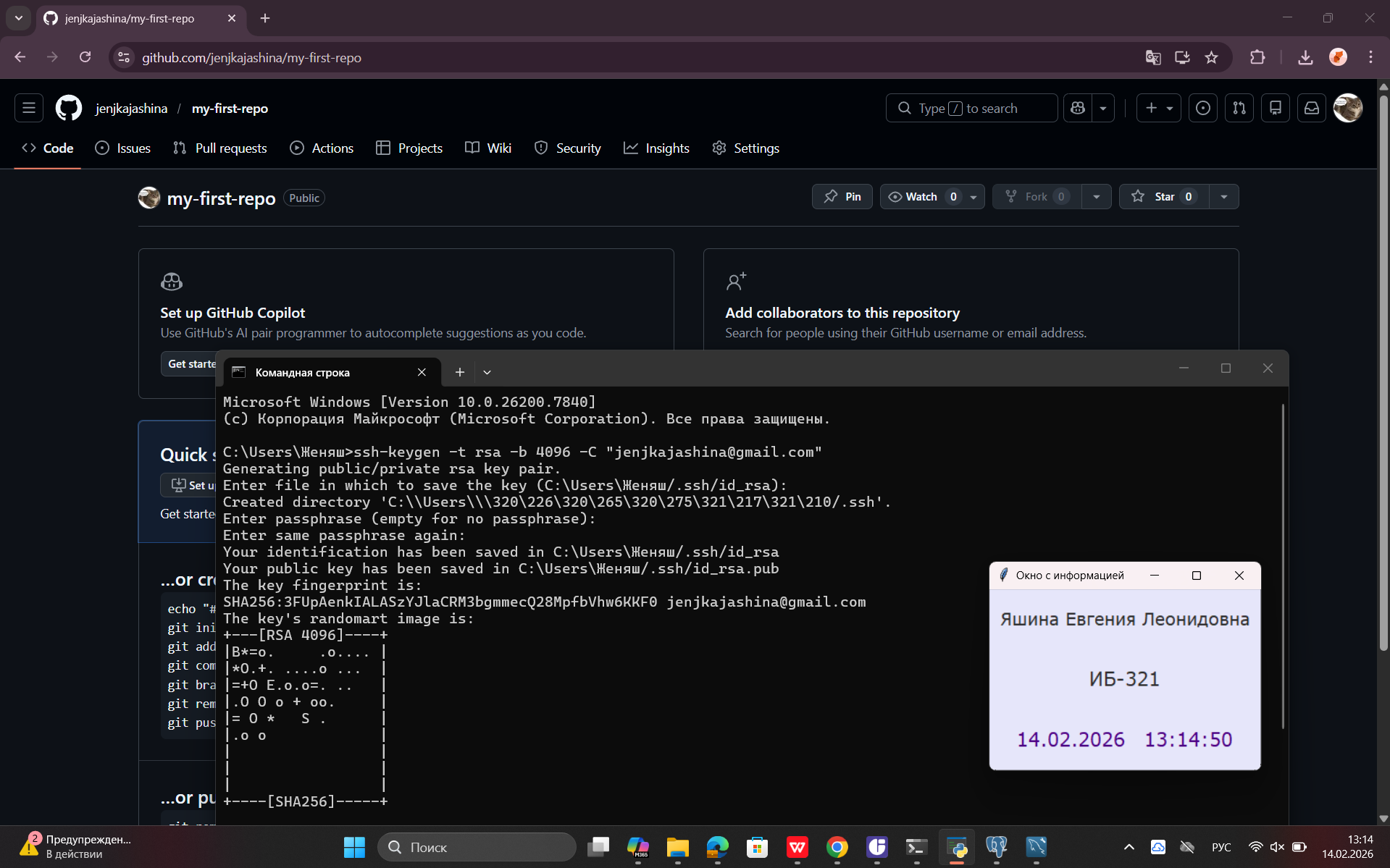
### 2. Создание репозитория



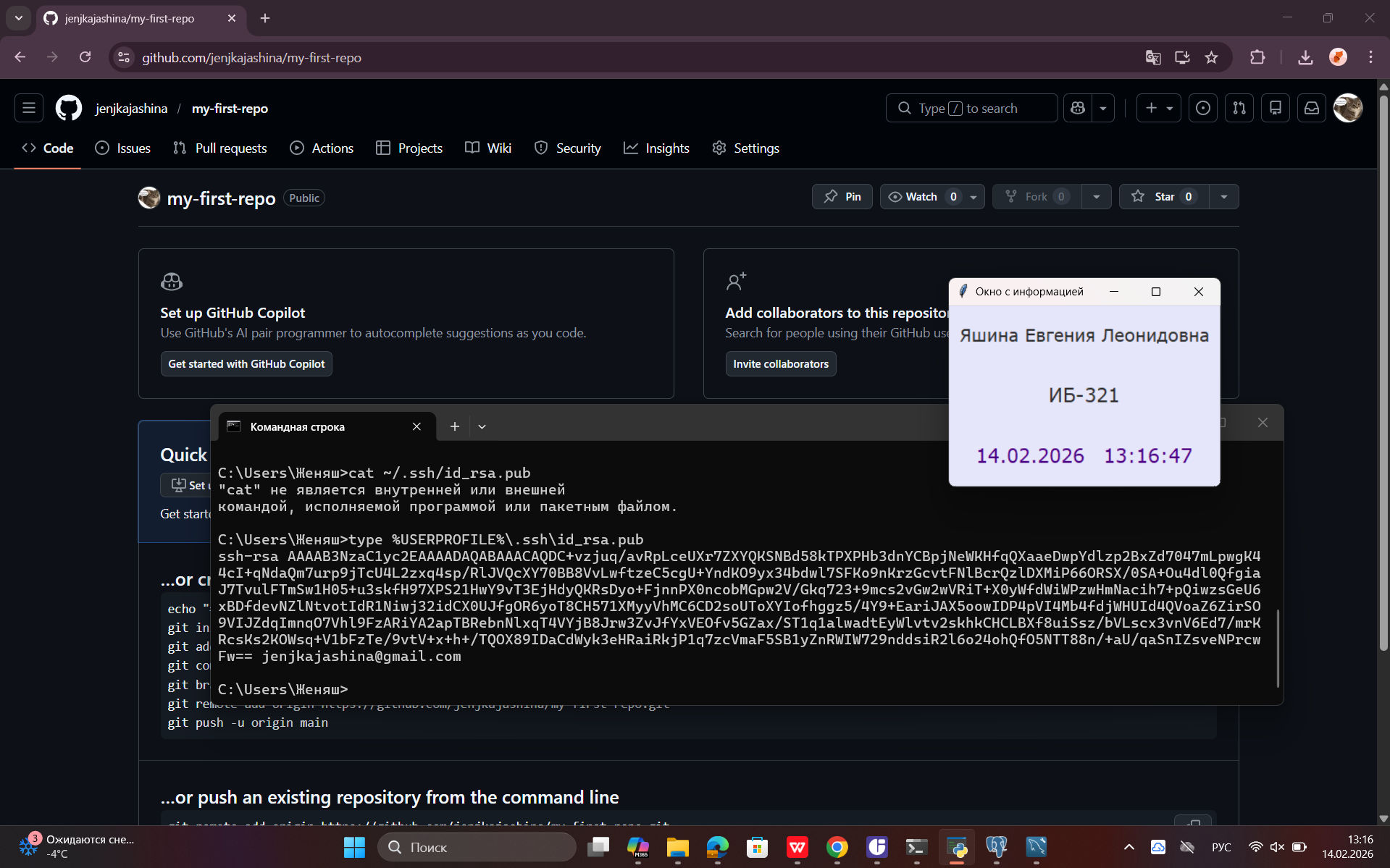


Создала свой первый репозиторий.

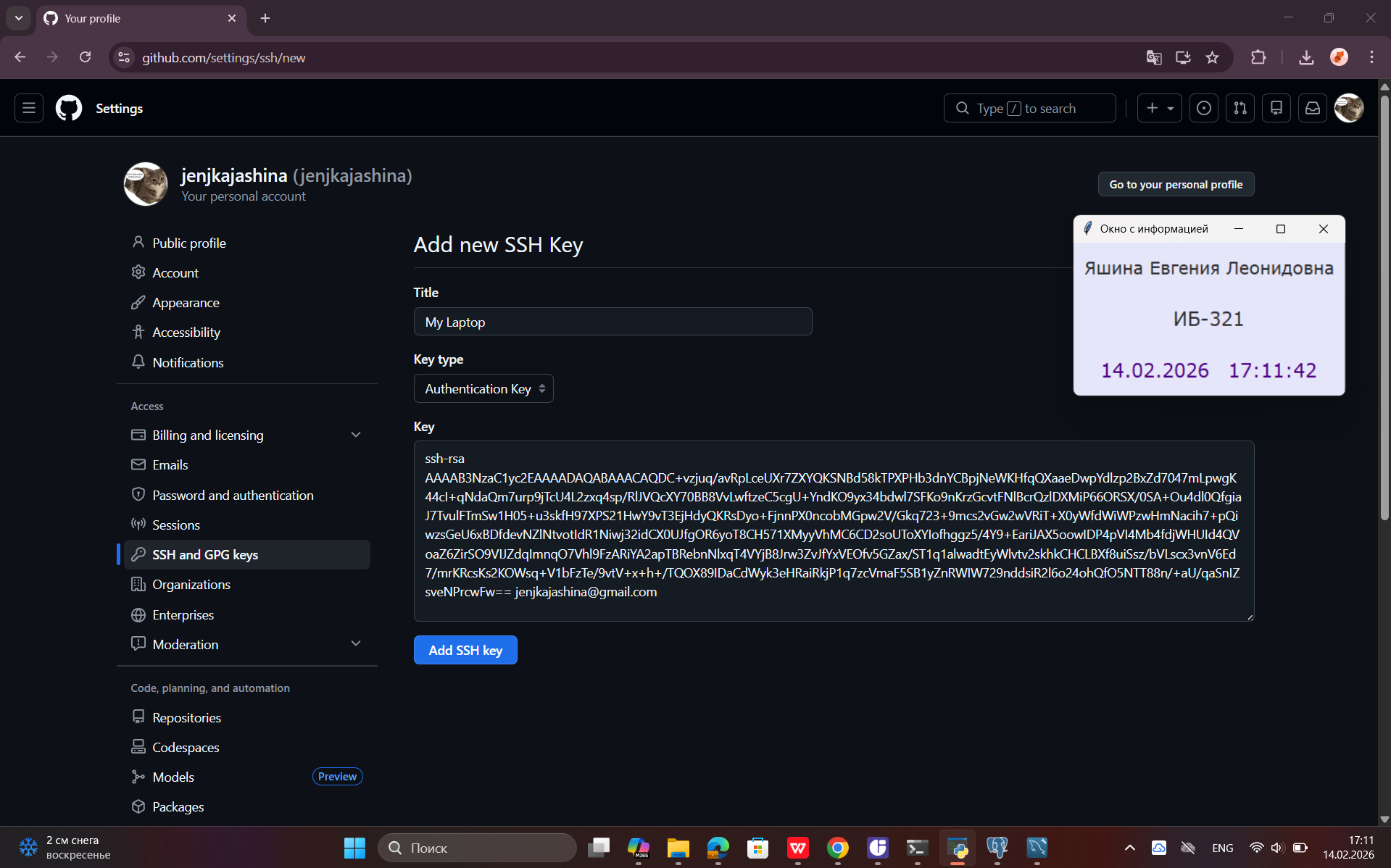
### 3. Настройка SSH‑ключа (для безопасной аутентификации).



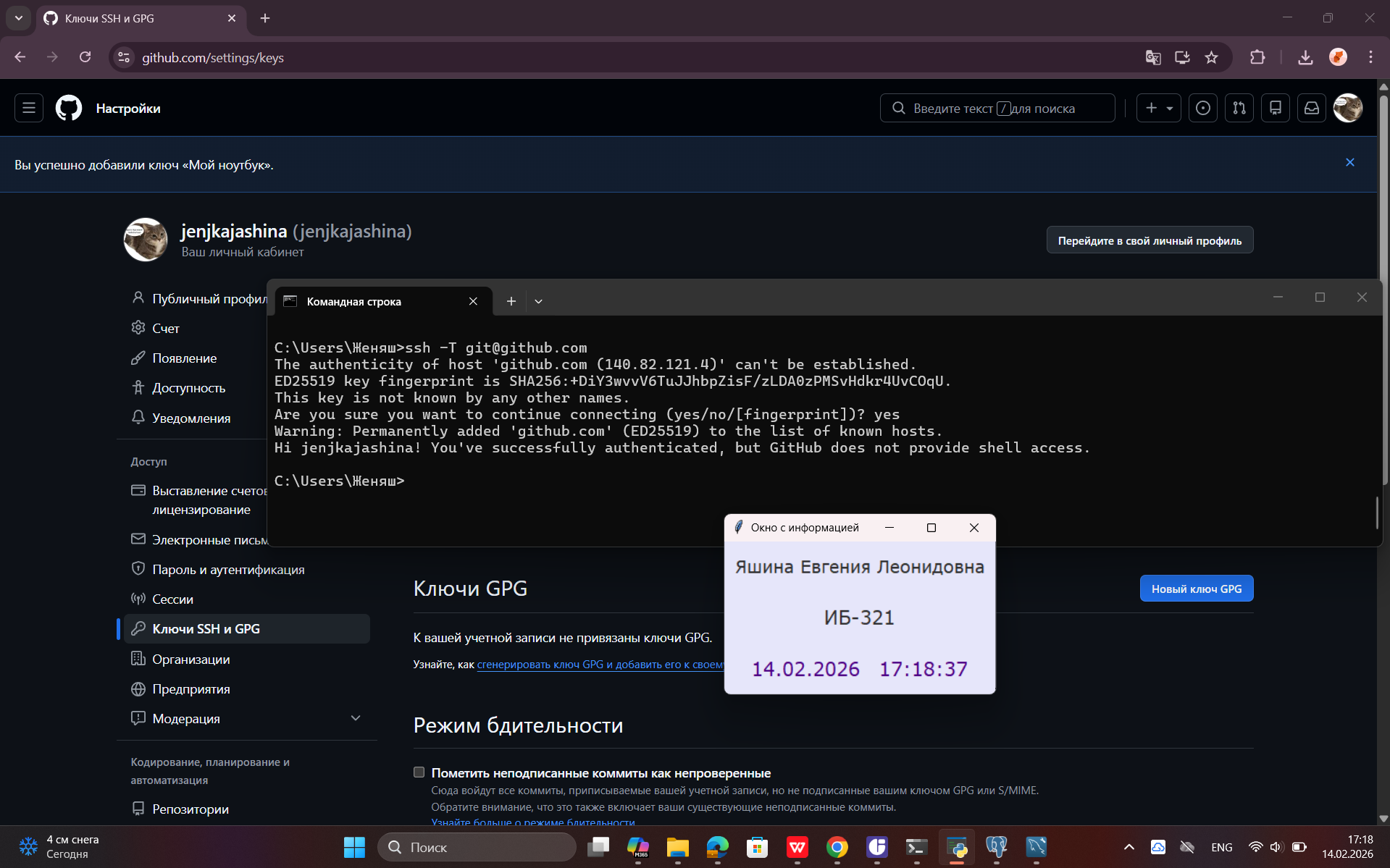
После ввода команды система предложила указать путь сохранения ключа. Было нажато Enter, чтобы использовать путь по умолчанию.  
 Также был пропущен ввод пароля (нажат Enter).



Отобразился текст, начинающийся с ssh-rsa. Этот текст был полностью скопирован.

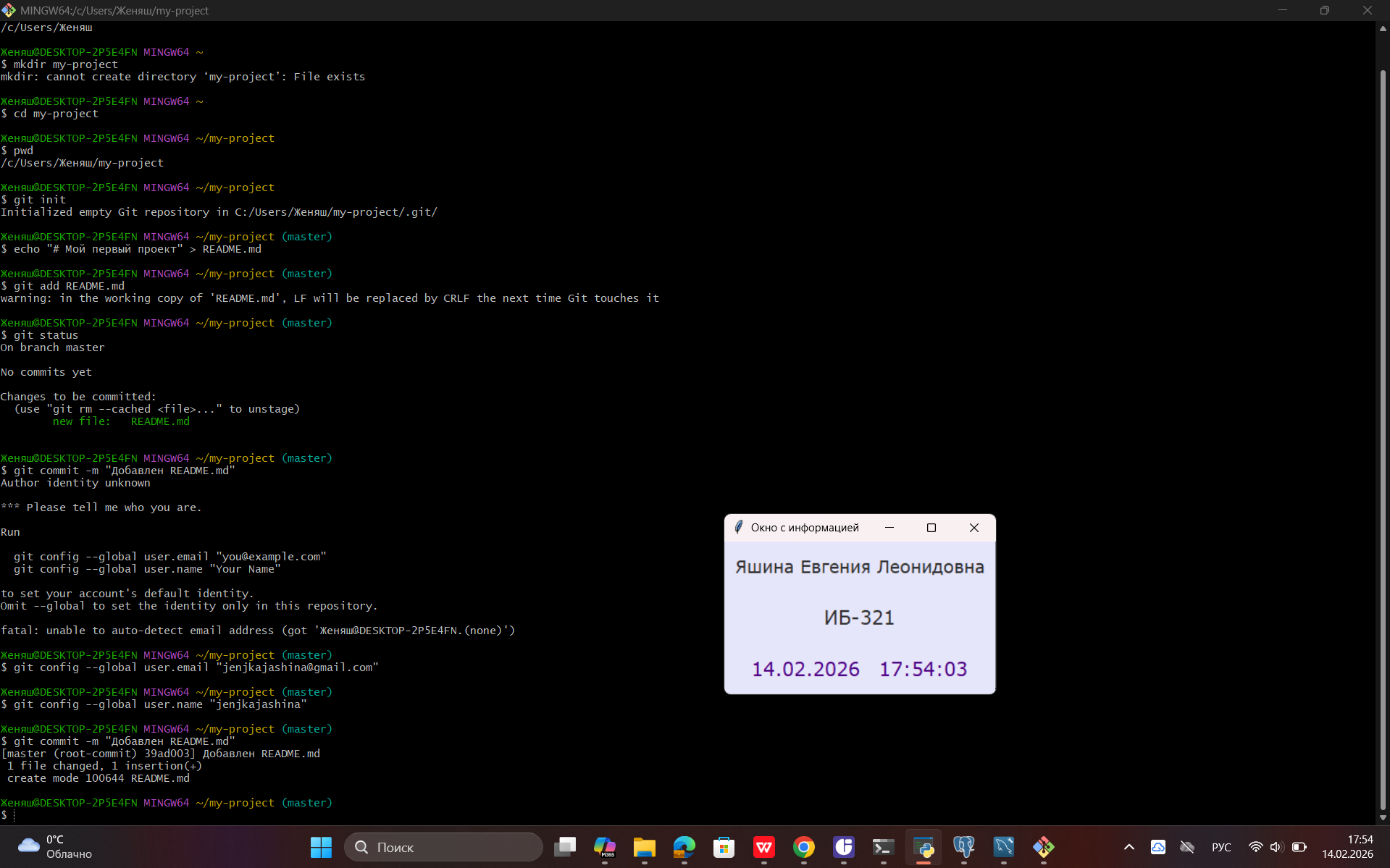


Добавление SSH-ключа в GitHub.



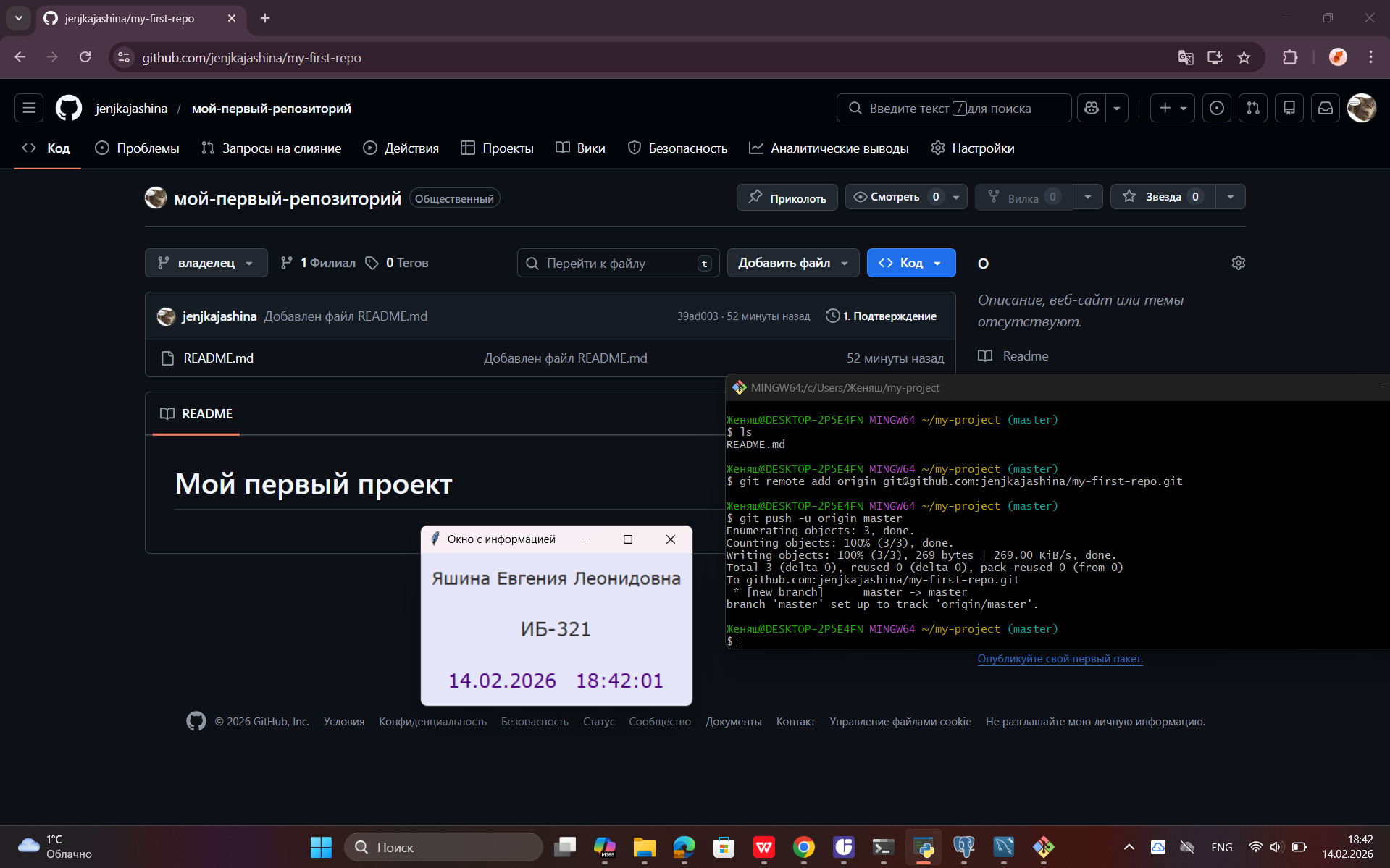
Компьютер успешно подключился к GitHub по SSH, проверка подключения это подтвердила.

### 4. Инициализация локального репозитория



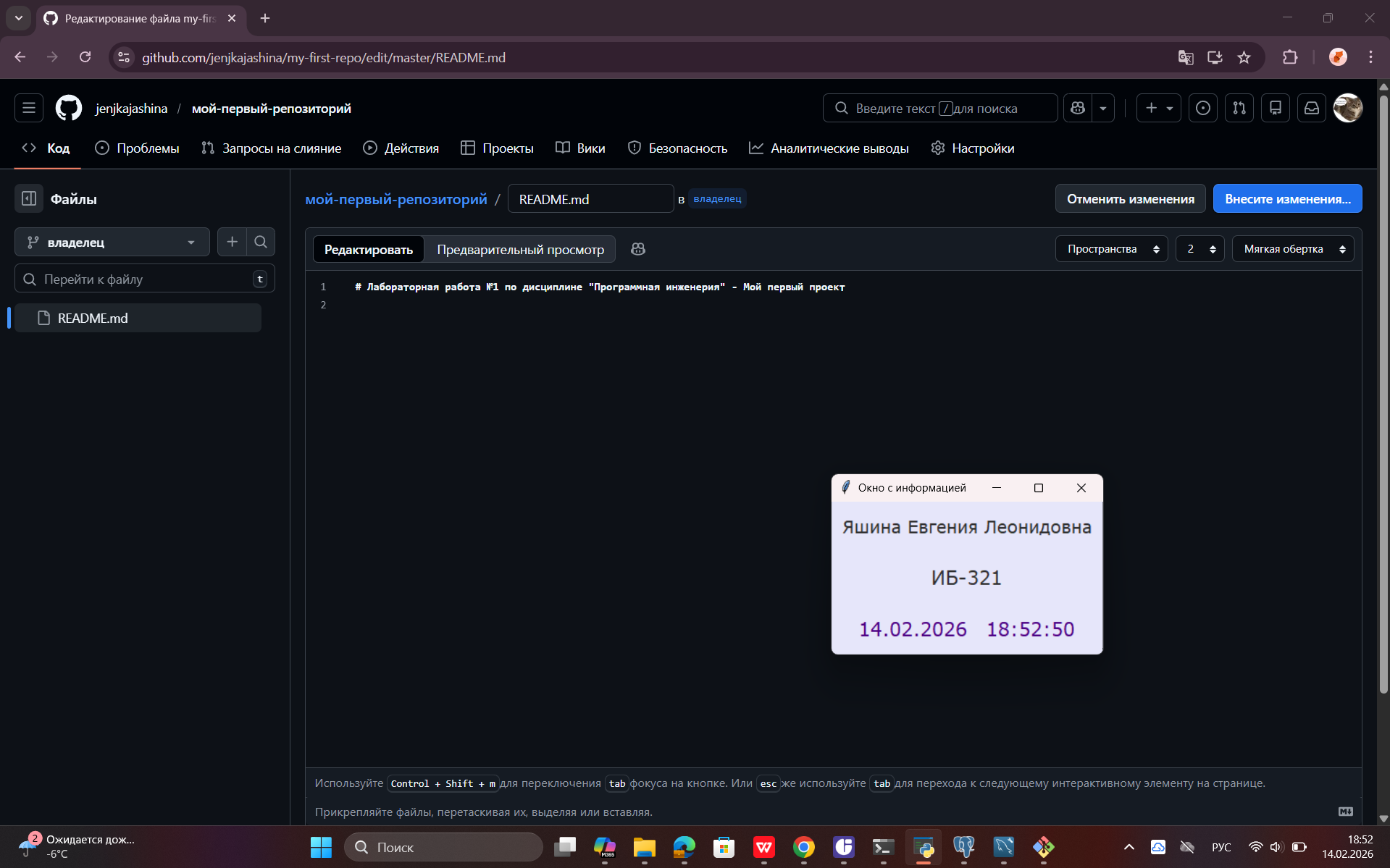
Мною была успешно проведена инициализация локального репозитория, в ходе которой я создала файл README.md с заголовком проекта и подготовила его к отслеживанию с помощью команды git add. После проверки состояния репозитория через git status, подтвердившей готовность файла, я настроила глобальные параметры пользователя (имя и почту) и зафиксировала изменения в истории проекта, выполнив первый коммит с сообщением "Добавлен README.md".

### 5. Подключение удалённого репозитория и отправка изменений.

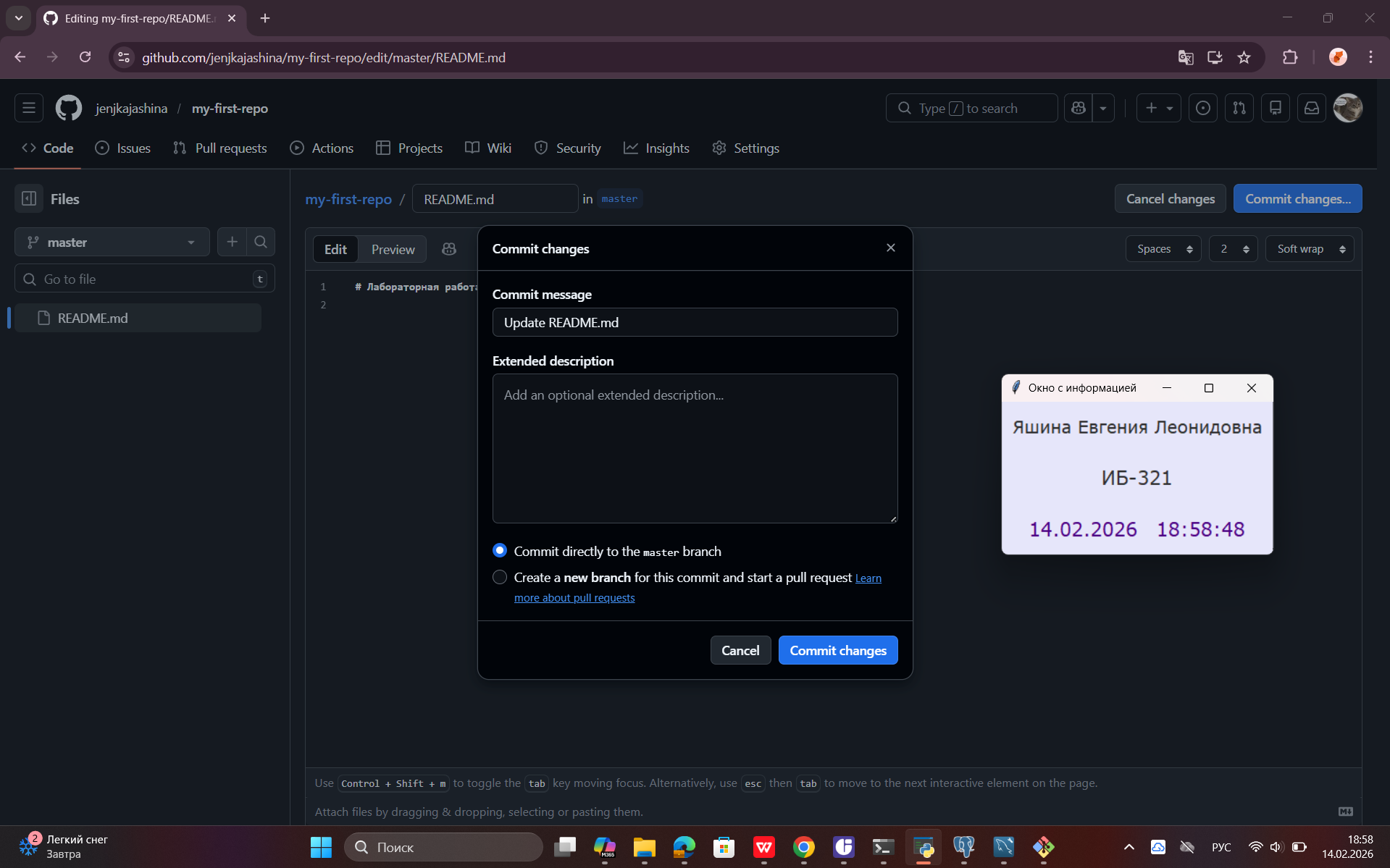


Сначала я ввела команду ls для просмотра файлов. Потом открыла cвой репозиторий на GitHub и скопировала адрес git@github.com:jenjkajashina/my-first-repo.git. Использовала ссылку в Git Bash. После осуществила отправку на GitHub c помощью команды git push -u origin master, обновила страницу и там появился файл README.

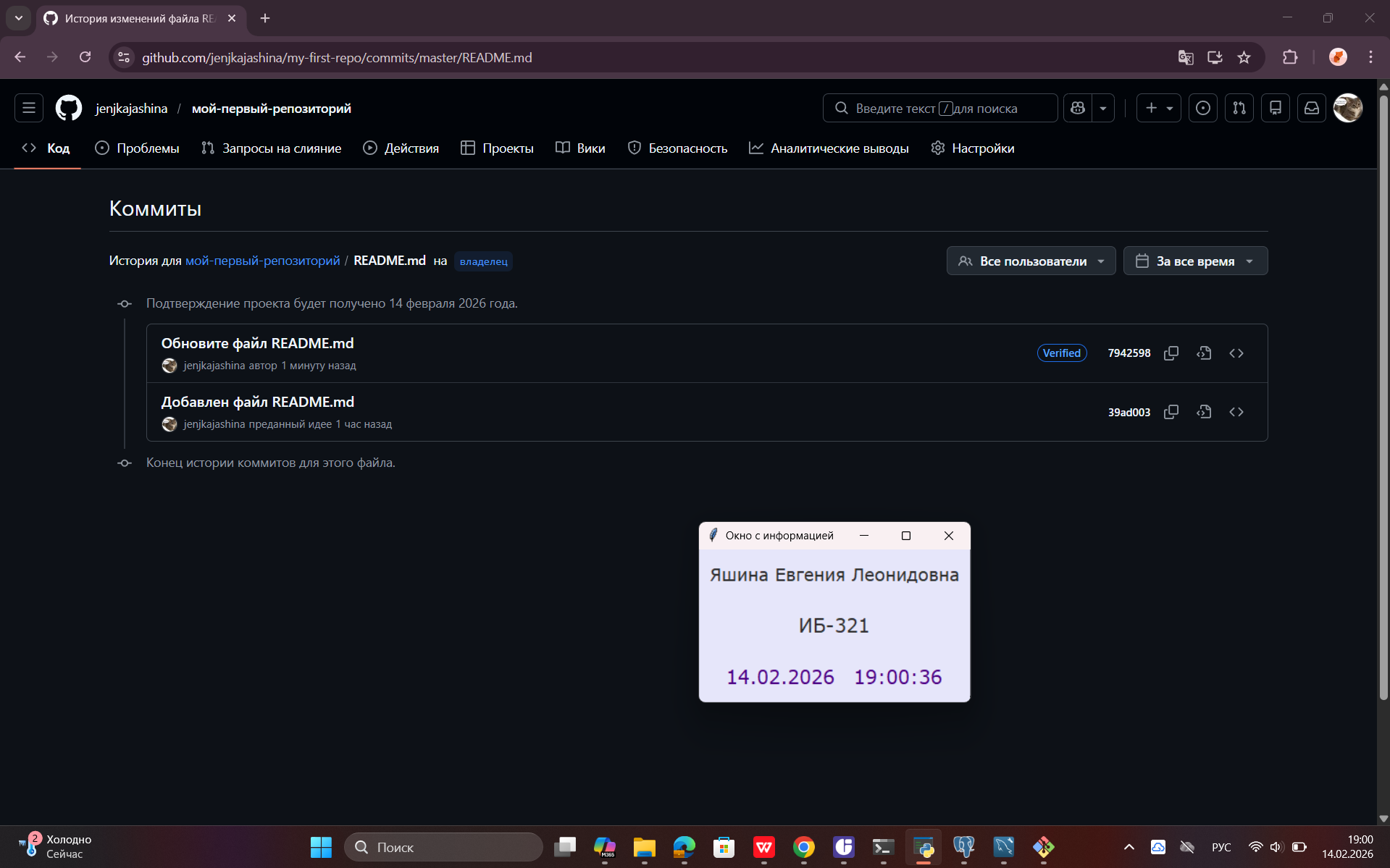
### 6. Работа с файлами на GitHub



Вношу изменения.

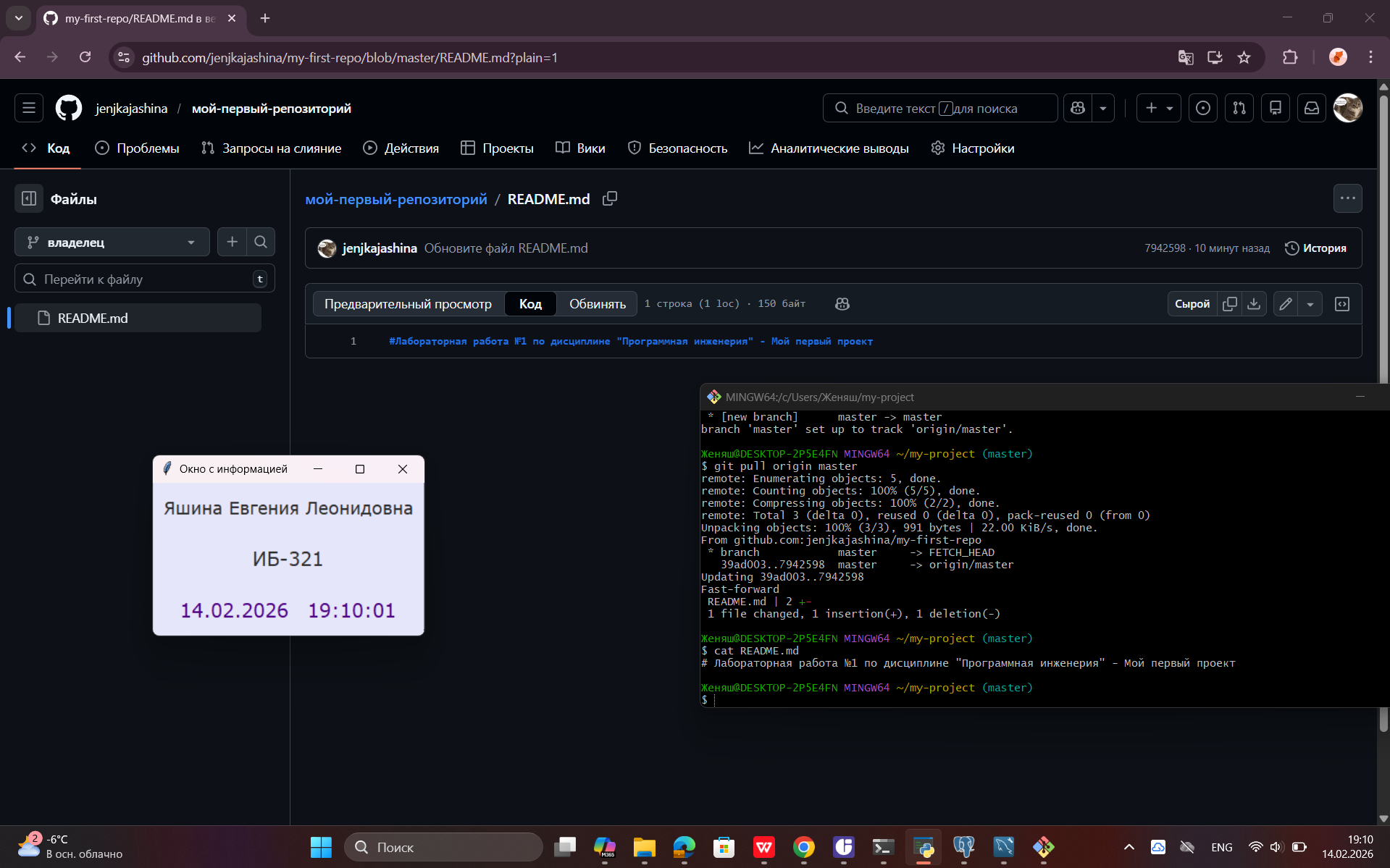


Сохраняю изменения.

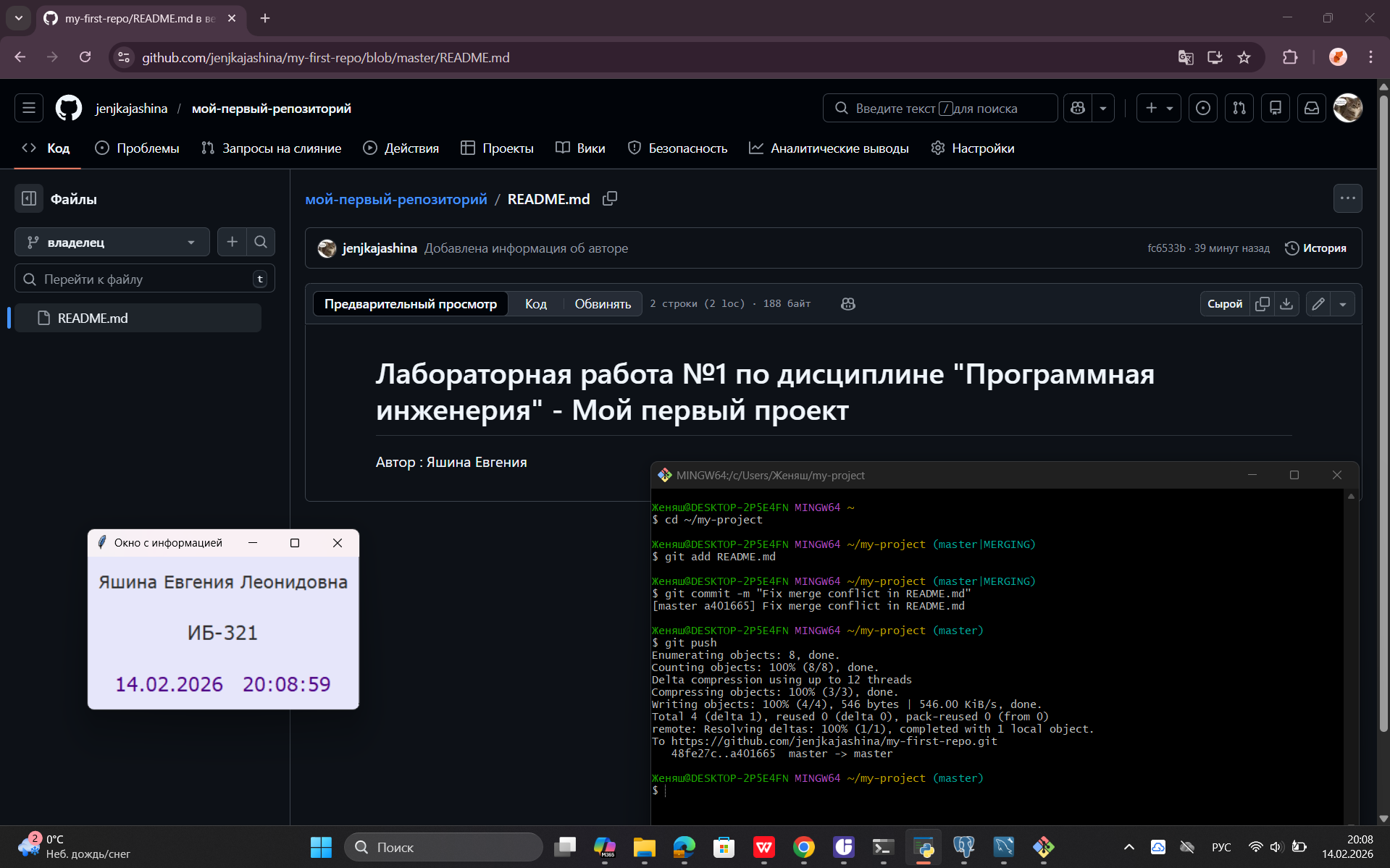


Изменение отображается успешно.

### 7. Синхронизация локального и удалённого репозиториев

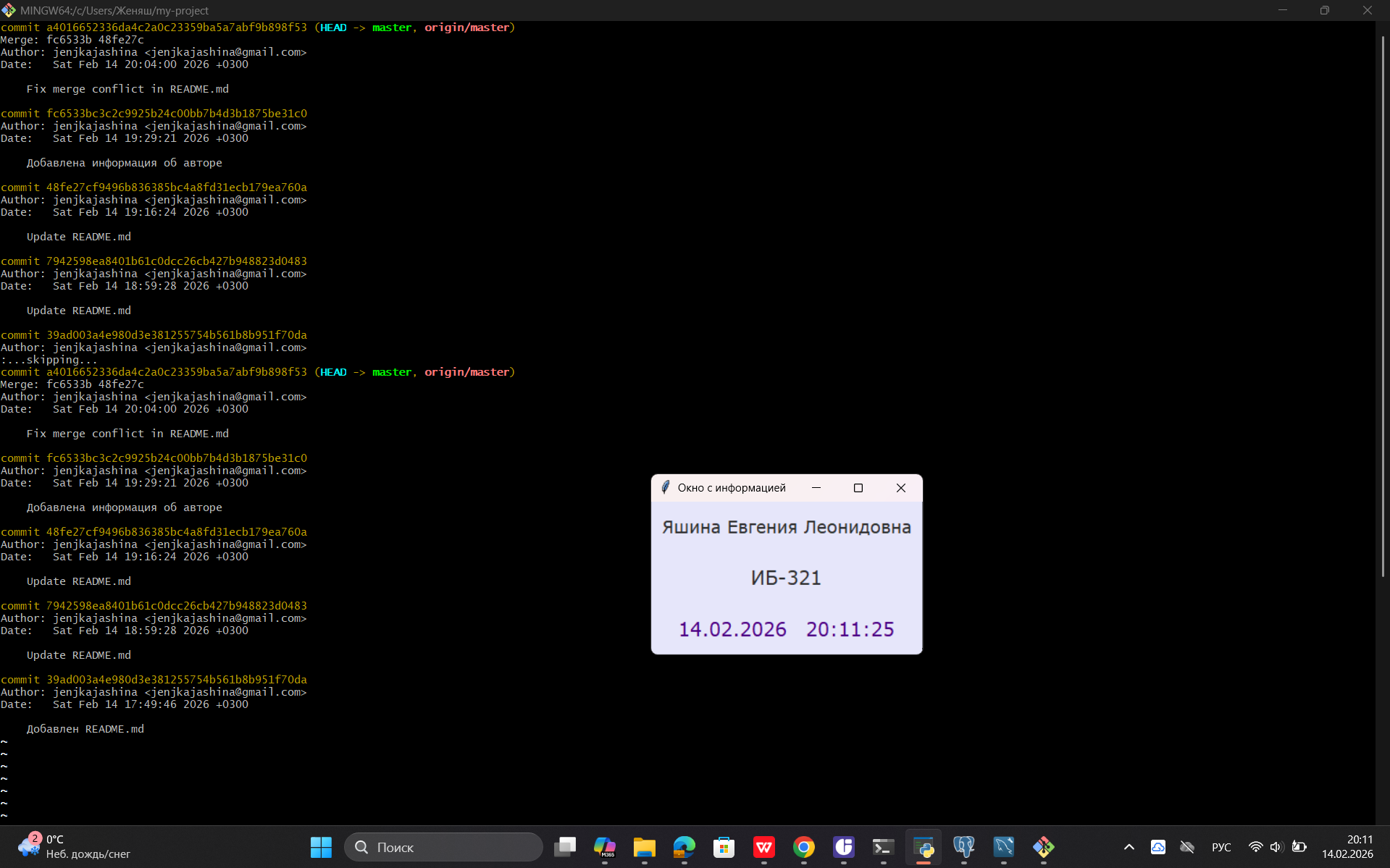


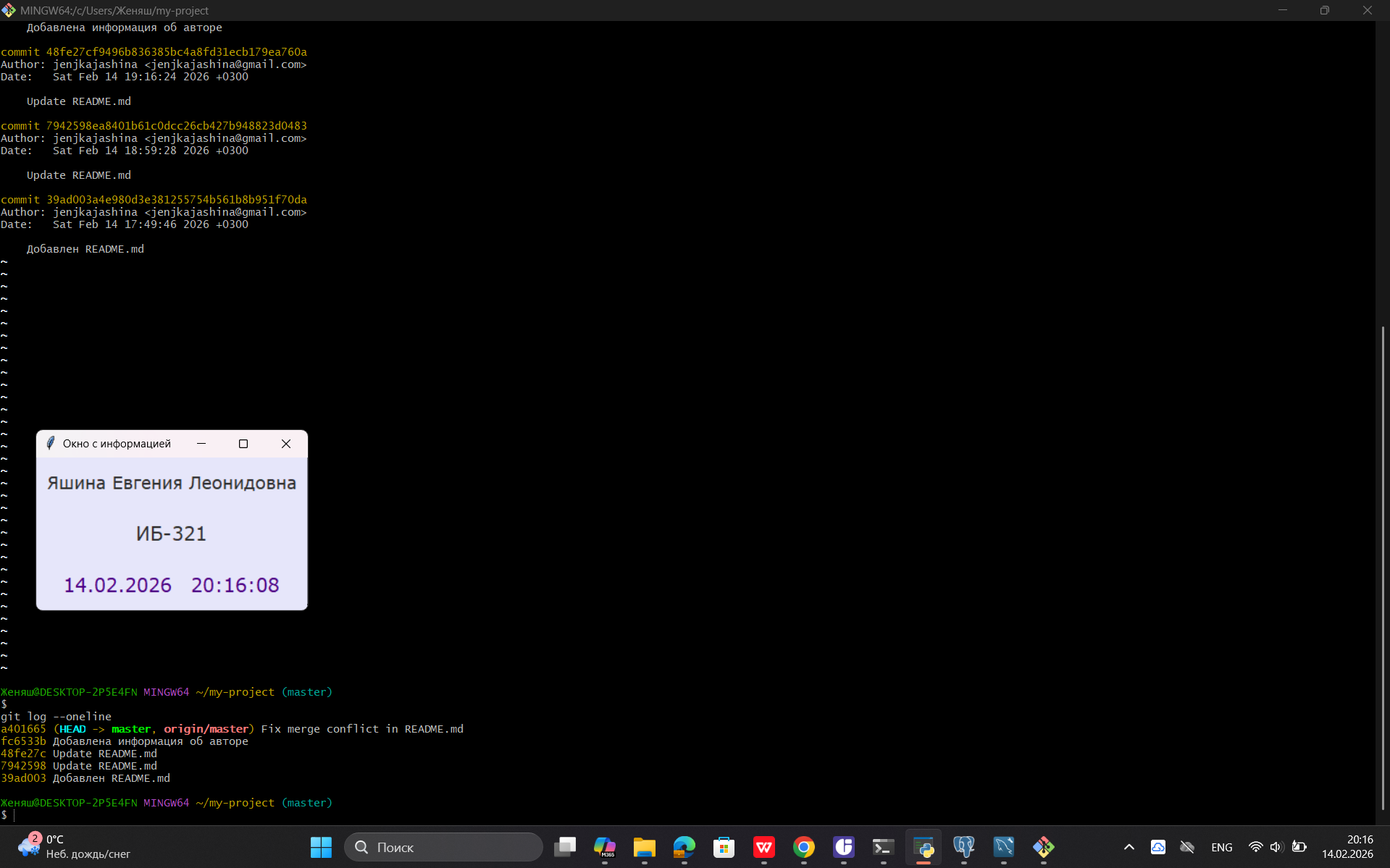
Загрузила изменения.



При выполнении седьмого этапа возник конфликт слияния, вызванный одновременным изменением файла README.md в локальном и удаленном репозиториях. Для решения этой проблемы я сначала синхронизировала локальную копию проекта с GitHub, загрузив актуальные данные, после чего вручную отредактировала файл README.md, удалив технические маркеры Git и сформировав финальный текст. Затем я подготовила исправленный файл к сохранению с помощью команды git add и зафиксировала результат специальным коммитом слияния. В завершение я выполнила команду git push, которая успешно перенесла итоговую версию проекта с учетом всех правок обратно на сервер.

### 8. Изучение истории изменений.





Для финального анализа проекта я использовала команды просмотра истории изменений. Команда git log позволила ознакомиться с полными данными о каждом коммите, включая автора и точное время правок. Далее, применив команду git log --oneline, я получила компактный вид истории, где каждый этап работы представлен одной строкой, что подтверждает корректную структуру всех выполненных мной сохранений и успешное завершение процесса слияния

## Ответы на контрольные вопросы:

1. ***Что такое репозиторий?***

Репозиторий представляет собой хранилище данных проекта, включающее все файлы, историю изменений и метаданные. Он позволяет отслеживать изменения, управлять версиями и обеспечивать совместную работу над проектом. В контексте Git репозиторий может быть локальным на компьютере разработчика или удалённым на платформе, такой как GitHub.

1. ***Для чего нужен SSH-ключ при работе с GitHub?***

SSH-ключ используется для безопасной аутентификации между локальным компьютером и удалённым репозиторием на GitHub. Он позволяет устанавливать защищённое соединение без необходимости вводить пароль при каждом взаимодействии с удалённым репозиторием.

1. ***В чём разница между git commit и git push?***

Команда git commit фиксирует изменения в локальном репозитории, сохраняя их в истории проекта. Команда git push передаёт эти локальные коммиты на удалённый репозиторий, делая их доступными для других участников проекта.

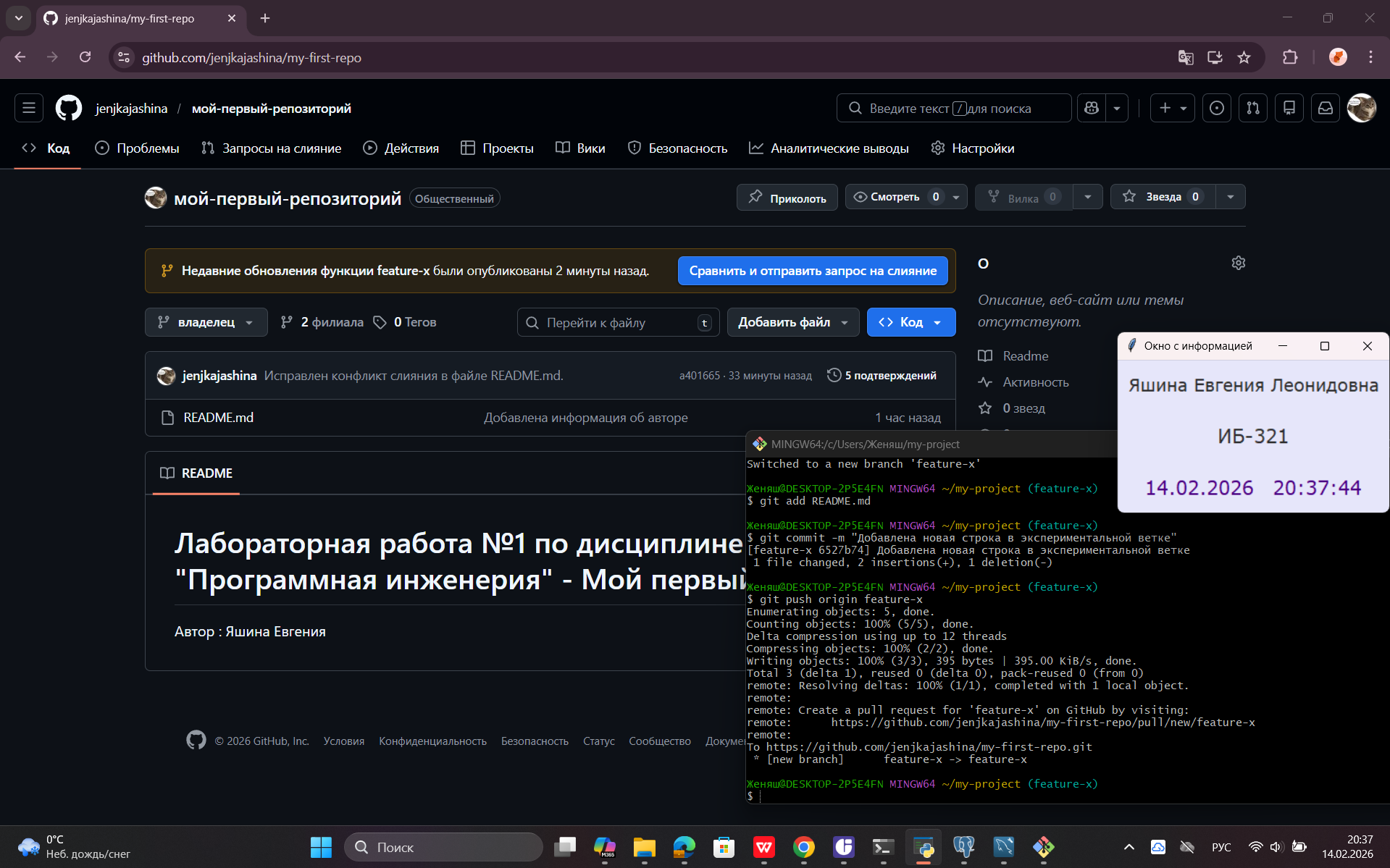
1. ***Как просмотреть список изменённых файлов перед коммитом?***

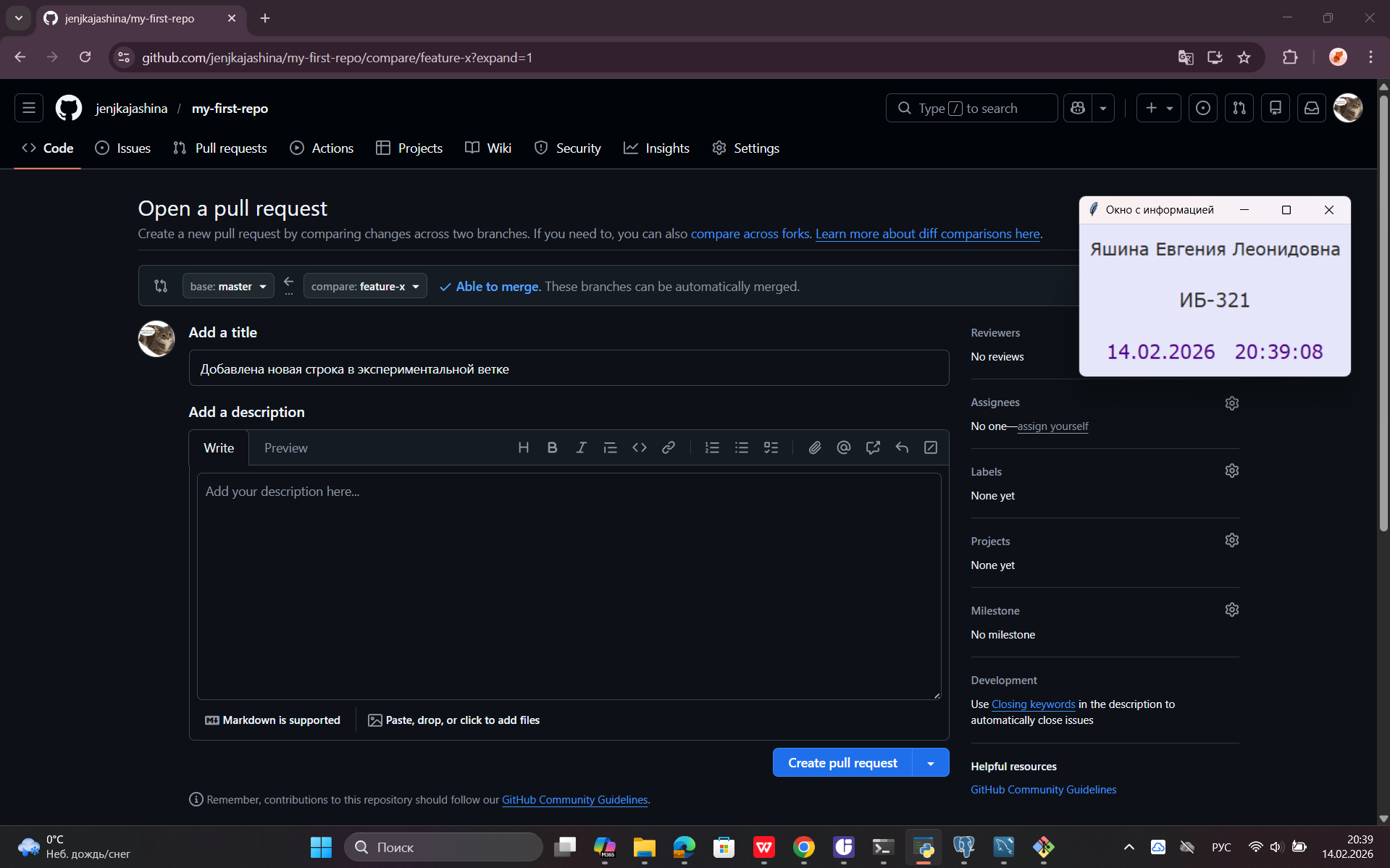
Список изменённых файлов можно просмотреть с помощью команды git status. Она показывает файлы, которые были изменены, добавлены или удалены, а также файлы, подготовленные к коммиту и находящиеся в стадии индексации.

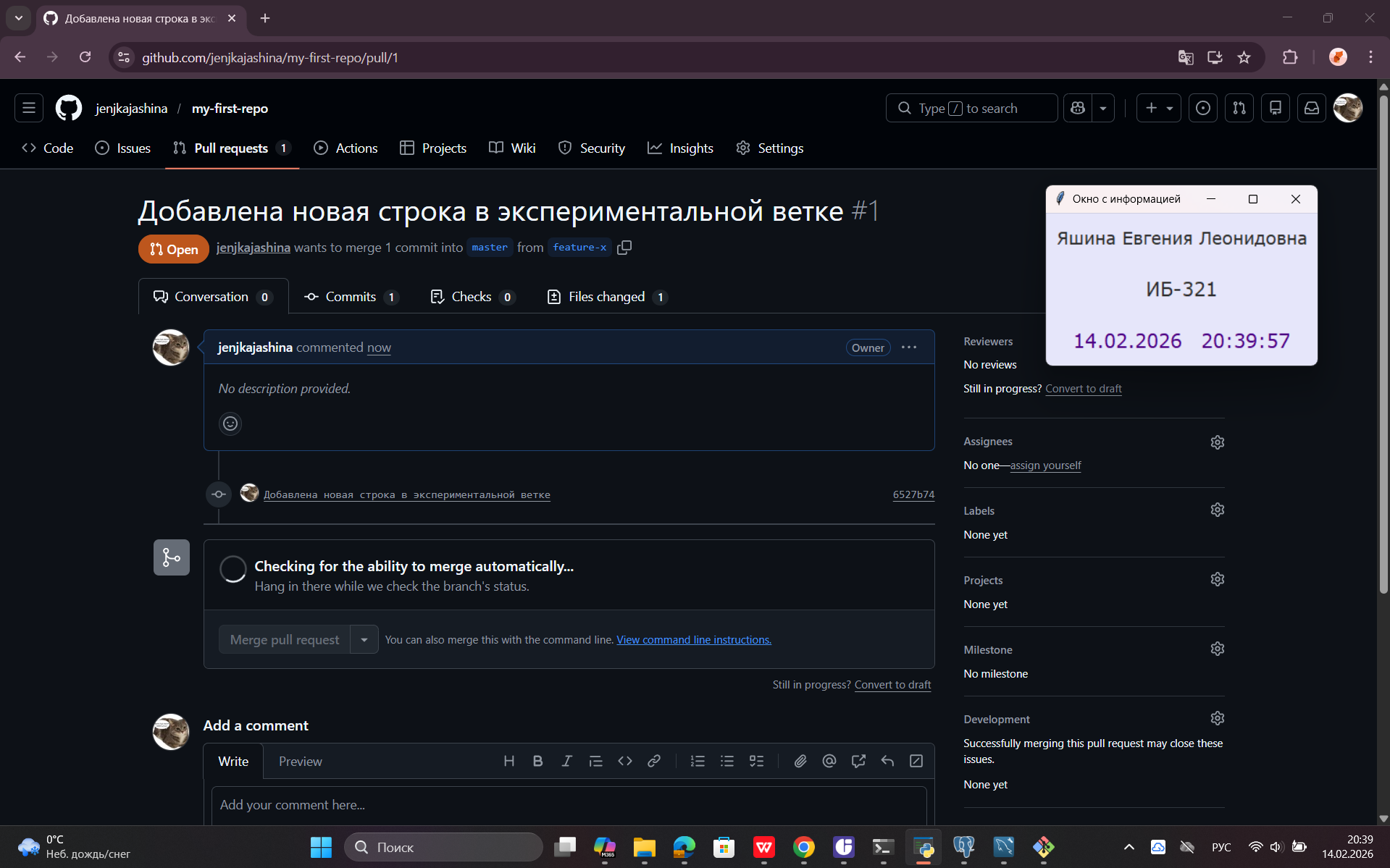
1. ***Что означает команда git pull?***

Команда git pull выполняет обновление локального репозитория, загружая изменения из удалённого репозитория и объединяя их с текущей веткой. Она сочетает в себе действия git fetch, который получает данные, и git merge, который интегрирует их в локальную копию.

## Дополнительное задание:







Для выполнения дополнительного задания я создала новую ветку feature-x и переключилась на неё, чтобы мои правки не влияли на основной код в ветке master. В этой ветке я добавила новую строку в файл README.md, сохранила изменения и отправила их на GitHub. В завершение я зашла на сайт GitHub и создала Pull Request - это специальный запрос, который показывает мои изменения и предлагает объединить их с основной версией проекта.

# Вывод:

Я освоила работу с терминалом Git и платформой GitHub, включая настройку SSH ключей, управление репозиториями и синхронизацию данных через команды commit, pull и push. Я научилась разрешать конфликты слияния, работать с ветками и создавать Pull Request, что позволило мне на практике пройти полный цикл управления версиями проекта.