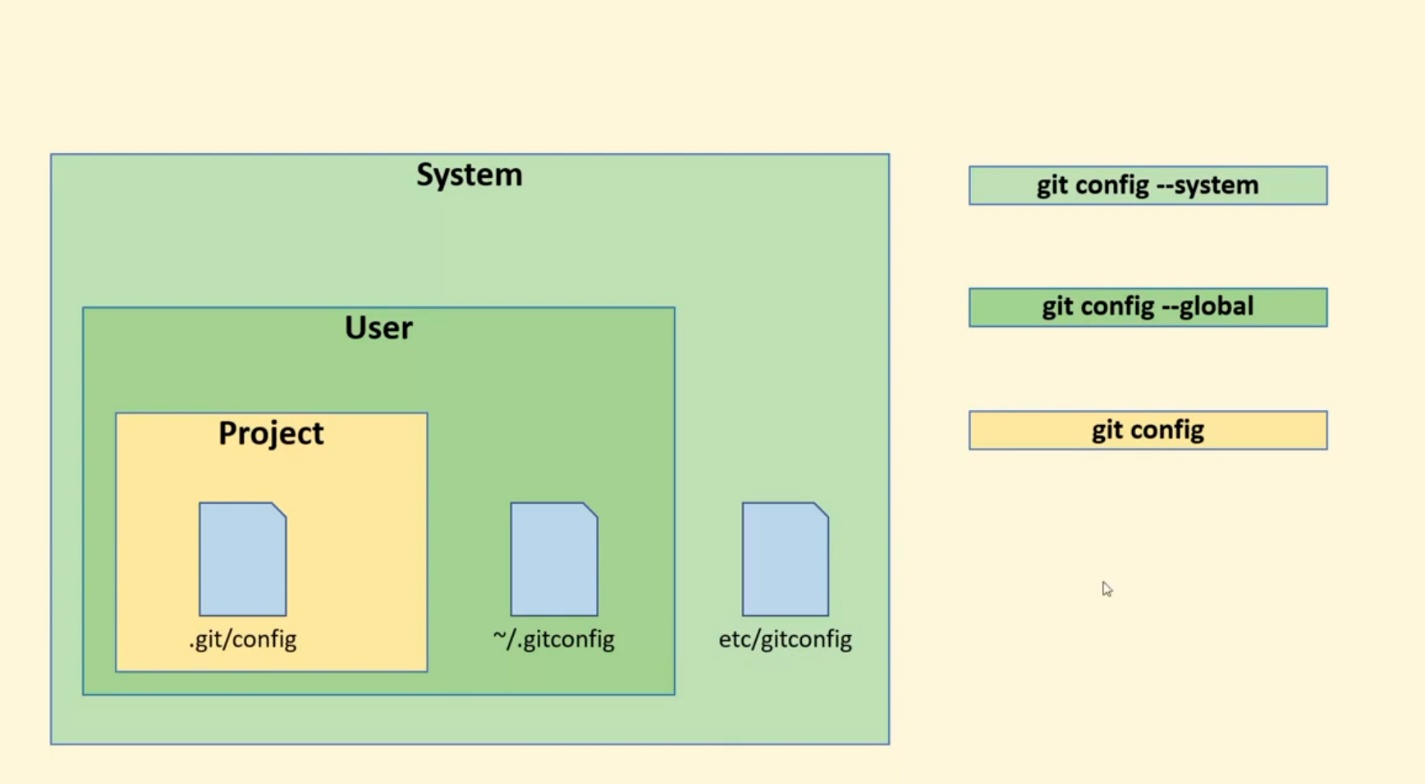
**Структура настройки GIT**

****

**git config --global user.email**

**git config --global user.name**

**проверка настроек - команда**

**git config --global --list**

**подсказка по командам:**

**git help -g**

**четыре дополнительные настройки:**

1. **git config --global core.autocrlf true - Если у вас Windows, то установите значение true — при извлечении кода LF окончания строк будут преобразовываться в CRLF (при установке Git есть галочка).**
2. **git config --global core.safecrlf warn - Если core.safecrlf установлен на "true" или "warm", Git проверяет, если преобразование является обратимым для текущей настройки core.autocrlf.**

**core.safecrlf true - отвержение необратимого преобразования lf<->crlf. Полезно, когда специфические бинарники похожие на текстовые файлы.**

**core.safecrlf warn - печать только предупреждение, но принимает необратимый переход.**

1. **git config --global core.quotepath off -Русификация git в консоли**
2. **git config --global init.defaultBranch main - общепринятое название основной ветки в репозитории в Git — main**

**Чтобы выйти из справки, нажмите Escape → наберите :wq → нажмите Enter, при этом символ «:» нажимать не в цифровом ряду, а в буквенном.**

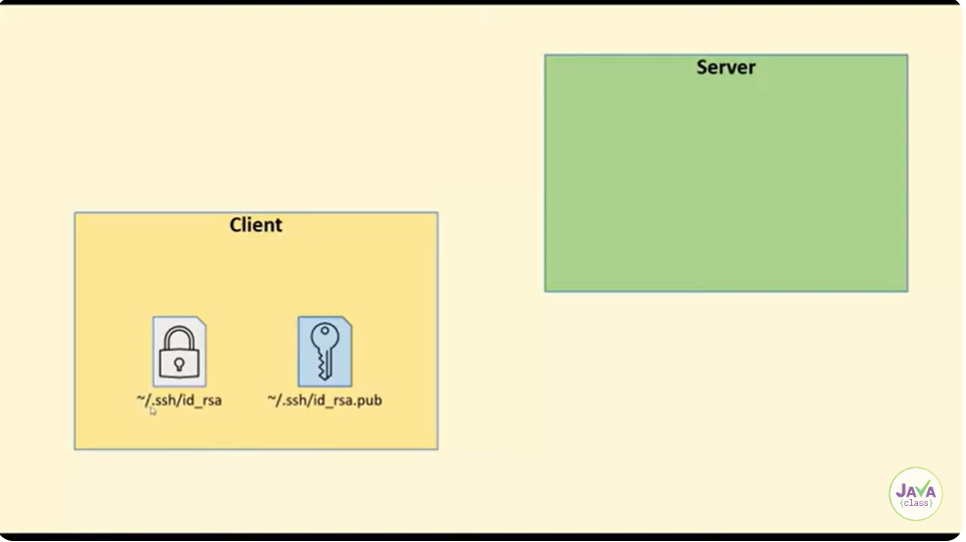
**Генерация и добавление SSH ключей**

**Принцип работы по протоколу SSH**

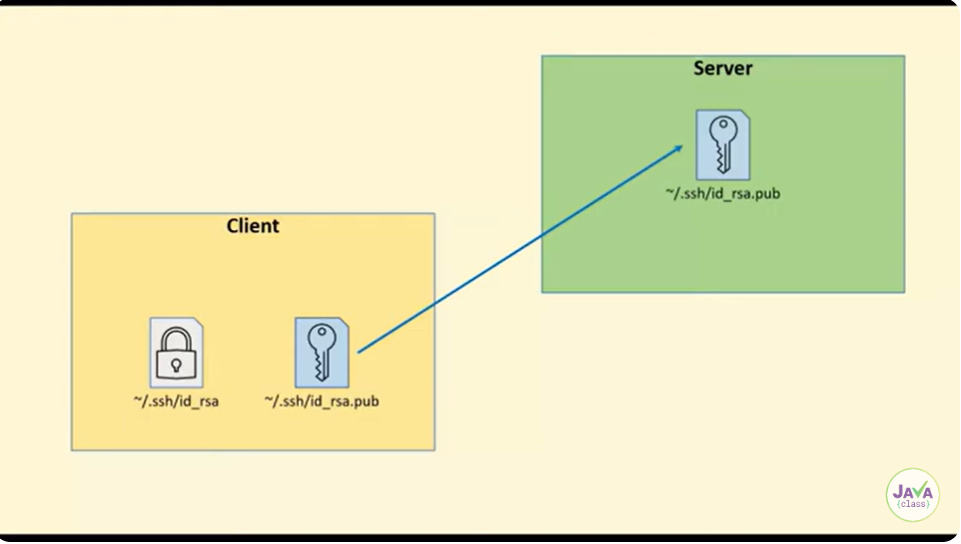
**SSH-ключ — безопасный способ соединения с сервером. Подключение по SSH с помощью ключа исключает риск, который связан с подбором и взломом вашего пароля.**

**Для аутентификации используются два ключа: приватный и публичный. Публичный ключ хранится на сервере в корневом каталоге, а приватный ключ остаётся на локальном компьютере в зашифрованном виде. Каждый раз, когда вы обращаетесь к серверу, происходит сопоставление ключей и идентификация клиента.**

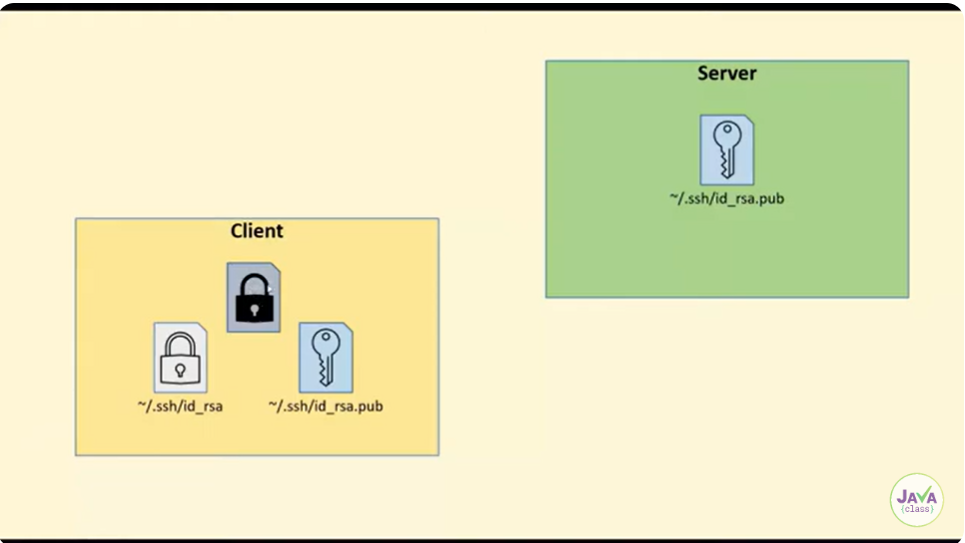
1. **Генерируем SSH-ключ (пару: секретный и публичный) на своем компе**

****

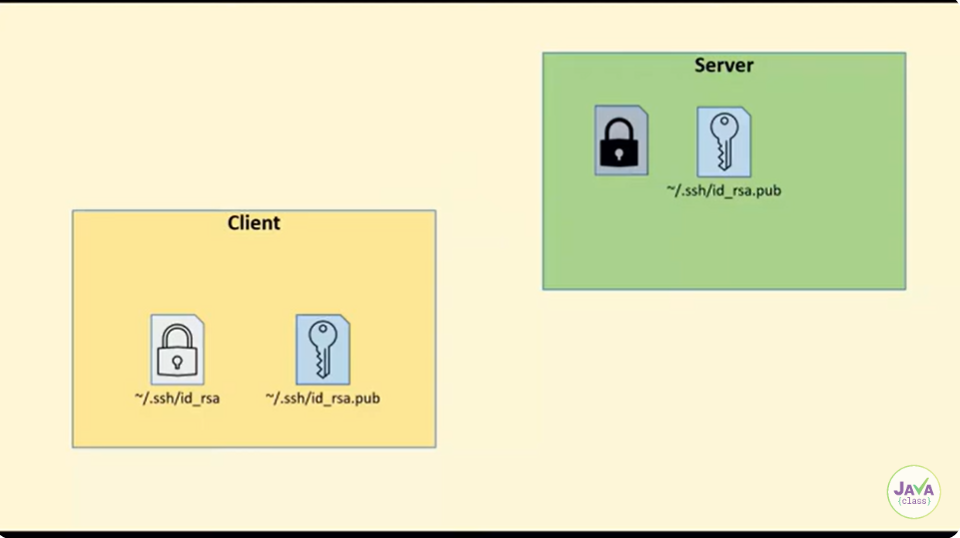
1. **Отправляем публичный SSH-ключ на сервер**

****

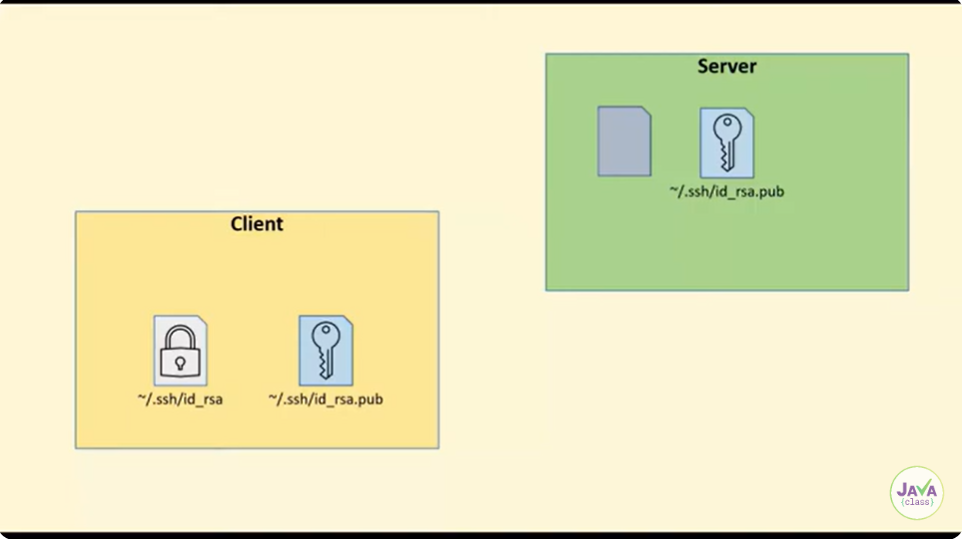
1. **Теперь при соединении с сервером программа, использующая протокол SSH на клиентском компе с помощью секретного ключа сгенерирует зашифрованный токен, который можно расшифровать только с помощью публичного ключа.**

****

1. **Сообщение (токен) отправляется на сервер, где хранится отправленный нами ранее публичный ключ.**

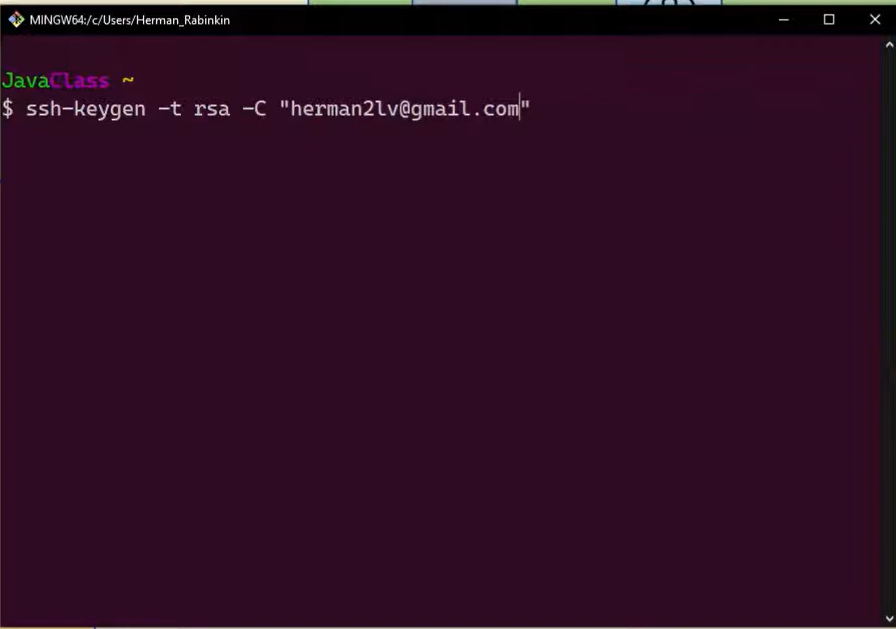
****

1. **Сервер открывает сообщение с помощью нашего публичного ключа и убеждается, что наш клиент подлинный.**

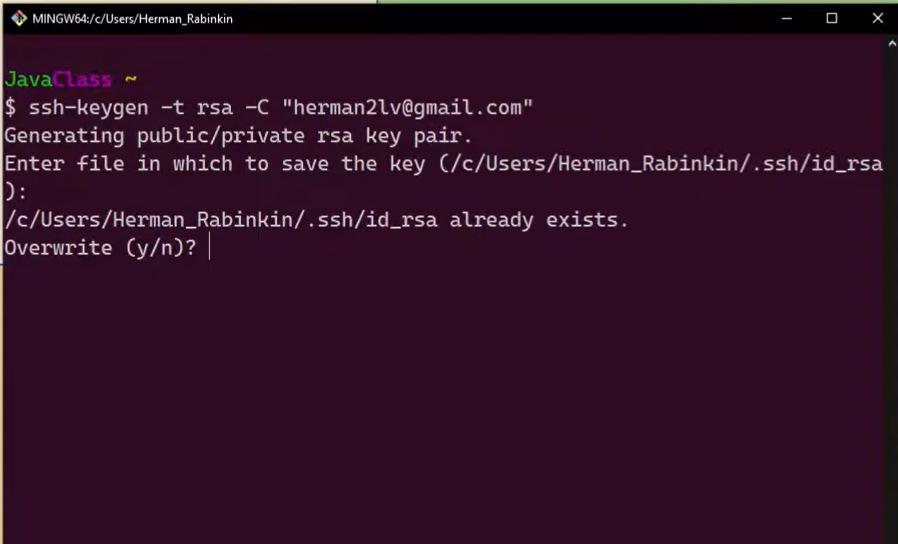
****

**Генерация ключей на клиенте**

1. **Ввести команду:**

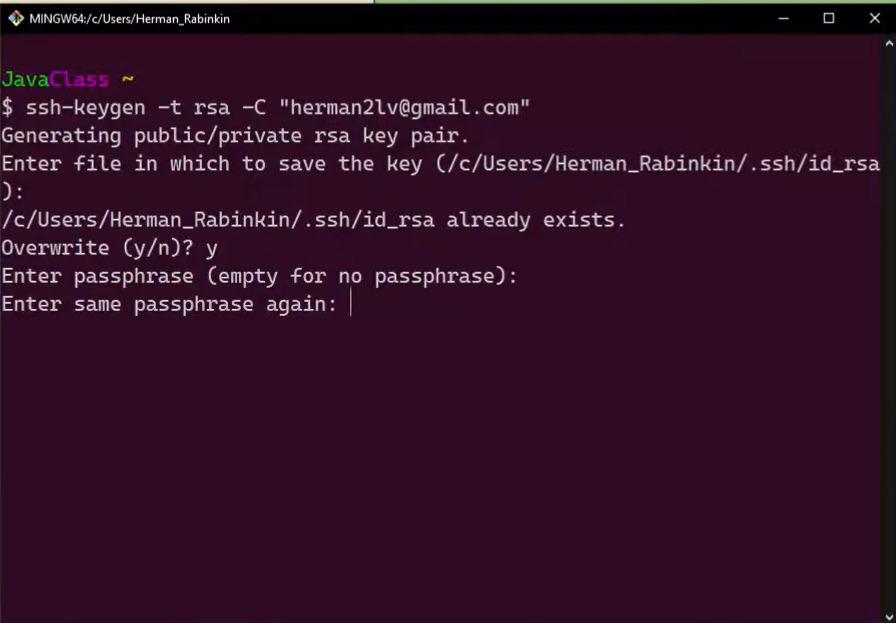
****

1. **Программа предложит выбрать место установки ssh-ключа и подскажет вариант:**

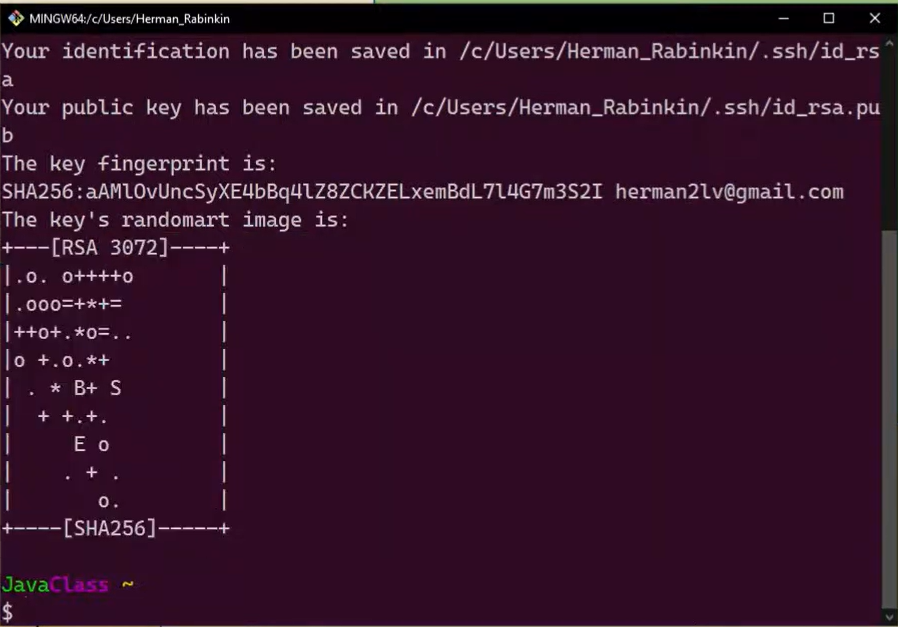
****

**Если у нас уже есть ключ в этой папке, программа попросит перезаписать его. Делаем выбор.**

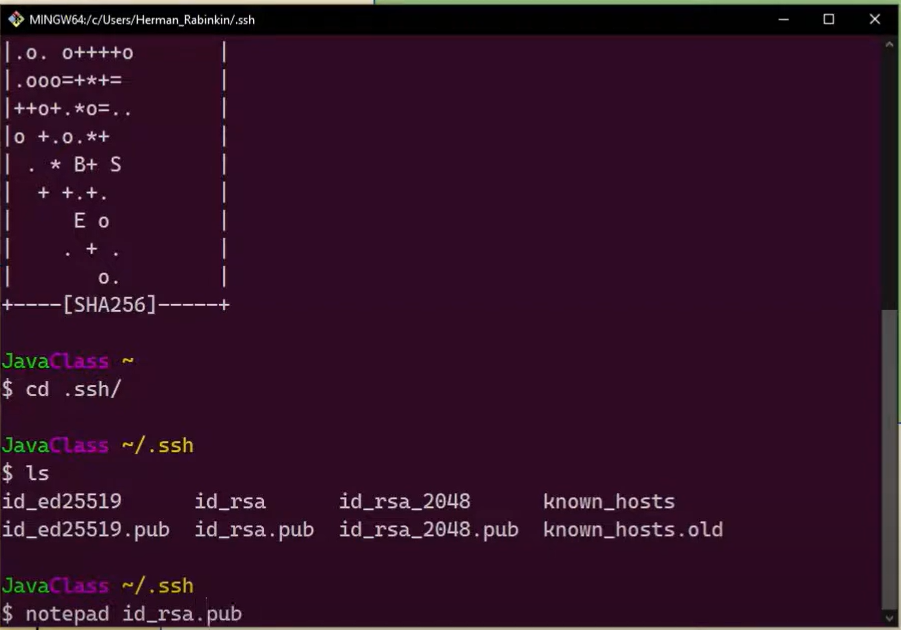
1. **Затем программа предлагает нам ввести кодовую фразу, которая будет использоваться каждый раз, когда мы будет передавать что-то на сервер по ssh протоколу. Если мы не боимся, что кто-то может войти в наш комп и воспользоваться параметрами пользователя, мы оставляем поле пустым. Далее программа просит повторить кодовую фразу, мы также нажимаем Enter при пустом поле. Если же мы вводим фразу, то следует обратить внимание на то, что поле в командной строке остается пустым и не показывает нам фразу, которую мы ввели, не смотря на то, что программа увидит её и примет.**

****

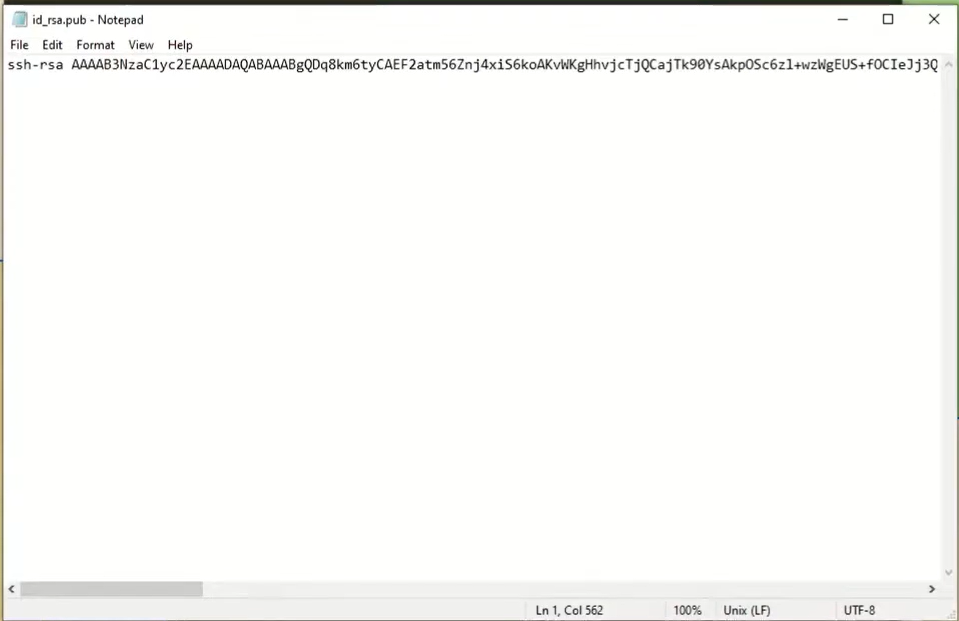
1. **Программа генерирует SSH-ключ, показывая некий отпечаток этого ключа в командной строке.**

****

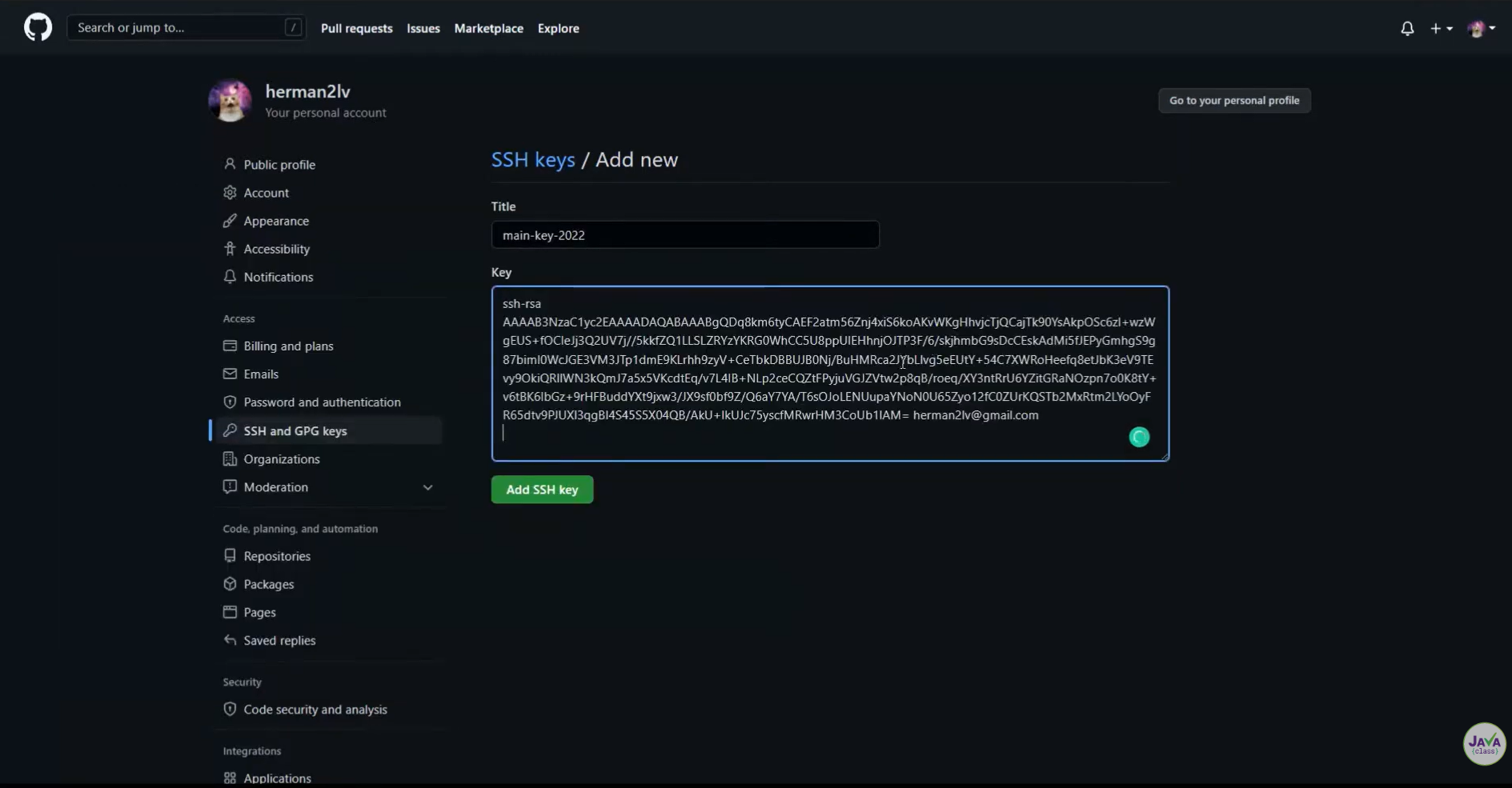
1. **Можно перейти в папку с ключами и убедиться, что ключ создан:**

****

1. **Если открыть публичный ключ (рекомендация: не трогать секретный ключ, чтобы не скомпрометировать его), его можно скопировать**

****

1. **Отправляем публичный ключ не сервер.**

****

**В данном случае сервером служит GitHub, в нашем аккаунте заходим в настройки, переходим в раздел SSH and GPG-key, добавляем новый SSH-ключ, даем ему некое название и вставляем в соответствующее поле скопированный на нашем клиенте публичный ключ.**

**Примечание 1. Данный сгенерированный ключ мы можем использовать в любом месте и в любой ситуации, требующей или дающей возможность соединения по SSH протоколу.**

**Примечание 2. Для каждого клиента нужен свой SSH-ключ. Если у нас несколько компьютеров или пользователей – для каждого из них необходим свой SSH-ключ.**