

Бессонова Светлана 483 гр.

ПМ 01.01 Разработка программных модулей

Работа с системой контроля версий Git: Создание нового репозитория

1. Регистрируемся в GitHub (Рисунок 1).

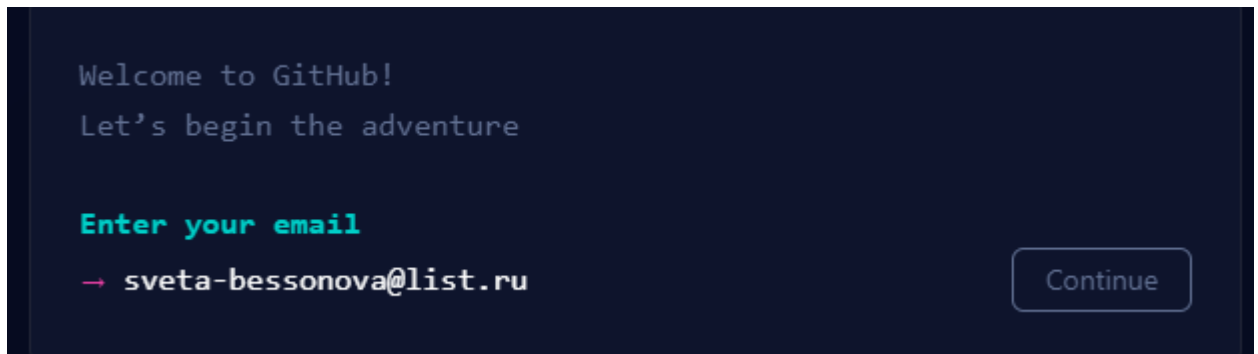




Рисунок 1. Регистрация в Github

2. Создаем репозиторий (рисунок 2).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner * Repository name *


 Cybergirl01 / TestProject1 

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-potato?](#)

Description (optional)

TestingProject1

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

[Create repository](#)

Рисунок 2. Создание репозиторий

3. Создаем проект в Visual Studio (рисунок 3).

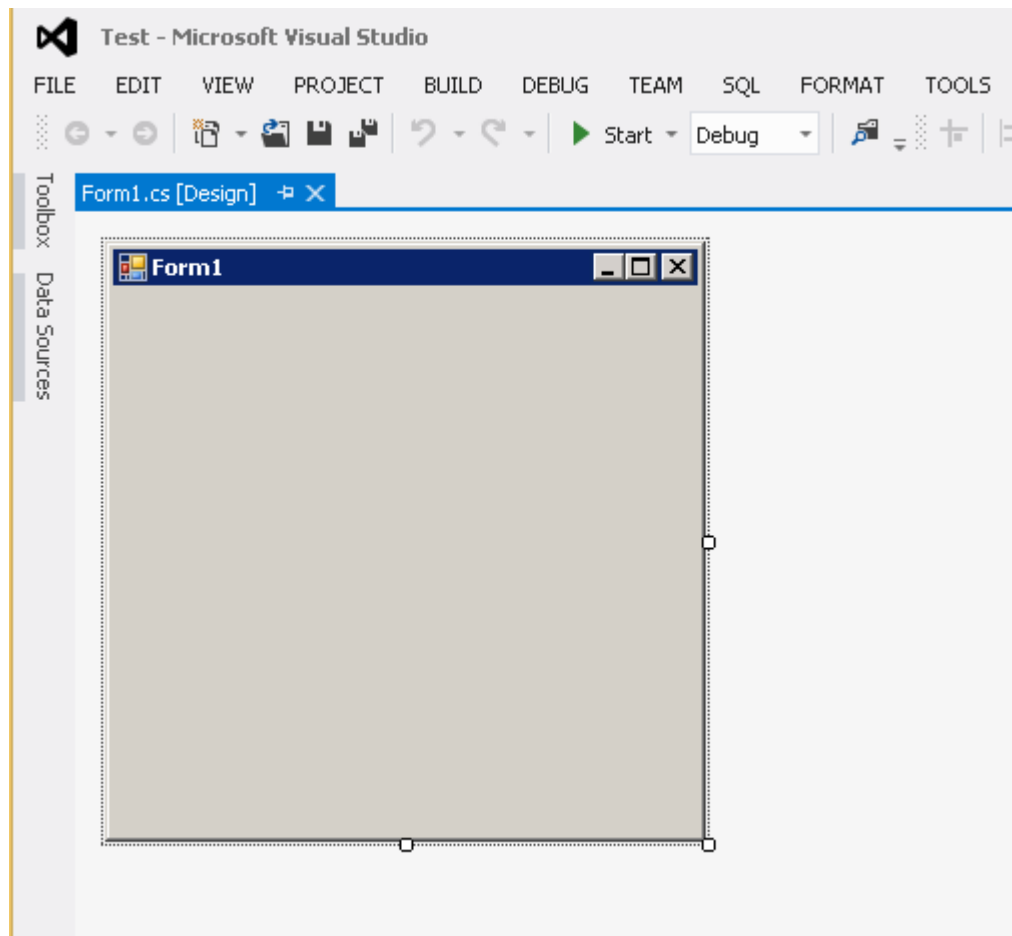


Рисунок 3. Создание проекта

4. Сохраняем проект и запускаем Git Bash в папке проекта (рисунок 4).

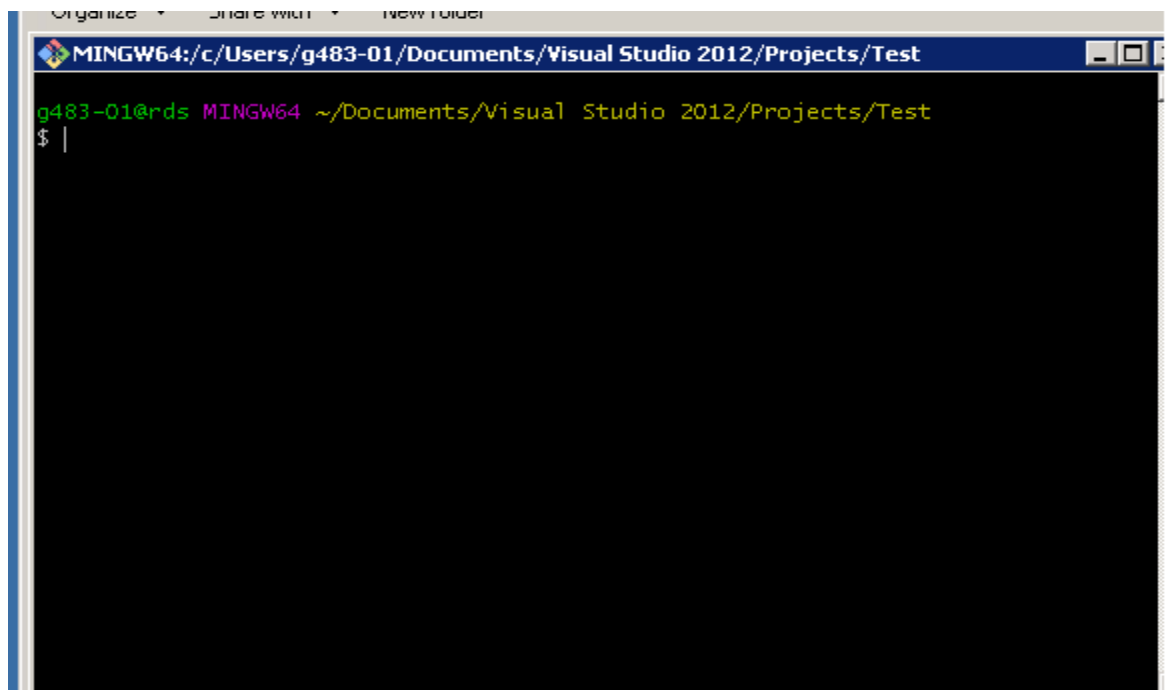
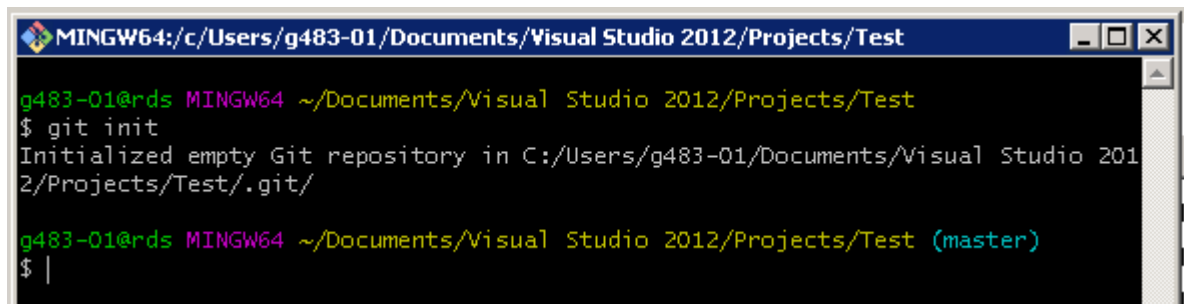


Рисунок 4. Запуск Git Bash

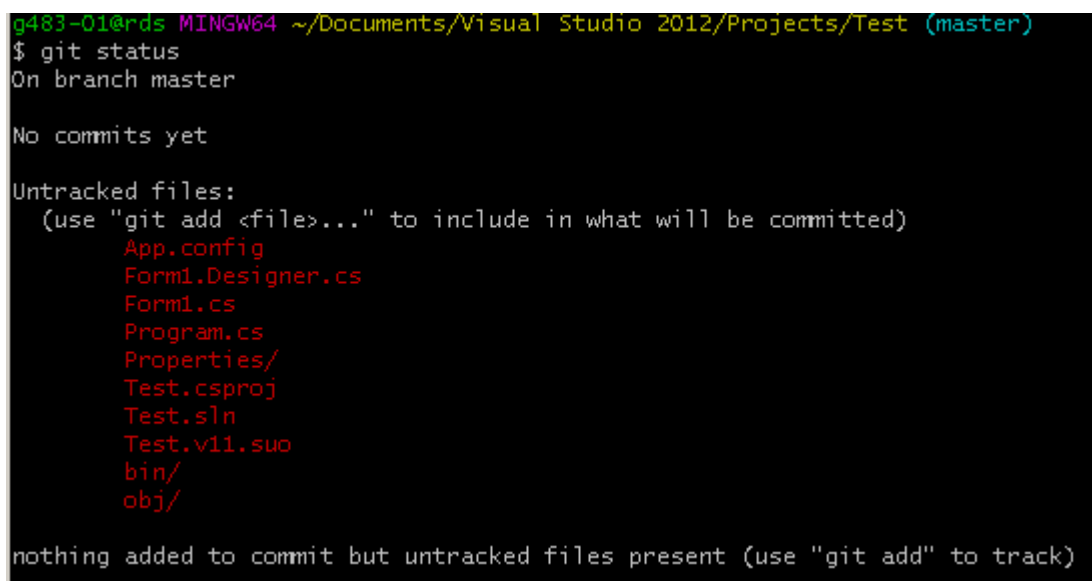
5. Выполняем инициализацию нового репозитория в текущей папке командой `git init` (рисунок 5). Эту команду выполняем только 1 раз.



```
MINGW64:/c:/Users/g483-01/Documents/Visual Studio 2012/Projects/Test
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test
$ git init
Initialized empty Git repository in C:/Users/g483-01/Documents/Visual Studio 2012/Projects/Test/.git/
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ |
```

Рисунок 5. Инициализация нового репозитория

6. Проверяем статус командой `git Status` (рисунок 6).



```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git status
On branch master

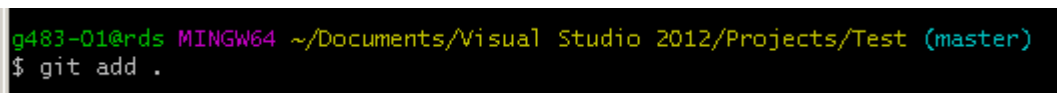
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    App.config
    Form1.Designer.cs
    Form1.cs
    Program.cs
    Properties/
    Test.csproj
    Test.sln
    Test.v11.suo
    bin/
    obj/

nothing added to commit but untracked files present (use "git add" to track)
```

Рисунок 6. Проверка статуса

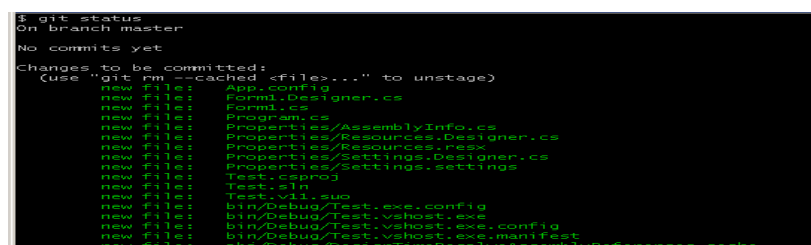
7. Добавляем все файлы для слежки изменений с помощью команды `git add .` (рисунок 7).



```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git add .
```

Рисунок 7. Добавление на отслежку

8. Проверяем командой `git status` (рисунок 8).



```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   App.config
    new file:   Form1.Designer.cs
    new file:   Form1.cs
    new file:   Program.cs
    new file:   Properties/AssemblyInfo.cs
    new file:   Properties/Resources.Designer.cs
    new file:   Properties/Resources.resx
    new file:   Properties/Settings.Designer.cs
    new file:   Properties/Settings.settings
    new file:   Test.csproj
    new file:   Test.sln
    new file:   Test.v11.suo
    new file:   bin/Debug/Test.exe.config
    new file:   bin/Debug/Test.vshost.exe
    new file:   bin/Debug/Test.vshost.exe.config
    new file:   bin/Debug/Test.vshost.exe.manifest
    new file:   obj/Debug/TemporaryAssemblyReferences.cache
```

Рисунок 8. Проверка статуса

9. Выполняем настройку в репозитории задав имя и адрес электронной почты используя команды `git config` (рисунок 9). Эту команду выполняем только 1 раз перед первой фиксацией изменений.

```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git config --local user.name "Cybergirl01"

g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git config --local user.email "sveta-bessonova@list.ru"
```

Рисунок 9. Настройка

10. Выполняем фиксацию изменений файлов в репозитории командой `git commit`, в кавычках после ключа `-m` указываем краткое описание изменений (рисунок 10).

```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git commit -m "created new project"
[master (root-commit) ab306c6] created new project
22 files changed, 477 insertions(+)
create mode 100644 App.config
create mode 100644 Form1.Designer.cs
create mode 100644 Form1.cs
create mode 100644 Program.cs
create mode 100644 Properties/AssemblyInfo.cs
create mode 100644 Properties/Resources.Designer.cs
create mode 100644 Properties/Resources.resx
create mode 100644 Properties/Settings.Designer.cs
create mode 100644 Properties/Settings.settings
create mode 100644 Test.csproj
create mode 100644 Test.sln
create mode 100644 Test.v11.suo
create mode 100644 bin/Debug/Test.exe.config
create mode 100644 bin/Debug/Test.vshost.exe
create mode 100644 bin/Debug/Test.vshost.exe.config
create mode 100644 bin/Debug/Test.vshost.exe.manifest
```

Рисунок 10. Фиксация изменений

11. Связываем локальный репозиторий с удаленным на сервере github командой `git remote add` (рисунок 11). Эту команду выполняем только 1 раз для каждого репозитория.

```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git remote add origin https://github.com/Cybergirl01/TestProject1.git
```

Рисунок 11. Связка с сервером Github

12. Отправляем выполненные фиксации в удаленный репозиторий командой `git push` (рисунок 12).

```
g483-01@rds MINGW64 ~/Documents/Visual Studio 2012/Projects/Test (master)
$ git push --set-upstream origin master
```

Рисунок 12. Отправка на сервер

13. В появляющихся окнах или в командной строке вводим имя пользователя от учетной записи github (рисунок 13).

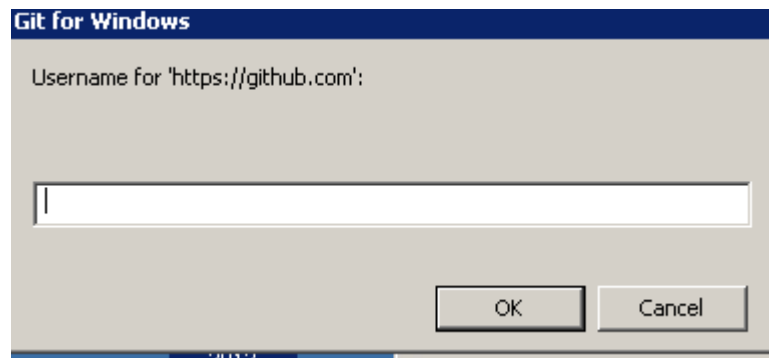


Рисунок 13. Ввод имени пользователя Github

14. Вместо пароля от учетной записи вводим ключ. Для этого создаем пользовательский ключ в GitHub (рисунок 14).

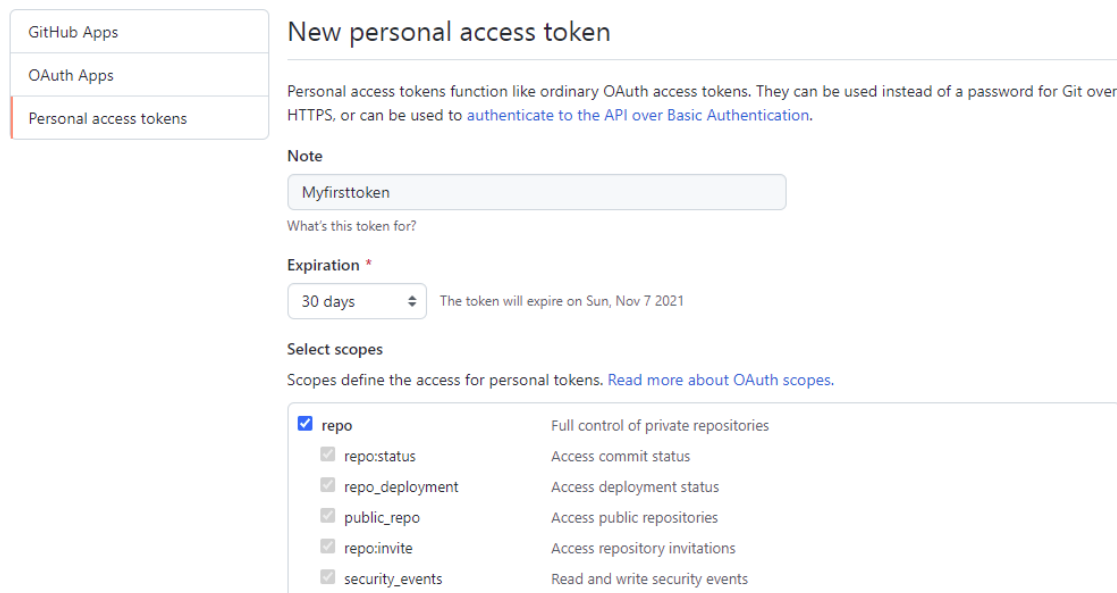


Рисунок 14. Создание пользовательского ключа

15. Вводим ключ вместо пароля и ждем результата (рисунок 15).

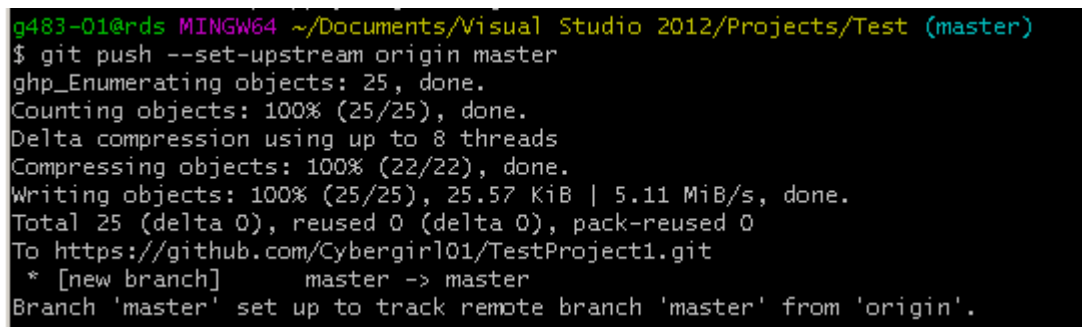


Рисунок 15. Результат команды git push

16. Обновите страницу репозитория в браузере и убедитесь в появлении на ней созданных файлов (рисунок 16).

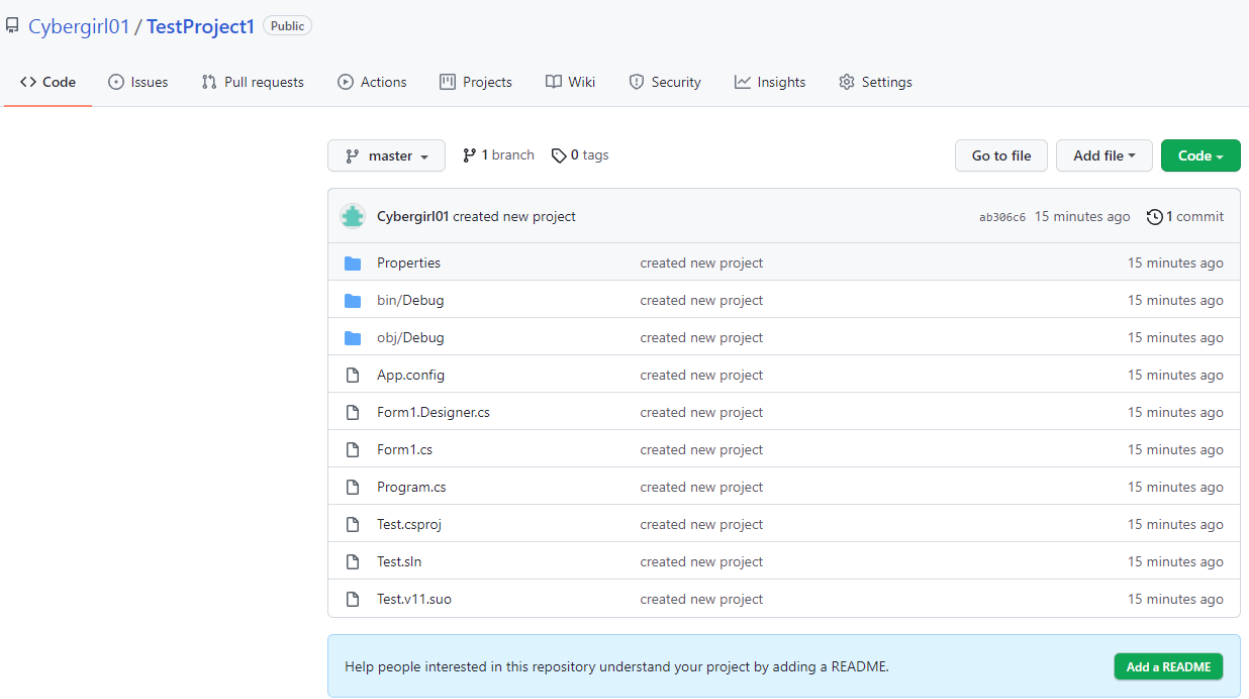


Рисунок 16. Страница репозитория