

Задание №1.

Создание репозитория

Описание задания:

1. Установите систему версионного контроля git. Для этого зайдите на сайт <http://git-scm.com/> и скачайте актуальную версию установщика. Во время установки будет много окон с выбором разных настроек – выбирайте все настройки по умолчанию.
2. Зарегистрируйтесь на сервисе онлайн-размещения репозитория GitHub по адресу <https://github.com/>.
3. Создайте репозиторий. Репозиторий, как правило, называют аналогично разрабатываемому приложению, в рамках лабораторных предлагается назвать репозиторий «Programming». Обращайте внимание на регистр букв и грамматические ошибки – их исправление в будущем может вызвать затруднения.
4. При создании репозитория **обязательно** укажите: а) закрытый репозиторий («Private repository»); б) добавление файла gitignore по шаблону Visual Studio (чекбокс «Add .gitignore» и пункт «Visual Studio» в появившемся выпадающем списке). По желанию можете добавить описание, файл README.md и лицензию.
5. Добавьте новую ветку develop, созданную на основе ветки main.
6. Сделайте ветку develop веткой по умолчанию (default) в настройках репозитория.
7. Склонируйте репозиторий на свой рабочий компьютер через Visual Studio. Для этого на вкладке Code вашего репозитория нажмите зеленую кнопку Code и скопируйте ссылку на ваш репозиторий. Ссылка должна оканчиваться на «*.git». Данную ссылку необходимо указать в Visual Studio как ссылку репозитория и задать папку локального репозитория на вашем рабочем компьютере.
8. При первом клонировании репозитория с GitHub, Visual Studio может запросить у вас логин и пароль для авторизации на сервисе. Введите логин и пароль от GitHub верно, иначе работа с репозиторием будет невозможна.
9. Убедитесь, что репозиторий был успешно клонирован. Для этого перейдите через Проводник в папку репозитория на вашем компьютере и проверьте, что в папке есть файл .gitignore и скрытая папка .git.
10. Создайте в папке локального репозитория новый текстовый файл и напишите в нём произвольный текст.
11. С помощью Visual Studio зафиксируйте новый файл в репозитории и синхронизируйте его с удаленным репозиторием (репозиторием на GitHub).

12. Убедитесь, что синхронизация репозитория прошла успешно. Для этого перейдите на GitHub и проверьте, что в ветке `develop` действительно появился новый текстовый файл, а фиксация добавилась в историю репозитория.

13. Теперь убедитесь, что любые изменения из удаленного репозитория синхронизируются с вашим компьютером. Для этого откройте ранее созданный текстовый файл через GitHub и добавьте в него какие-либо изменения (несколько новых строк с текстом). Зафиксируйте изменения в репозитории.

14. Откройте Visual Studio и нажмите кнопку синхронизации. После синхронизации текстовый файл на локальном компьютере должен совпадать с файлом из удаленного репозитория.

Типовые ошибки и возможные проблемы:

Наиболее частой проблемой является ситуация, когда репозиторий успешно клонирован и обновляется, но отправить фиксации с локального репозитория в удаленный не удастся. Есть несколько причин:

1. Неправильно указаны логин и пароль для подключения к GitHub в Visual Studio. Для смены пароля откройте окно «Справка -> Зарегистрировать Visual Studio». В открывшемся окне заново добавьте аккаунт GitHub (рис. 1.1).
2. На компьютер не установлен git. Инструменты git должны устанавливаться вместе с Visual Studio, но, если при установке компоненты не были выбраны, то их нужно установить самостоятельно. Просто выполните шаг 1 из списка заданий.
3. Вы пытаетесь синхронизироваться с чужим репозиторием, к которому у вас нет доступа.

Также могут возникнуть проблемы в случае работы не из-под Visual Studio или не операционной системы Windows. Решение проблемы подключения индивидуально и будет зависеть от сообщения об ошибке, которое вам укажет среда разработки.

Вне зависимости от возникших проблем, наиболее универсальным способом работы с системой versionного контроля, независимым от среды разработки или ОС, является работа через консоль (см. документацию на сайте <http://git-scm.com/>) или работа через сторонние приложения (также представлены на сайте <http://git-scm.com/>). Данный подход не так удобен, как работа напрямую из среды разработки, но он универсален и отказоустойчив.

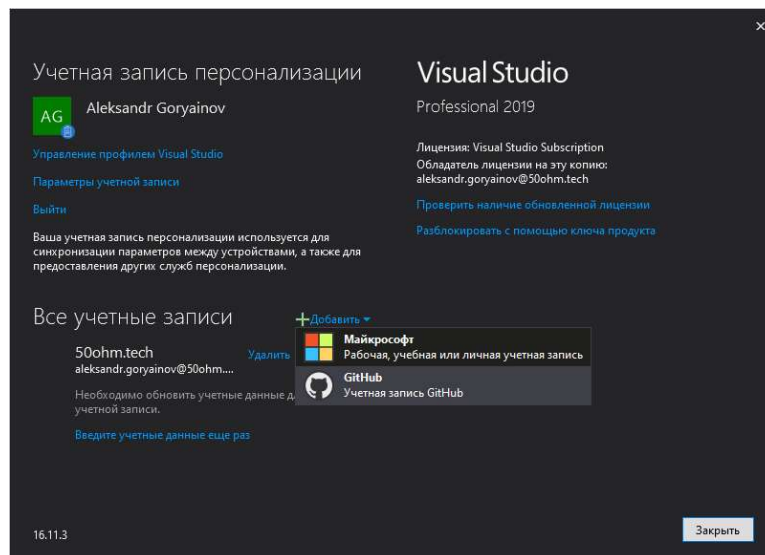


Рис.1.1 Ввод учетной записи GitHub в Visual Studio

Защита задания:

1. Во время защиты студент должен показать работу с репозиторием.
2. Репозиторий должен быть правильно согласно заданию.
3. В репозитории должен находиться файл «.gitignore», созданный по шаблону для Visual Studio.
4. В репозитории должна быть ветка develop, она должна быть обозначена как ветка по умолчанию (default).
5. Студент должен показать клонированный репозиторий в Visual Studio.
6. Студент должен внести изменения в произвольный текстовый файл в локальном репозитории, сделать фиксацию и синхронизировать изменения с удаленным репозиторием. Затем через GitHub показать, что изменения файла действительно теперь есть в удаленном репозитории.
7. Студент должен показать историю фиксаций своего репозитория.