# Создание проектов с использованием контроля версий в Visual Studio

- 1. Зарегистрируйте аккаунт на Github.com
- 2. Создайте новый репозитарий на Github.com

### Нажмите «New» (рис. 1)





Введите имя, например «Compiler». В списке Add .gitignore выберите «Visual Studio» и нажмите «Create repository» (рис. 2). Не изменяйте других настроек. Если не добавить правильный .gitignore будет много проблем!

Owner	Papacitory pama *		
SergeyVISorokir	n - / Compiler	~	
Great repository names Description (optional)	s are short memorable. N	eed inspiration? How about eff	ective-octo-funicular?
Public     Anyone can see t     Private     You choose who	this repository. You choose who ca can see and commit to this repos	ın commit. itory.	
	importing an existing reposito	ory.	



#### 3. Запомните ссылку на ваш репозиторий

Вы попадёте на страницу нового проекта, нажмите "Clone or download", там будет написана ссылка (рис. 3). Запомните эту ссылку.

SergeyVlSorokin / Compiler			o	Unwatch 👻 1	★ Sta	r 0 ¥Fork 0
↔ Code ① Issues 0 ۩ Pull requests 0	Projects 0 El Wiki	C Security	Insights	🔅 Settings		
No description, website, or topics provided. Manage topics						Edit
1 commit	₽ 1 branch	© 0	releases		<b>22</b> 10	contributor
Branch: master   New pull request			Create new file	Upload files	Find file	Clone or download <del>-</del>
SergeyVISorokin Initial commit			Clone	with HTTPS	0	Use SSH
).gitignore	Initial commit		Use Git	or checkout wi	th SVN usin	ig the web URL.
Help people interested in this repository unde	rstand your project by adding a RE	EADME.				

Рис. 3. Ссылка на репозиторий на Github

#### 4. Клонируйте репозиторий в Visual Studio

Запустите Visual Studio. Выберите пункт «Clone or check out code» (рис 4.).

🕅 File Edit View Project Build Debug Test Analyz	e Tools Extensions	Window Help	Search Visual Studio (Ctrl+Q)	
				×
What would you like to	do?			Filter Histor
Open <u>r</u> ecent		Ge	t started	
C:\Users\sergey\Source\Repos	08.10.2019 15:11		Clone or check out code Get code from an online repository like GitHub or Azure DevOps	
ConsoleApplication1.sln C:\Users\sergey\source\repos\NewRepo2\ConsoleApplication	08.10.2019 15:11 on1		റ്റ് Open a project or solution	
ConsoleApplication1.sln	08.10.2019 15:09		Open a local Visual Studio project or .sln file	

Рис. 4. Запуск клонирования репозитария в Visual Studio

Введите адрес репозитория, полученный на GitHub.com и путь, где проект будет находится на локальном диске и нажмите Clone (рис. 5).

Repository location			
https://github.com/SergeyVISorokin/Compiler.gi	đ		
Local path			
C:\Users\sergey\Source\Repos\Compiler		2**	
Browse a repository			
<ul> <li>Azure DevOps</li> </ul>			
G GitHub			

Рис. 5. Ввод размещения репозитория и его локальной копии

#### 5. Создайте проект в локальном репозитории

Откройте созданный локальный репозиторий в Team Explorer в Visual Studio. Если повезёт, там будет открыт правильный репозиторий и будет ссылка «Create new project or solution in this repository», нажмите её.

Если там открыт какой-то другой репозиторий нажмите на иконку с зелёной вилкой «Manage Connections» и выберите созданный репозиторий в списке (рис. 6).

Если у вас открыт правильный репозиторий, но нету ссылки «Create new project ...» просто щёлкните «New...» в списке решений (рис. 6.)

8.0.			• Debug	* X00
Team Ex	plorer - Home			<b>▼</b> ₽ ×
G 🖸	🕰 🛱 🛛 🖒			
Hor	Compiler			•
▲ Projec	t			
Ŀ	Changes	Ŷ	Branches	
↑↓	Sync		Tags	
₽	Settings			
Soluti New	ons .   Open   Show Folder Vi were no solutions found.	ew		

Рис. 6. Создание решения через Team Explorer

Выберите тип проекта, введите название проекта (например Scanner для лексического анализатора) и название решения (можно назвать Complier для всего компилятора).

Не создавайте решение и проекты в обход TeamExplorer, так как тогда необходимые файлы не будут добавлены под контроль версий, и их придётся добавлять руками.

### 6. Отправьте созданный проект на Github

В Team Explorer нажмите кнопку Changes. В текстовом поле ввода введите описание, что вы сделали (сейчас, например «Создан новый проект»), и нажмите стрелку справа от «Commit All». В меню выберите «Commit All and Push». Копка Commit не будет доступна, пока не введён текст (рис. 7).

Auto			
Team Explorer - Changes 🔹 🔻 🛠			
O ◯ ☆ ¥ C			
Changes   Compiler 🔹 🔊			
Branch: master			
Создан новый проект			
Commit All 🔹 Stash 🗸 Actions 🗸			
📤 Commit All 🛛 🕂 …			
Commit All and Push			
Commit All and Sync			
++ Scanner.cpp [add]			
💁 Scanner.vcxproj [add]			
Scanner.vcxproj.filters [add]			
g_ Compiler.sln [add]			
▲ Stashes (0)			
Drop All			
There are no stashed changes.			
Solution Explorer Class View Property Manager learn Explorer			
Test Fundamentary and All X			

Рис. 7. Ввод описания коммита

# Подключение к существующему проекту с другого компьютера

#### 7. Клонируйте проект

Всё совершенно аналогично пункту 4. Вы получите проект в том виде, в котором он есть на Github.

## Работа в ветках

Для внесения изменений в проекте обычно создаются ветки, работа ведётся в них, и когда изменение завершено, оно переносится в основную ветку проекта, например master.

#### 8. Создание ветки

Чтобы создать новую ветку зайдите в Team Explorer и нажмите «Branches». Нажмите «New Branch» и введите имя ветки (рис. 8). Нажмите «Create Branch».

Team Explorer - Branches	₹ų×
© ○ ☆ ¥   Ċ	
Branches   Compiler2	- 7
New Branch 👻   Merge 👻   Rebase 👻   Actions 👻	
make-some-change	
master	-
Checkout branch	
Create Branch Cancel	
▲ Active Git Repositories	
Type here to filter the list	P
🔺 🚸 Compiler2 (master)	
¶vn master	
🔺 🐻 remotes/origin	
👦 master	

Рис. 8. Создание ветки

Созданная ветка будет показана в нижней части окна. Текущая ветка выделяется жирным шрифтом. Ветки, которые находятся в дереве remotes/origin находятся на сервере (github), которые снаружи – на Вашем компьютере. Переключаться между ветками можно выбрав команду Checkout, которая доступна в всплывающем меню по правой кнопке мыши на имени ветки.

#### 9. Решите задачу

Напишите код, который делает что-то полезное.

#### 10. Передайте свой код на сервер

Перейдите в Team Explorer, зайдите в раздел Changes, сделайте Commit All and Push.

Пояснение:

- Соmmit сохраняет изменения в локальном репозитории на вашем компьютере. Рекомендуется регулярно делать Commit'ы, тогда, если дальше Вы сделаете ошибку, можно будет всегда откатиться к предыдущему состоянию, сохранённому через Commit. Или, например, переключится на другую ветку, поработать в ней и вернуться.
- Push передаёт изменения на удалённый сервер, с которого Вы исходно взяли свой код.

#### 11. Скачайте изменения других авторов

Зайдите в раздел Branches в Team Explorer. Переключитесь на ветку, изменения которой хотите получить или просто нажмите на ней правую кнопку мыши и выберите Pull.

#### 12. Объедините свою работу с работой других

Зайдите в раздел Branches в Team Explorer.

Нажмите «Merge» и выберите изменения из какой и в какую ветку надо перенести. Например, если Вы сделали какую-то работу в ветке make-some-change, то её нужно выбрать в списке «... from ...», а в «Into ...» скорее всего нужен master (рис. 9).

Нажмите кнопку Merge.

Если изменения, произошедшие со времени создания вашей ветки в самом master'е и в ветке затрагивают разные строки, то объединение будет выполнено полностью автоматически.



Рис. 9. Объединение веток

Если Git не сможет выполнить объединение, то Вы увидите сообщение о конфликте (рис. 10).



Рис. 10. Конфликт во время слияния

Щёлкните по ссылке «Conflicts..». Вы попадёте в список с местами конфликтов. Если щёлкнуть мышью по конкретному пункту списка, то там раскроется подменю с возможными действиями.

Чтобы разобраться с конфликтом нажмите «Merge» (рис. 11).

🖉 ♥ ♥   📸 ♥ 🎦 🖬 📑   🎔 ♥ ♥ ♥   Debug 🔹 x64	🔹 🕨 Local Windows Debugger 🔹 Auto 🔹 👘 🚽	1월 1월   📕 🕅 11 11 🚽 🕑 Live Share 🔐
Team Explorer - Resolve Conflicts     ▼ ₽ ×       ○ ○ ☆ ♥ ○	Merge - Compiler/Ster (both modified)*       +       X       Scanner.cpp*	Diff - Compiler/Sc/Scanner.cpp;master 🔒 🛎 🗙 👻
Resolve Conflicts   Compiler2	1 Conflicts (0 Remaining) Source: Compiler/Scanner/Scanner.cpp;make-some-change	Target: Compiler/Scanner/Scanner.cpp;master
Commit Merge Abort  A Conflicts (1)	5 6 int main()	5 6 int main()
Name Path ++ Scanner.cpp [both m Compiler\Scanner Merge Jel Conflicting content changes have been made.	<pre>7 { 8 [] std::cout &lt;&lt; "Hi again!\n"; 9 } 10 11 // Bun program: [tn] + E5 on Dahug &gt; Start</pre>	7 { 8
Edited on make-some-change   Diff   Take Source Edited on master   Diff   Keep Target	Result: Compiler/Scanner/Scanner.cpp 5 6 int main()	
	<pre>/ 1 8 std::cout &lt;&lt; "Hi again!\n"; 9 }</pre>	
	120 % 👻 🕐 No issues found 🔹	

Рис. 11. Отображение конфликта слияния

Какой вариант попадёт в результат слияния можно выбрать «галками» слева от текста, в том числе можно поместить оба варианта один за другим. В случае необходимости можно внести изменения в конечный результат вручную в нижней части окна.

Чтобы принять финальный вариант нажмите «Accept merge».

После решения всех конфликтов обязательно нажмите «Commit Merge» чтобы зафиксировать результат. Чтобы сразу отправить его на сервер используйте «Commit and Push». Впрочем, перед отправкой на сервер не будет лишним проверить работоспособность программы.