

Ціль : Ознайомитись з можливостями Visual Studio по керуванню контролем версій за допомогою Git та Github

Завдання 1. ОПАНУВАТИ ТЕХНОЛОГІЇ Git для РОБОТИ з ЛОКАЛЬНИМ РЕПОЗИТОРІЄМ

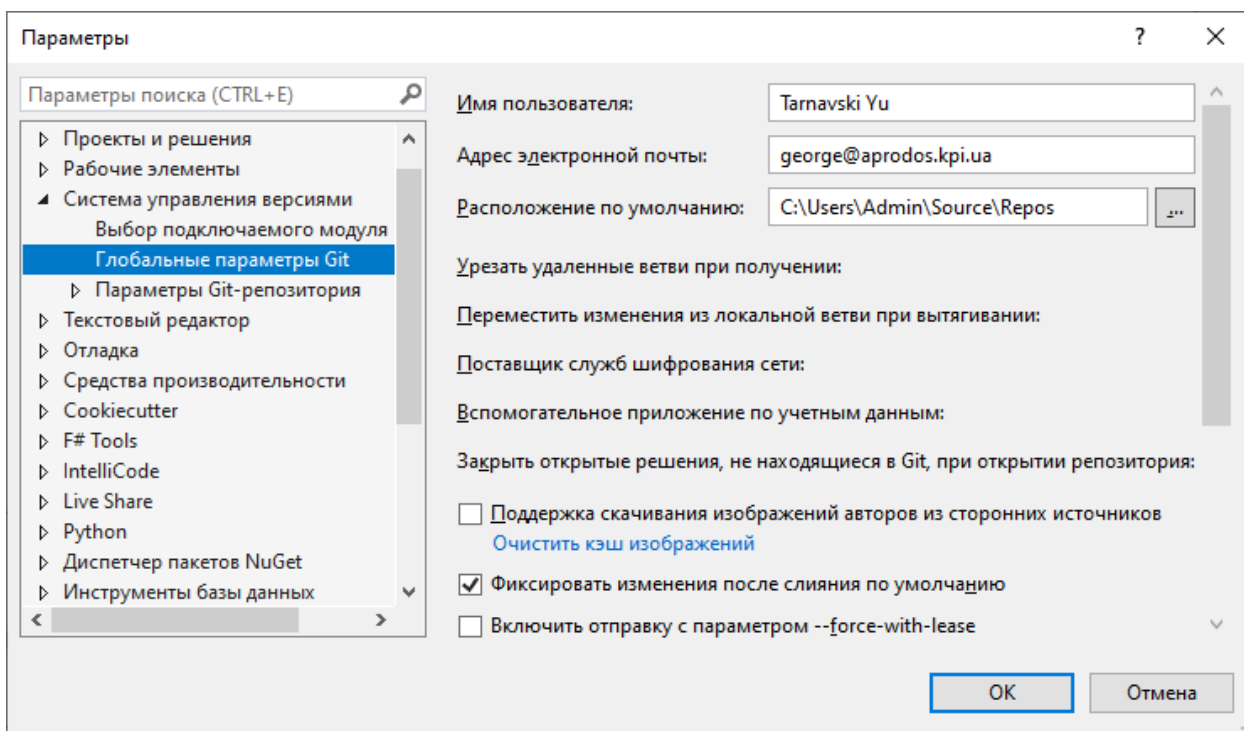
ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Git - це набір консольних утиліт для відстежування і фіксації змін в файлах. Найчастіше мова йде про вихідний код програм, але можна використовувати його для будь-яких файлів. З його допомогою можна відкотитися на більш стару версію проекту, порівнювати, аналізувати, зливати зміни і багато іншого. Цей процес називається *контролем версій*.

Починаючи з Visual Studio 2013 Update 1, користувачам Visual Studio доступний Git-клієнт, вбудований безпосередньо в IDE. Підтримка Git у Visual Studio 2013 була істотно перероблена порівняно зі старою версією, і в результаті вдалося отримати кращу інтеграцію Visual Studio та Git.

Початкові налаштування

Щоб настроїти параметри Git, перейдіть до **Git>Параметри** в рядку меню і виберіть **> Система керування версіями > Глобальні параметри Git** та виберіть потрібні установки:



Ім'я користувача та адреса електронної пошти, які надаються тут, будуть використовуватися як відомості для будь-якої фіксації.

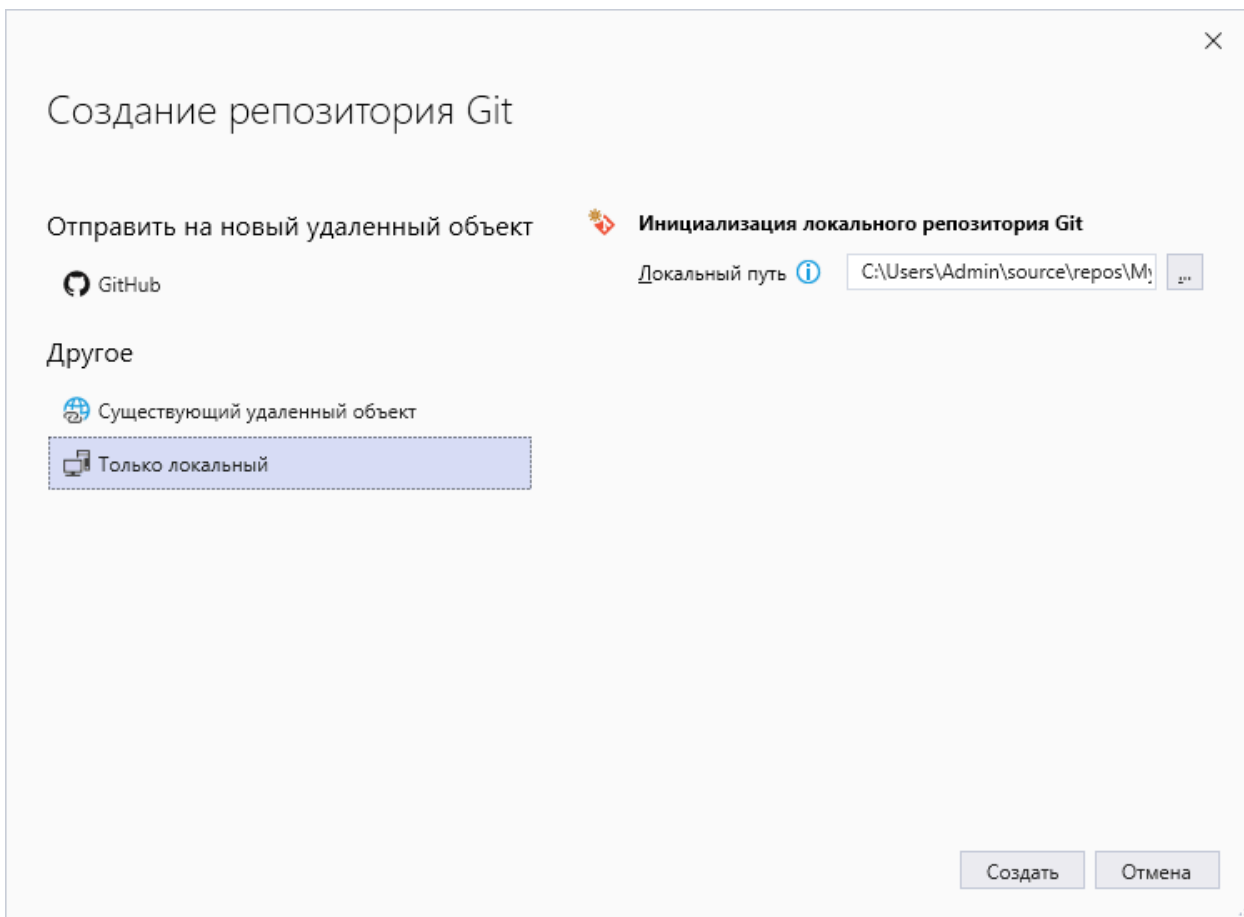
Створення локального репозиторію

Git зберігає свої файли та історію в прихованій папці проекту **.git**, яка створюється автоматично при передачі проекту під управління Git і називається *репозиторієм Git*. Репозиторій Git – це віртуальне сховище проекту, у якому можна зберігати версії коду для доступу за необхідності.

Visual Studio спрощує створення репозиторію безпосередньо з інтегрованого середовища розробки.

Для створення локального репозиторію Git у відкритому проєкті:

1. Відкрийте в Visual Studio проєкт.
2. У меню Visual Studio **Git** виберіть "**Створити репозиторій Git**".
3. У діалоговому вікні "**Створення репозиторію Git**" у розділі "**Надіслати**" виберіть "**Тільки локальний**" і натисніть кнопку "**Створити**".



Швидко відкрити вікно "**Створення репозиторію Git**" дозволяє також гіперпосилання "**Додати в систему керування версіями**" в правому нижньому вуглі вікна Visual Studio.

Щоб повідомити Git, які файли та каталоги не включати під контроль системи, тобто ігнорувати, у кореневому каталозі свого сховища створюється текстовий файл **.gitignore**. Git підтримує офіційний список рекомендованих **.gitignore**-файлів для багатьох популярних операційних систем, середовищ і мов. Тому, як правило, необхідності в редагуванні згенерованого Visual Studio **.gitignore** не виникає.

Після створення репозиторію або клонування репозиторію Visual Studio виявляє репозиторій Git і додає його до списку локальних репозитаріїв у меню **Git**.

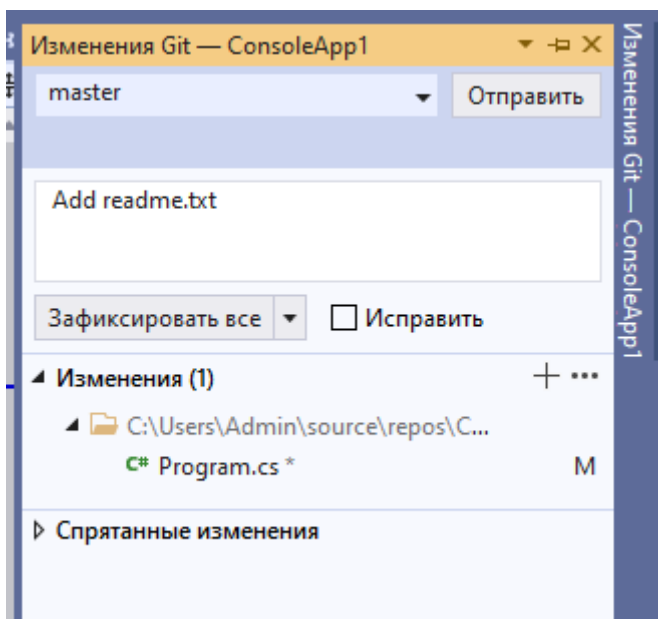
Фіксація змін

Основна частина будь-якого робочого процесу Git полягає у зміні файлів та **фіксації** змін у цих файлах. Ці фіксації додаються за замовчуванням в гілку **master**.

Для створення фіксації:

1. Внесіть необхідні зміни в файли проєкту, які будуть фіксуватись.
2. Виконайте команду **Git>Фіксація або приховування...**

- В панелі “Зміни Git” задайте пояснювальний текст для фіксації і натисніть кнопку **Зафіксувати все**.

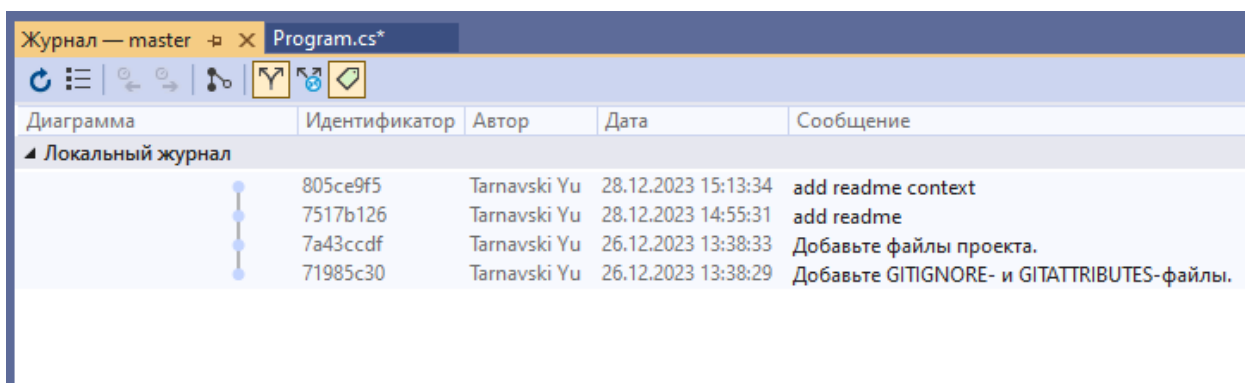


Панель “Зміни Git” в згорнутому вигляді постійно відображається також з правого боку вікна Visual Studio, що більш зручно для швидкого створення фіксації.

В ході створення фіксація отримує універсальний цифровий ідентифікатор (хеш-код), який може використовуватись для переключення на зафіксований нею стан - *ревізію*.

Перегляд історії і перемикання на ревізію

Переглянути ревізії в поточній гілці можна в журналі, який відкривається по команді **Git>Переглянути журнал гілок**:

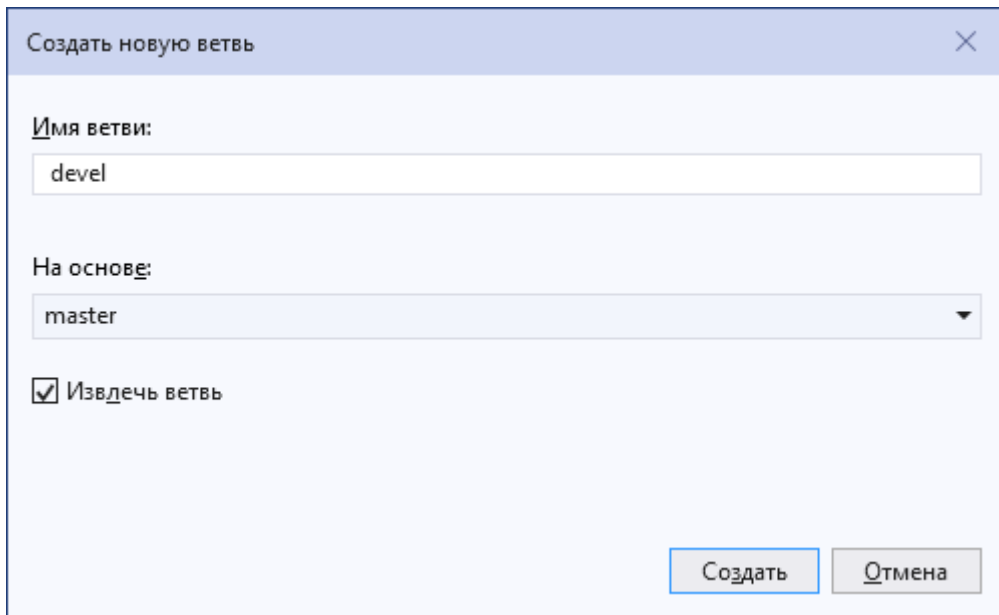


Графічних засобів для переходу до вибраної ревізії в Visual Studio не передбачено. Пропонується використовувати для цього консольну утиліту Git **checkout**:

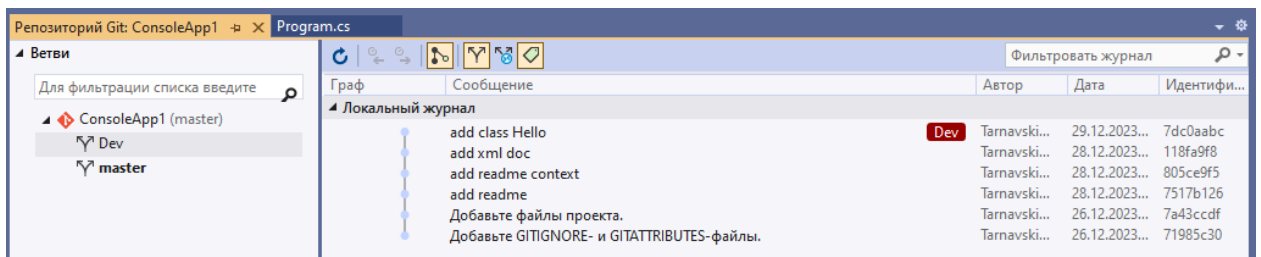
- Виконайте команду **Git>Відкрити в командному рядку**.
- Введіть в командному рядку команду: **git checkout <commit>**, де **<commit>** - хеш-код фіксації (наприклад, **git checkout 805ce9f5**).
- Натиснути клавішу **<Enter>**.

Робота з гілками

Крім гілки фіксацій **master** користувач може створювати власні гілки за допомогою команди **Git>Створити локальну гілку з ...**, яка відкриває діалог для введення назви гілки і вибору тієї гілки, яка вибирається в якості основи:



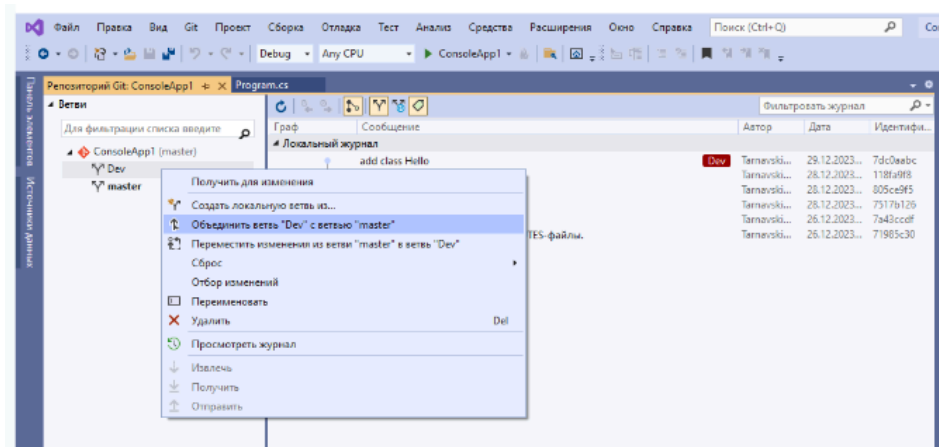
З кількома гілками зручніше працювати у вкладці **Репозиторій Git**, яка відкривається по команді **Вид>Репозиторій Git (Ctrl+O, Ctrl+R)** або **Git>Керування гілками**:



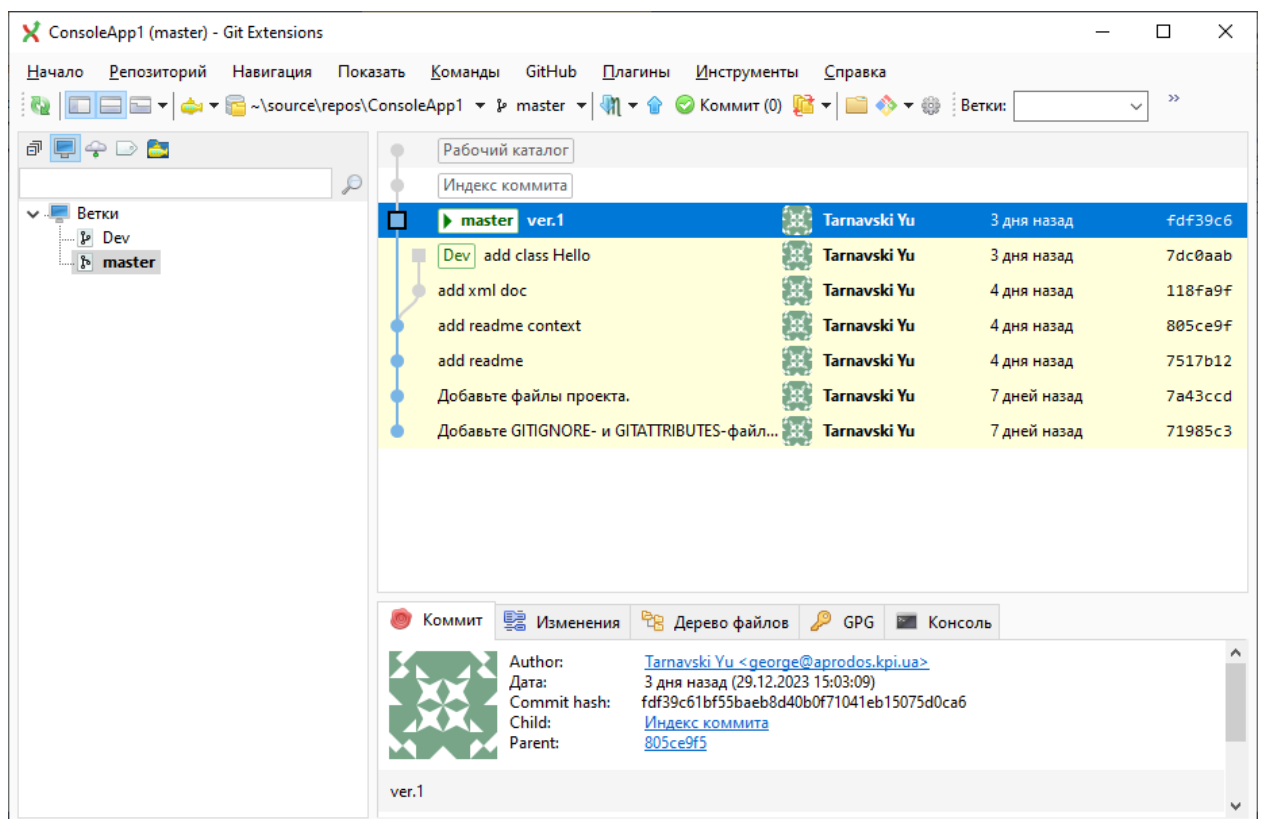
В ній відображається журнал фіксацій для поточної гілки. Вибір гілки здійснюється за допомогою подвійного кліку по назві гілки.

Гілки можуть об'єднуватись. Для цього у вкладці **Репозиторій Git** треба:

1. Вибрати гілку, з якою передбачається об'єднання.
2. Клацнути правою кнопкою миші по назві гілки, яка має об'єднуватись.
3. Викликати з контекстного меню команду **Об'єднати гілку <...> з гілкою <...>**



Слід зазначити, що для більш комфортної роботи з гілками і ревізіями можна скористатися спеціальною графічною утилітою **Git Extension** (<https://gitextensions.github.io/>):



Хід виконання

1. Створіть у Visual Studio новий проект за шаблоном консольного додатка. При цьому утворюється простий додаток, який виводить рядок тексту "Hello World!".
2. Перевірте правильність налаштування глобальних параметрів Git та в разі необхідності задайте ім'я користувача і його адресу електронної пошти.
3. Створіть локальний репозиторій Git.
4. Відкрийте папку проекту і переконайтесь в наявності прихованої папки репозиторію **.git**.
5. Відшукайте в папці проекту файл **.gitignore**, відкрийте його в текстовому редакторі і переконайтесь у правильності заданих налаштувань.

6. Додайте в проект текстовий файл **readme.txt** та зафіксуйте цю зміну в автоматично створеній гілці **master**.
7. Додайте в файл **readme.txt** контекстну інформацію про проект і також зафіксуйте цю зміну.
8. Додайте в код програми (**Program.cs**) коментарі для створення XML-документації і також зафіксуйте цю зміну.
9. Відкрийте журнал фіксацій і переконайтесь в наявності створених ревізій в гілці **master**.
10. Створіть нову гілку **dev** на основі гілки **master**.
11. Модифікуйте код програми (**Program.cs**), створивши в ній клас **Hello** для виведення привітання і зафіксуйте цю зміну.
12. Об'єднайте гілку **dev** з гілкою **master**.
13. Додайте в файл **readme.txt** інформацію про дату створення версії №1 програми і також зафіксуйте цю зміну як першу версію програми.
14. Перейдіть до однієї з попередніх ревізій і переконайтесь, що стан програми повністю відновлюється до стану фіксації.
15. Задokumentуйте перехід до ревізії за допомогою скріншота в звіті про виконання цієї лабораторної роботи.
16. Повторіть п.п.14-15 для інших ревізій.
17. Поверніться на гілку **master**.

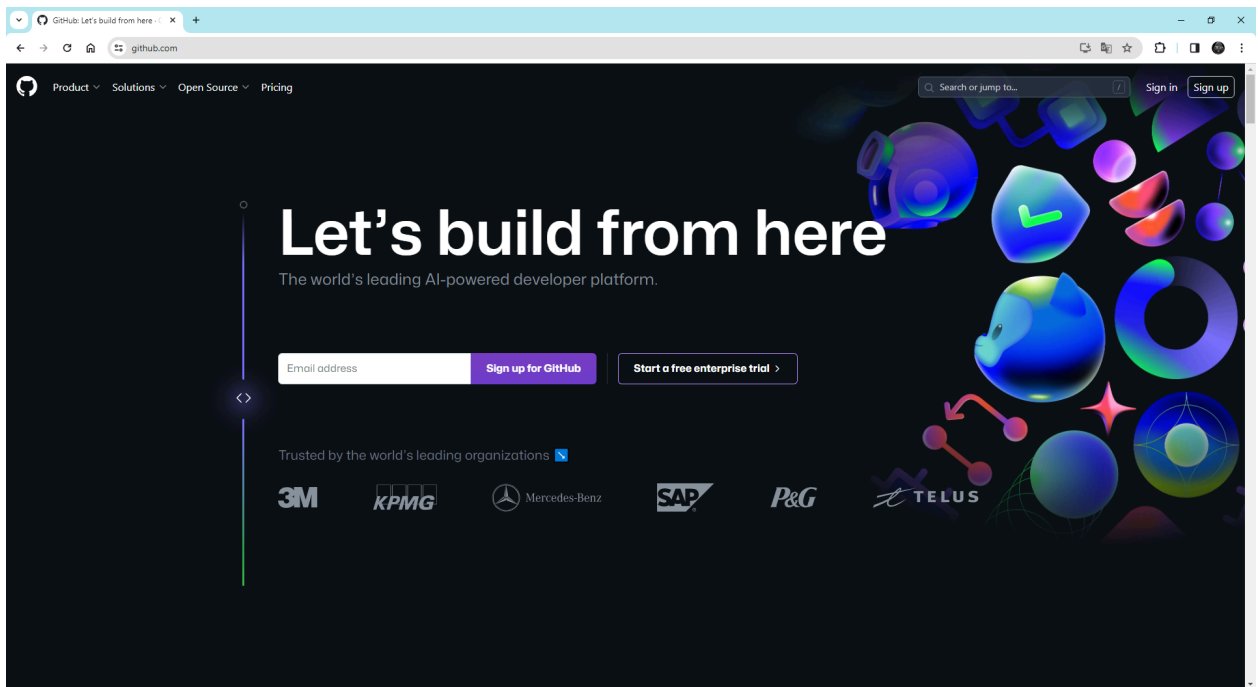
Завдання 2. Опанувати технології Git для роботи з віддаленим репозиторієм

ОСНОВНІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Для роботи з віддаленим репозиторієм на GitHub необхідно мати обліковий запис на GitHub.

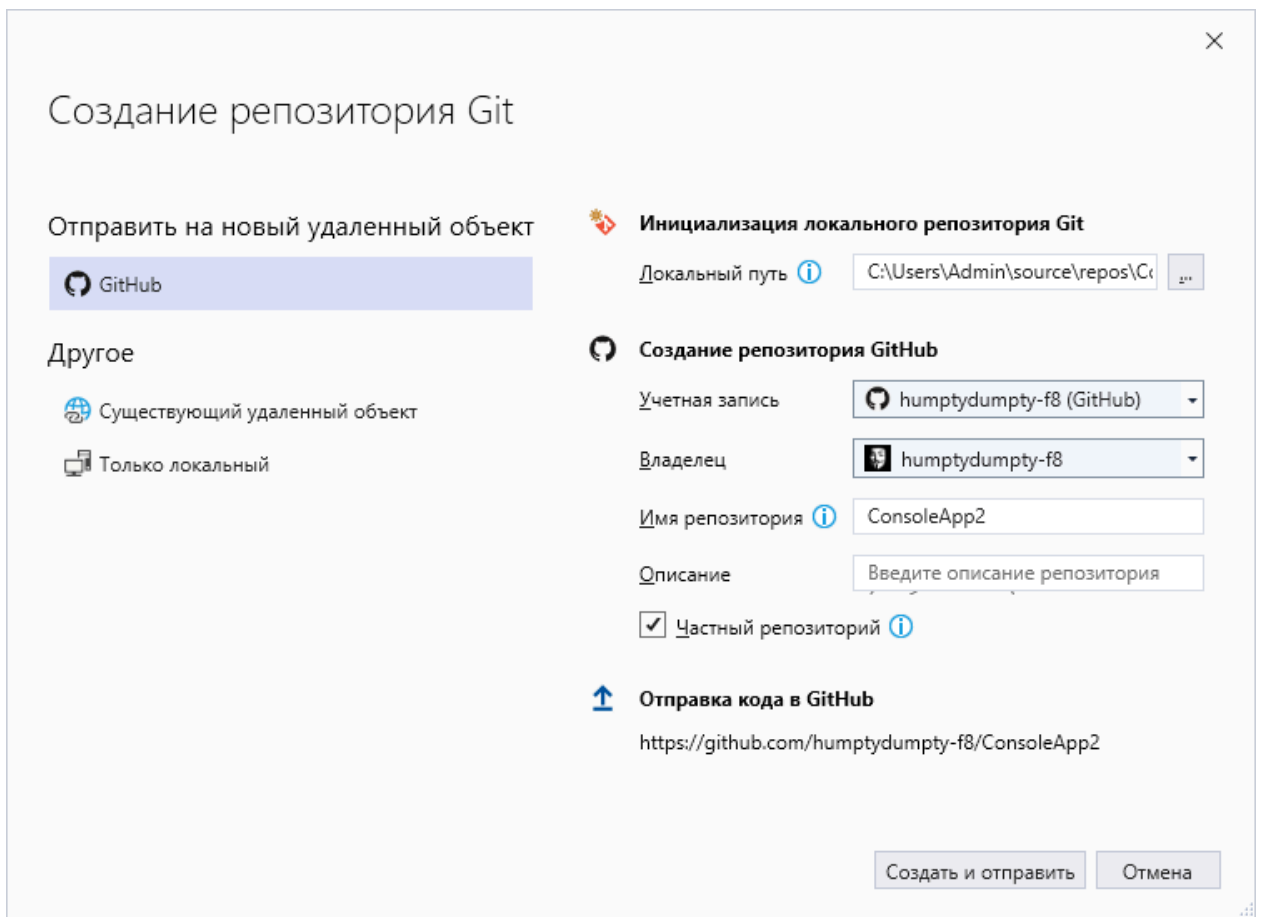
Якщо у вас ще немає облікового запису GitHub, створіть його. Для цього:

1. Відкрийте <https://github.com> у веб-браузері та натисніть кнопку "**Зареєструватися**".
2. Введіть адресу електронної пошти.
3. Створіть пароль для нового облікового запису GitHub та введіть ім'я користувача.
4. Виберіть, чи хочете ви отримувати оновлення та оголошення електронною поштою, а потім натисніть кнопку "**Продовжити**".
5. Перевірте свій обліковий запис, вирішивши головоломку.
6. Після перевірки параметрів облікового запису натисніть кнопку "**Створити обліковий запис**".
7. GitHub надсилає код запуску на адресу електронної пошти.
8. Введіть код запуску у діалоговому вікні "Введення коду", а потім натисніть клавішу ENTER.
9. Виберіть, чи використовувати безкоштовний обліковий запис або обліковий запис групи. Щоб вибрати безкоштовну обліковий запис, натисніть кнопку "**Пропустити персоналізацію**".
10. GitHub відкриває персональну сторінку в браузері.

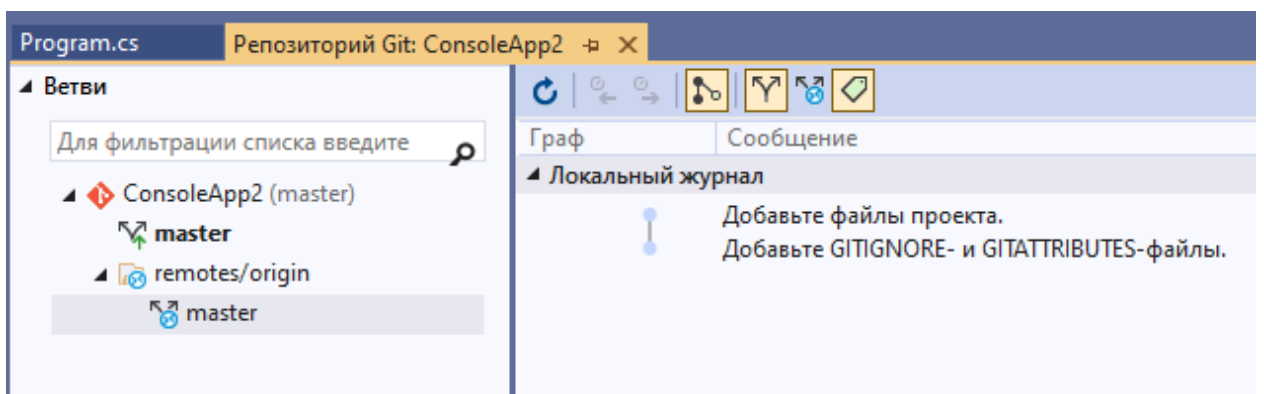


Для створення віддаленого репозиторію GitHub:

1. Відкрийте в Visual Studio проект.
2. У меню Visual Studio **Git** виберіть "**Створити репозиторій Git...**".
3. У діалоговому вікні "**Створення репозиторію Git**" у розділі "**Відправити на новий віддалений об'єкт**" виберіть **GitHub**.
4. Після введення необхідних параметрів репозиторію натисніть кнопку "**Створити і відправити**".



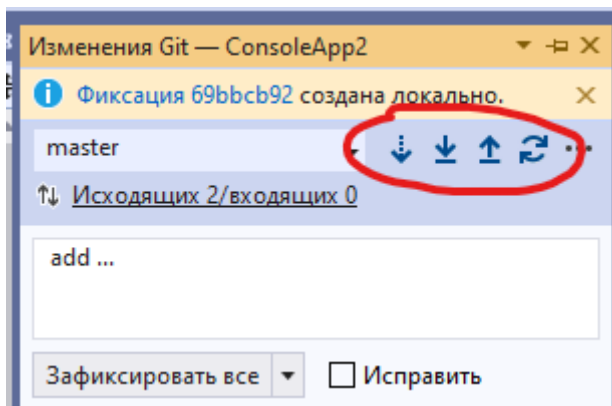
При цьому одночасно створюється локальний (з ім'ям локального проекту) і віддалений репозитарій (з ім'ям **remotes/origin**), які відображаються в журналі фіксацій:



Далі фіксації в гілку **master** локального репозиторію можуть додаватись звичайним чином за допомогою панелі змін, а для синхронізації з гілкою **master** віддаленого репозиторію використовуються команди з меню **Git**:

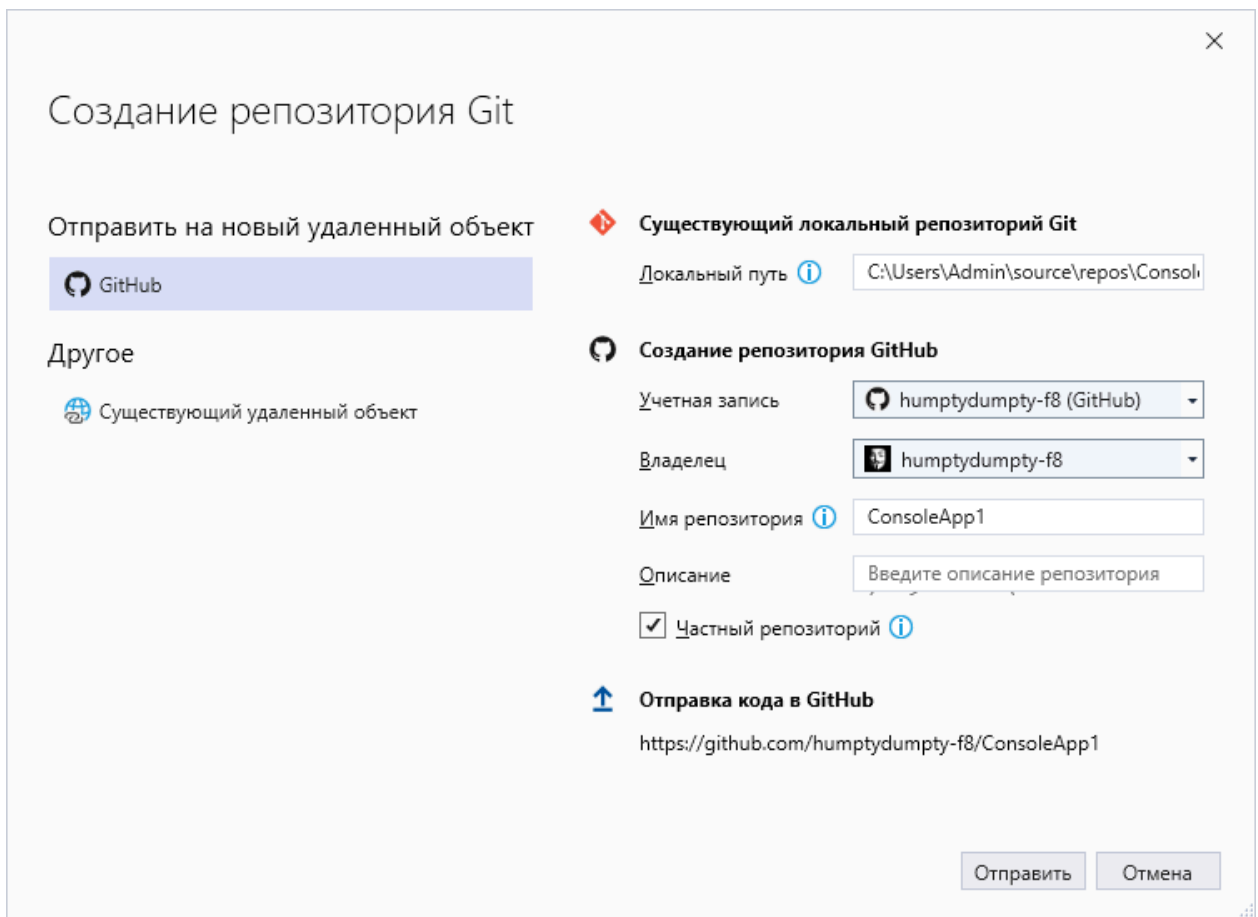
- **Вилучити** - перевіряє, чи віддалені фіксації можна включити в локальні зміни, і усуває конфлікти злиття.
- **Отримати** - перевіряє, чи віддалені фіксації можна включити в локальні зміни, і за відсутності конфліктів завантажуються.
- **Відправити** - фіксації з локальної гілки передаються у віддалену гілку.
- **Синхронізувати** - спочатку виконується команда **Отримати**, а потім - **Відправити**.

Ці команди можуть виконуватись також з панелі змін за допомогою кнопок у вигляді стрілок:



Віддалений репозиторій можна також створити шляхом публікації локального репозиторію Git в GitHub:

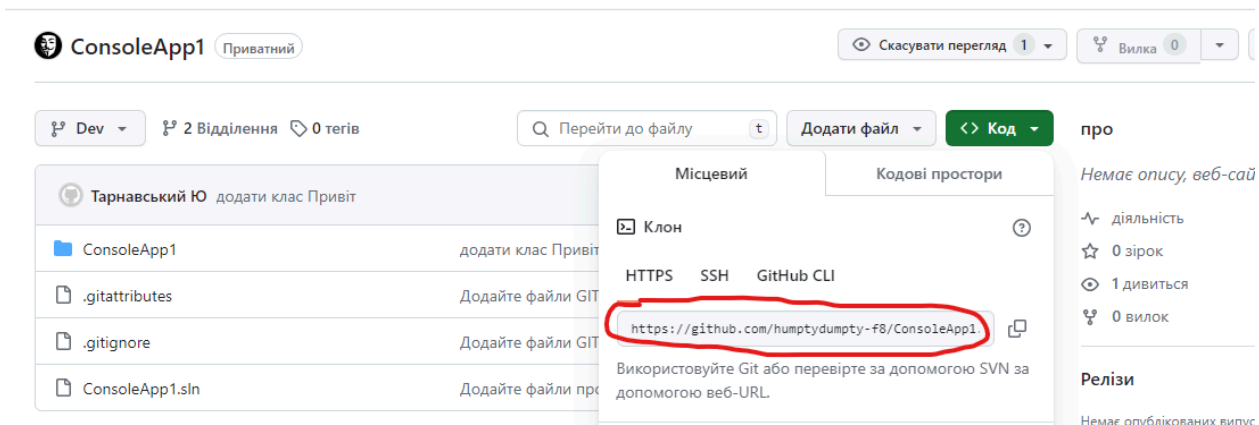
1. Відкрийте в Visual Studio проект.
2. У меню Visual Studio **Git** виберіть "**Відправити в службу Git...**".
3. У діалоговому вікні "**Створення репозиторію Git**" у розділі "**Відправити на новий віддалений об'єкт**" виберіть **GitHub**.
4. Після введення необхідних параметрів репозиторію натисніть кнопку "**Відправити**".
5. Відкрийте панель "**Зміни Git**", виберіть гілку фіксацій і натисніть кнопку панелі "**Відправити**".



Кожний віддалений репозиторій отримує власну URL-адресу. Для її отримання треба виконати наступні дії:

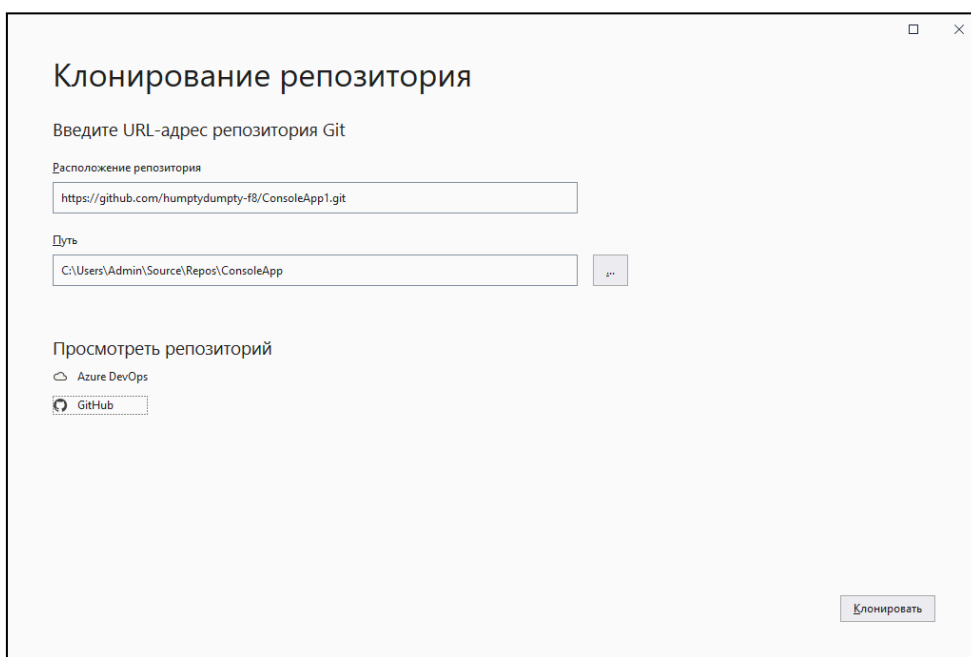
1. Перейдіть на свою персональну сторінку GitHub і в переліку проектів виберіть потрібний.
2. На сторінці проекту (див. рисунок) розкрийте список **Код**.

3. Натисніть кнопку копіювання праворуч від URL-адреси, щоб скопіювати її до буферу обміну.



Маючи URL-адресу проекту інший користувач може клонувати його в свій локальний репозиторій. Для цього необхідно:

1. Відкрийте Visual Studio.
2. У меню **Git** виберіть "**Клонувати репозиторій**".
3. У вікні клонування репозиторію (див. рисунок) у розділі "**Введіть URL-адресу репозиторію Git**" в полі "**Розташування репозиторію**" вставте URL-адресу. Потім у розділі "**Шлях**" можна прийняти шлях за промовчанням до локальних вихідних файлів або перейти до іншого розташування.
4. Натисніть кнопку "**Клонувати**".



Хід виконання

1. Отримайте у викладача URL-адресу репозиторію GitHub з прикладами до лекцій та виконайте його клонування.
2. Виберіть одну з ревізій в гілці прикладів до лекції для переходу в стан демонстрації прикладу і запустіть проект на виконання.

3. Задokumentуйте перехід до ревізії за допомогою скріншота в звіті про виконання цієї лабораторної роботи.
4. Перейдіть до іншої ревізії і ознайомтесь з іншим прикладом.
5. Повторіть п.п.3-4 для кожної ревізії в гілці прикладів до лекції.
6. Створіть новий (віддалений або локальний) репозиторій GitHub для збереження коду наступних лабораторних робіт практикуму з дисципліни, передбачивши для кожної роботи окрему гілку.
7. У випадку створення віддаленого репозиторію GitHub визначте його URL-адресу та включіть її в звіт про виконання цієї лабораторної роботи.
8. У випадку створення локального репозиторію Git інформацію про це внесіть в звіт про виконання цієї лабораторної роботи.