

Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №1
з дисципліни «Декларативне програмування графічних
інтерфейсів»
«Основи роботи з Git»
Варіант №19

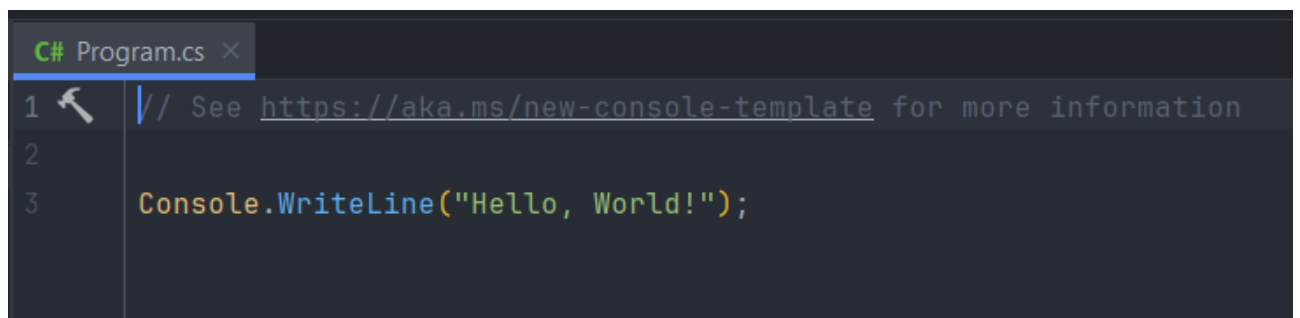
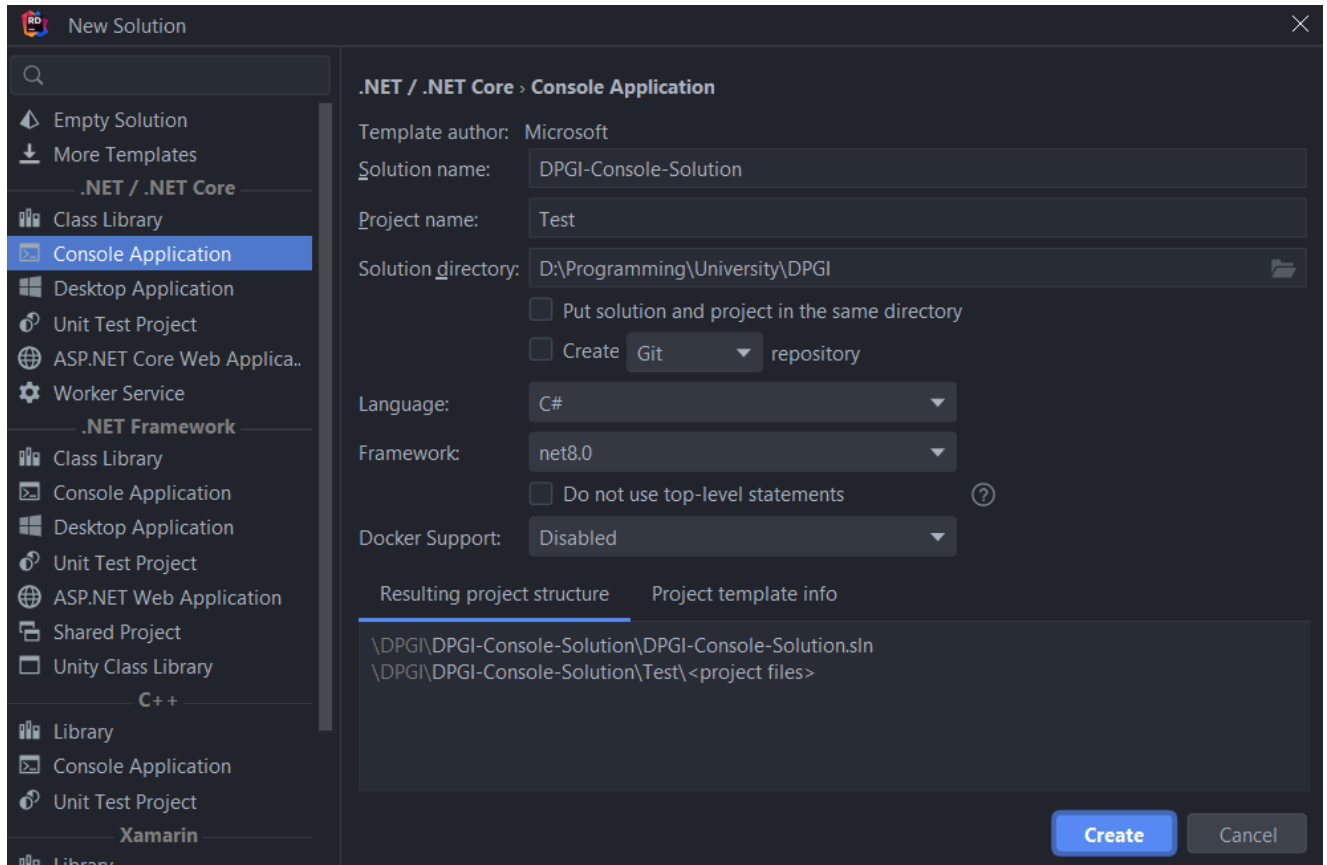
Виконав: Студент групи ТР-12
Ковальов Олександр
Перевірів: доцент, к.ф.-м.н.
Тарнавський Ю. А.

Мета роботи. Ознайомитись з можливостями JetBrains Rider по керуванню контролем версій за допомогою Git та GitHub.

Завдання 1. Опанувати технології Git для роботи з локальним репозиторієм.

Створіть у IDE новий проект за шаблоном консольного додатка. При цьому утворюється простий додаток, який виводить рядок тексту "Hello World!".

Створюємо:



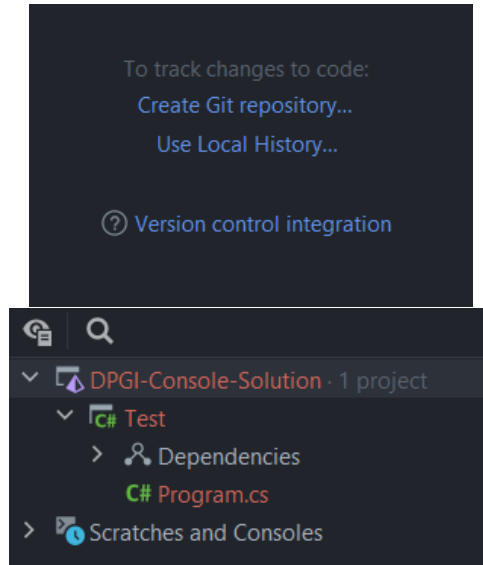
Перевірте правильність налаштування глобальних параметрів Git та в разі необхідності задайте ім'я користувача і його адресу електронної пошти.

Перевіримо ці дані в консольній утиліті Git Bash:

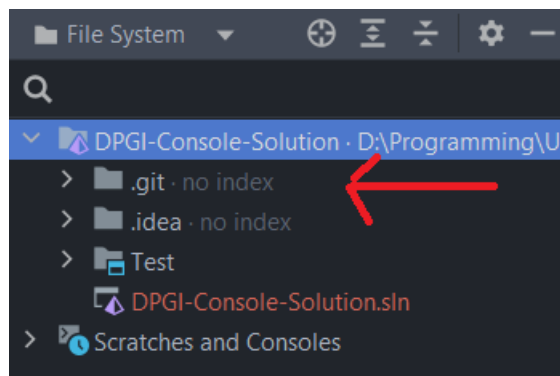
```
Alex@DESKTOP-RG3AG4Q MINGW64 ~
$ git config --list
user.name=xairaven
user.email=alexkovalievkpi@gmail.com
```

Створіть локальний репозиторій Git.

Створюємо за допомогою стандартних засобів IDE:

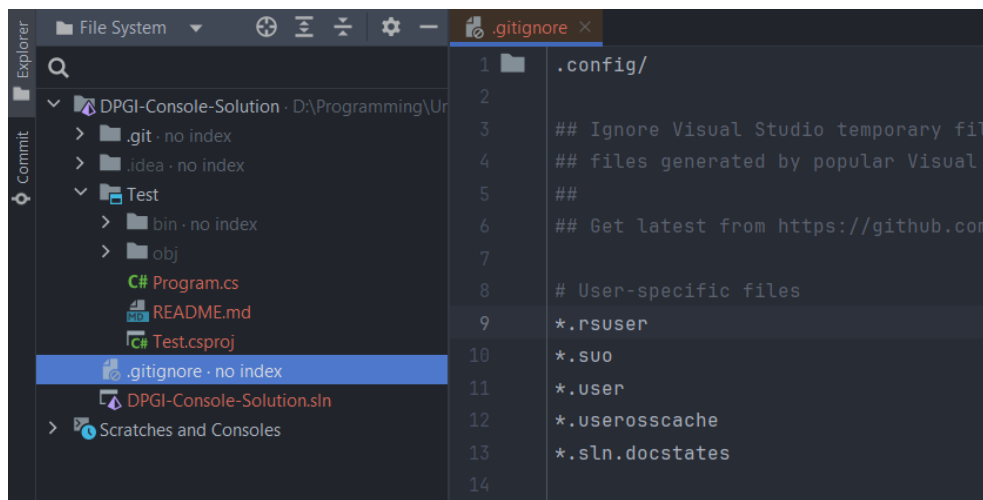


Відкрийте папку проекту і переконайтесь в наявності прихованої папки репозиторію .git.

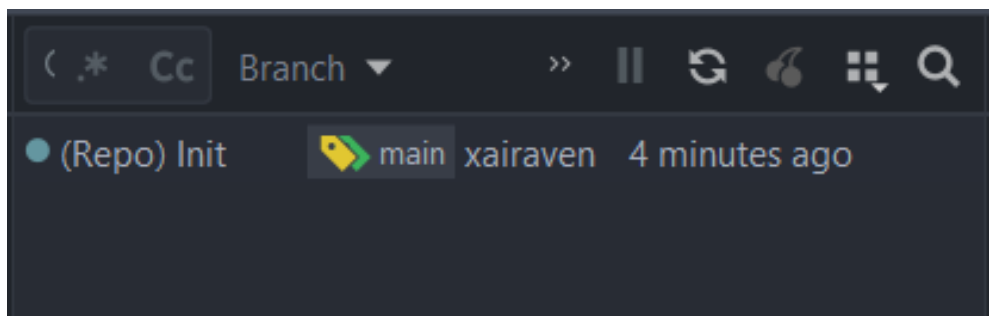
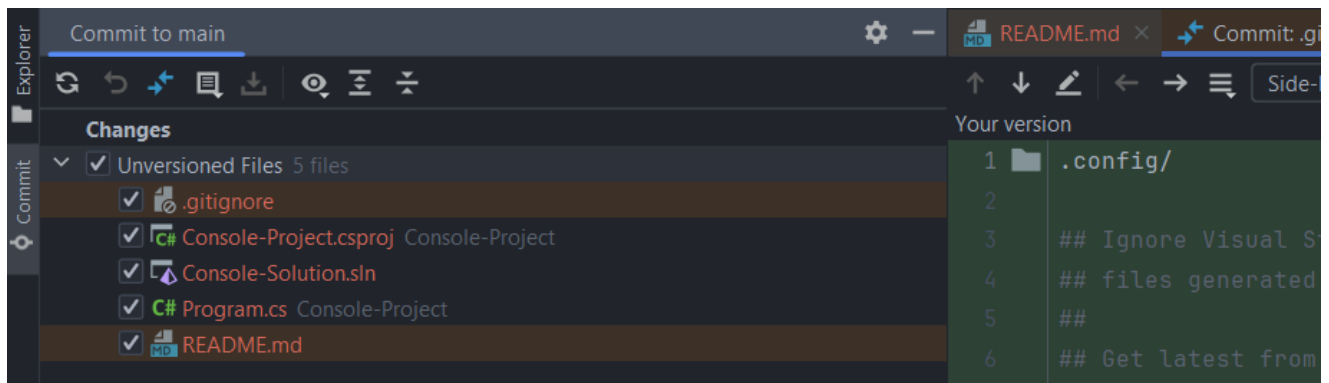
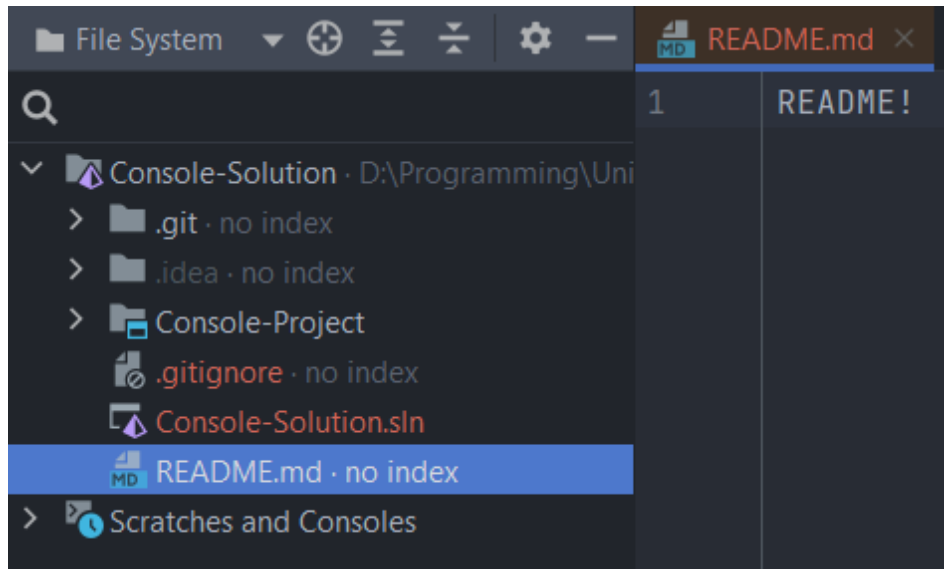


Відшукайте в папці проекту файл .gitignore, відкрийте його в текстовому редакторі і переконайтесь у правильності заданих налаштувань.

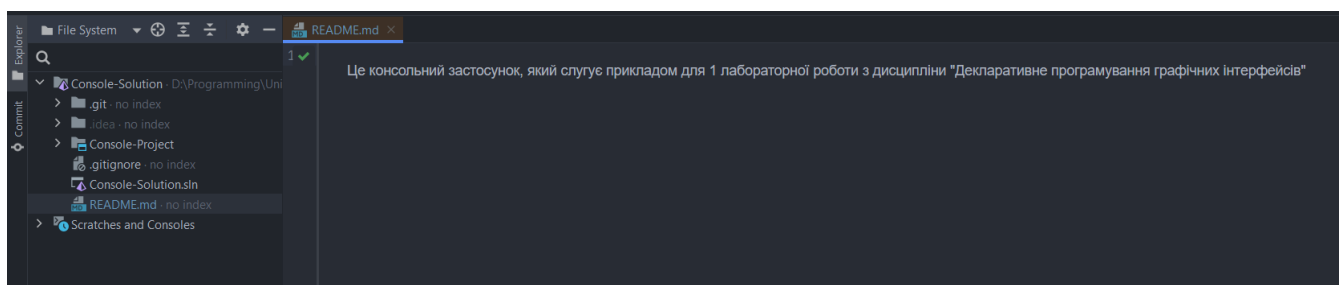
В JetBrains Rider його потрібно створювати самому. Створений файл:

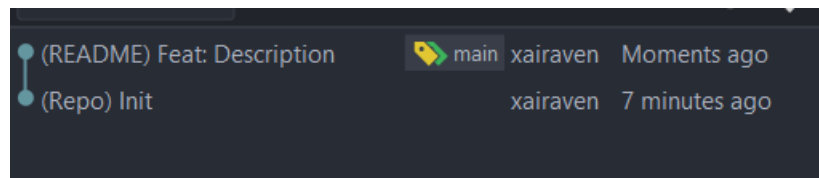


Додайте в проект текстовий файл readme та зафіксуйте цю зміну в автоматично створеній гілці main.



Додайте в файл readme контекстну інформацію про проект і також зафіксуйте цю зміну.

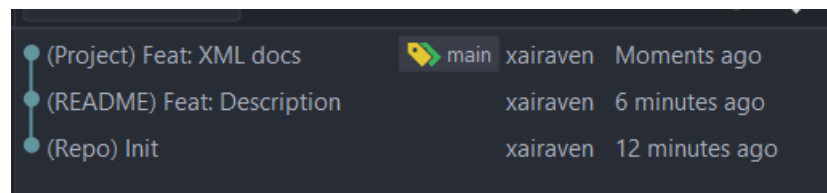




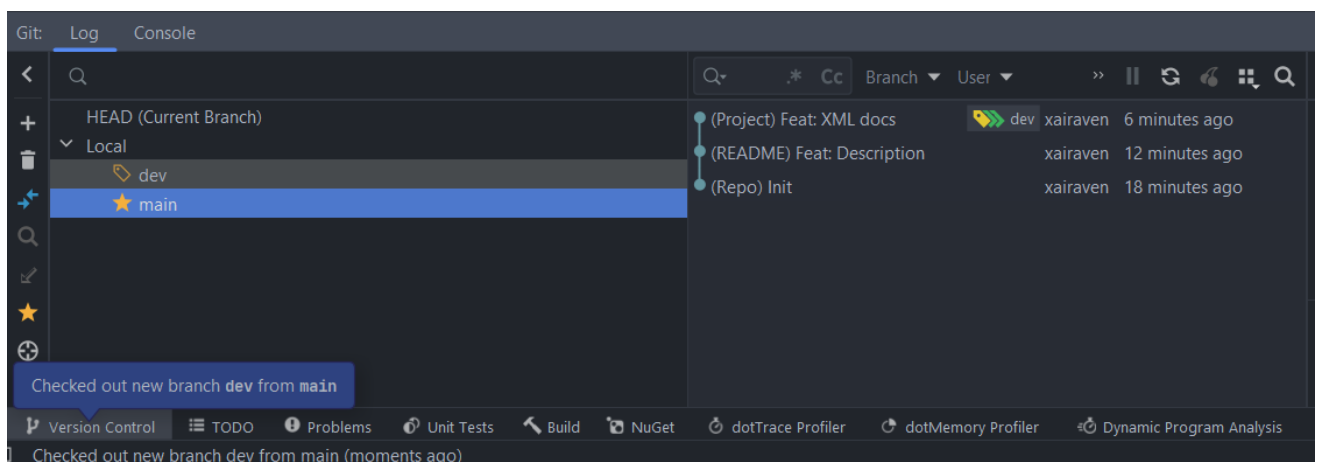
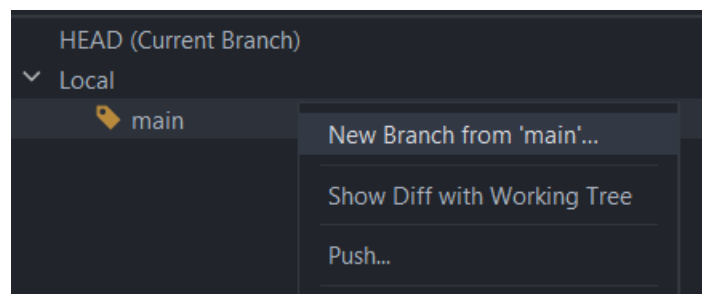
Додайте в код програми (Program.cs) коментарі для створення XML-документації і також зафіксуйте цю зміну.

```
2 {
3     /// <summary>
4     /// Entry point of a program.
5     /// </summary>
6     new *
7     public static void Main(string[] args)
8     {
```

Відкрийте журнал фіксацій і переконайтесь в наявності створених ревізій в гілці main:



Створіть нову гілку dev на основі гілки master.

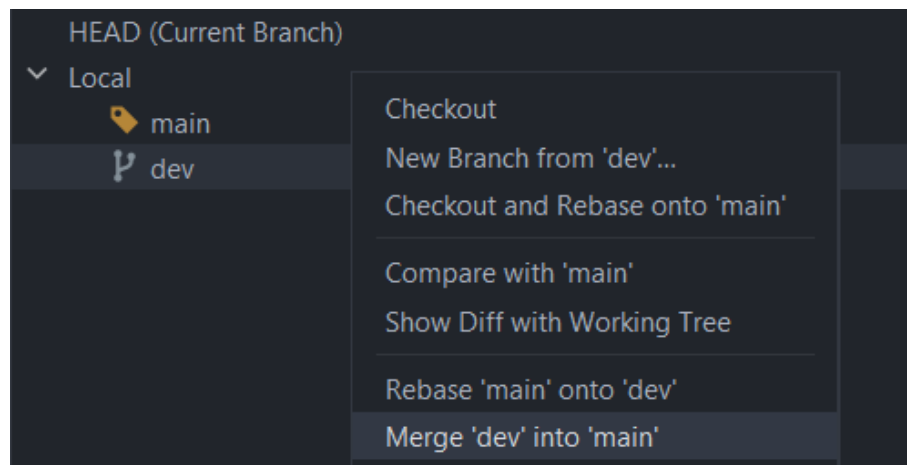


Модифікуйте код програми (Program.cs), створивши в ній клас Hello для виведення привітання і зафіксуйте цю зміну.

```
8      Console.WriteLine("Hello World!");
9
10     var hello = new Hello();
11     hello.Write();
12 }
13
14
15 public class Hello
16 {
17     public void Write()
18     {
19         Console.WriteLine("Hello!");
20     }
21 }
```

(Project) Feat: Hello Class	dev	xairaven	Moments ago
(Project) Feat: XML docs	main	xairaven	10 minutes ago
(README) Feat: Description		xairaven	16 minutes ago
(Repo) Init		xairaven	23 minutes ago

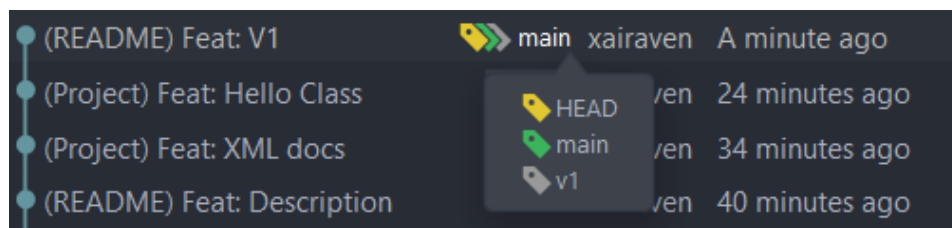
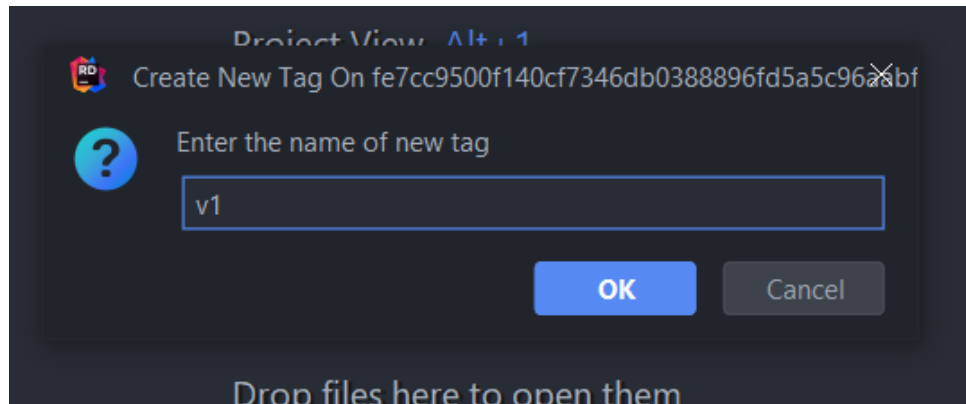
Об'єднайте гілку dev з гілкою main.



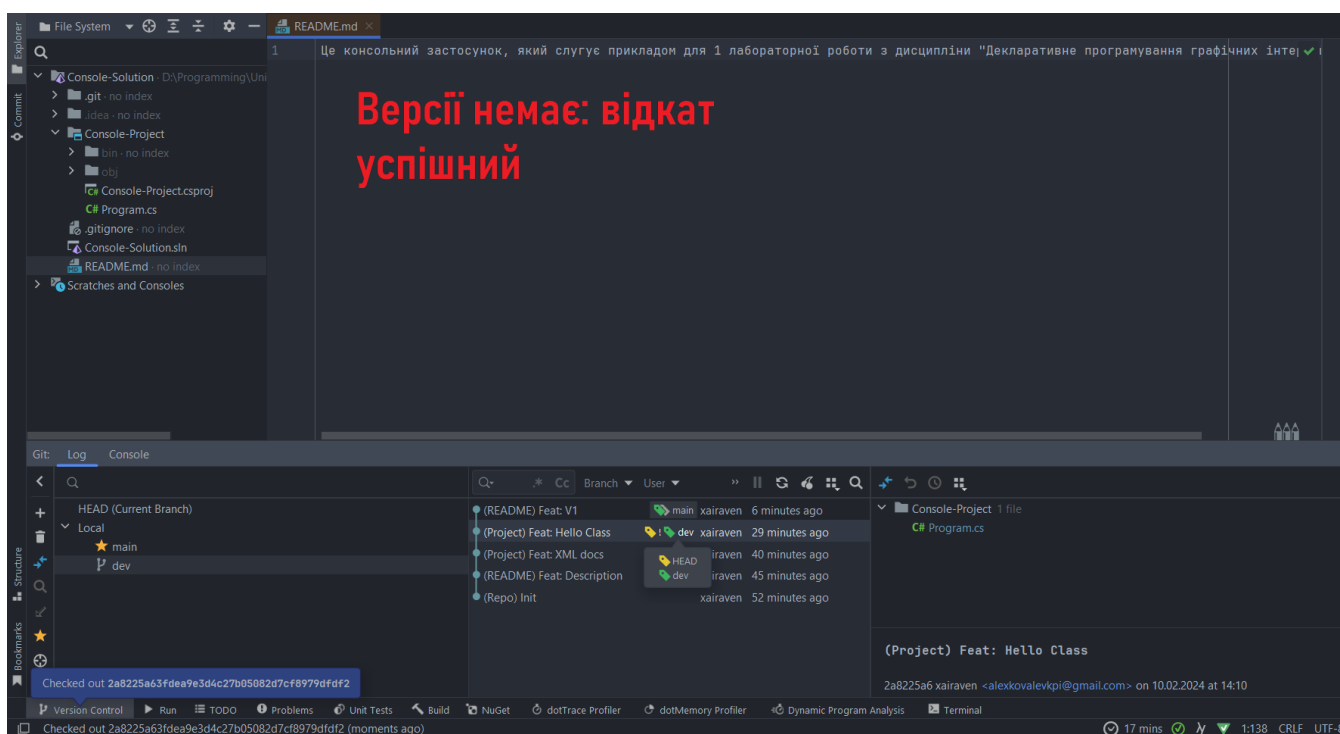
(Project) Feat: Hello Class	main	xairaven	7 minutes ago
(Project) Feat: XML docs		xairaven	17 minutes ago
(README) Feat: Description		xairaven	23 minutes ago
(Repo) Init		xairaven	30 minutes ago

Додайте в файл readme інформацію про дату створення версії №1 програми і також зафіксуйте цю зміну як першу версію програми.

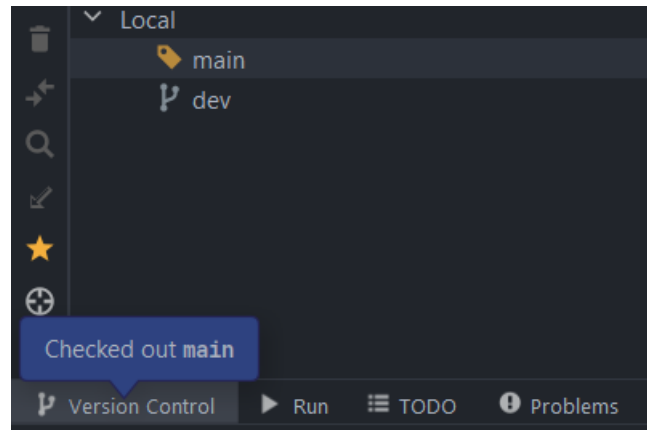
```
Версія №1: 10.02.2024 14:33
```



Перейдіть до однієї з попередніх ревізій і переконайтесь, що стан програми повністю відновлюється до стану фіксації. Задokumentуйте перехід до ревізії за допомогою скріншота в звіті про виконання цієї лабораторної роботи.

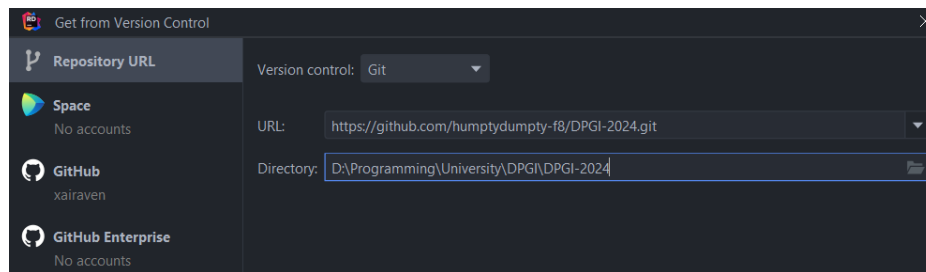
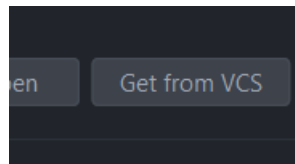


Повторіть п.п.14-15 для інших ревізій. Поверніться на гілку main.

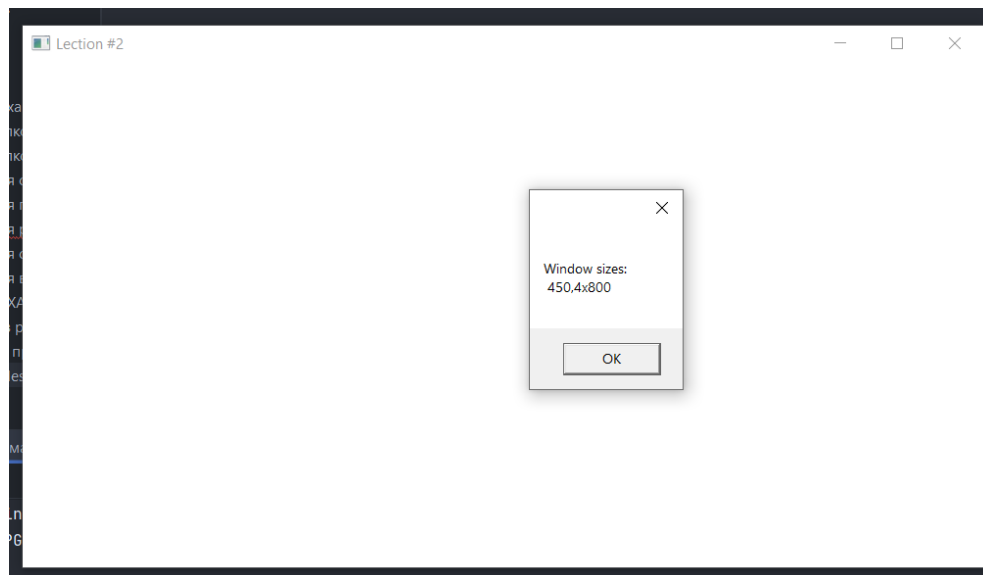


Завдання 2. Опанувати технології Git для роботи з віддаленим репозиторієм.

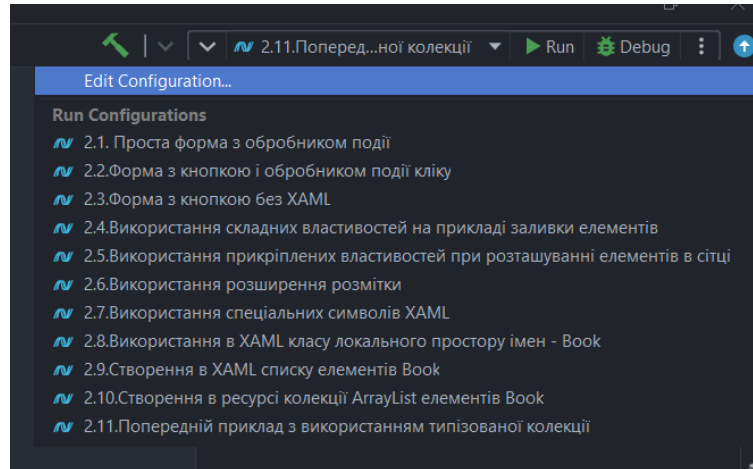
Отримайте у викладача URL-адресу репозиторію GitHub з прикладами до лекцій та виконайте його клонування.



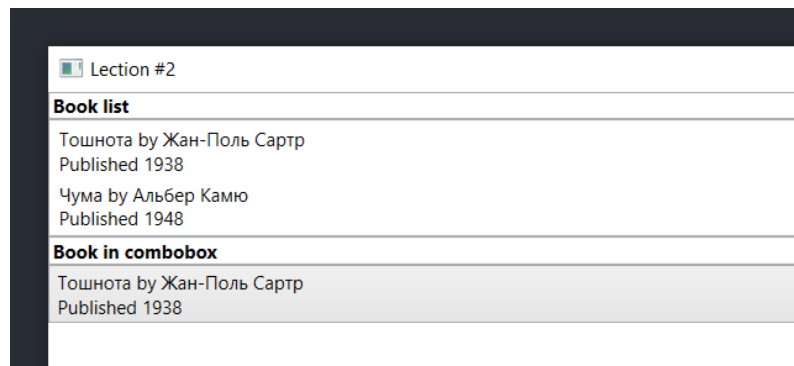
Виберіть одну з ревізій в гілці прикладів до лекції для переходу в стан демонстрації прикладу і запустіть проект на виконання.



Задokumentуйте перехід до ревізії за допомогою скріншота в звіті про виконання цієї лабораторної роботи.

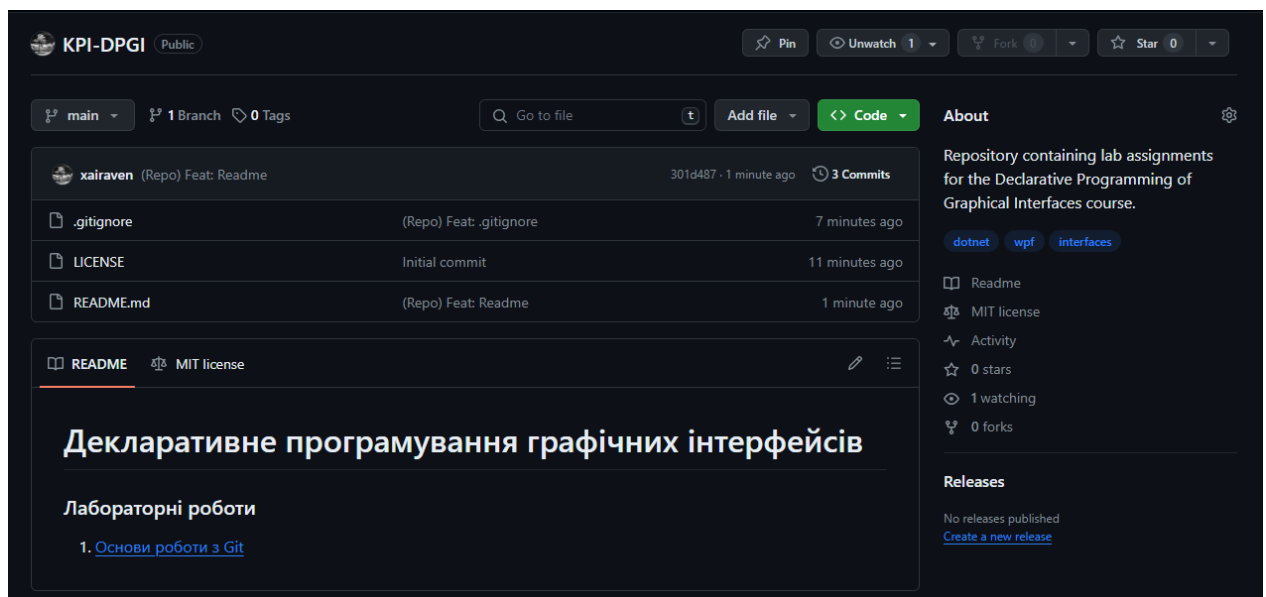


Перейдіть до іншої ревізії і ознайомтесь з іншим прикладом.



Повторіть п.п.3-4 для кожної ревізії в гілці прикладів до лекції.

Створіть новий (віддалений або локальний) репозиторій GitHub для збереження коду наступних лабораторних робіт практикуму з дисципліни, передбачивши для кожної роботи окрему гілку.



У випадку створення віддаленого репозиторію GitHub визначте його URL-адресу та включіть її в звіт про виконання цієї лабораторної роботи.

Посилання: <https://github.com/xairaven/KPI-DPGI>

Висновок: за результатами виконання цієї лабораторної роботи було ознайомлено з основами утиліти Git, та роботі з нею за допомогою вбудованих засобів IDE JetBrains Rider.