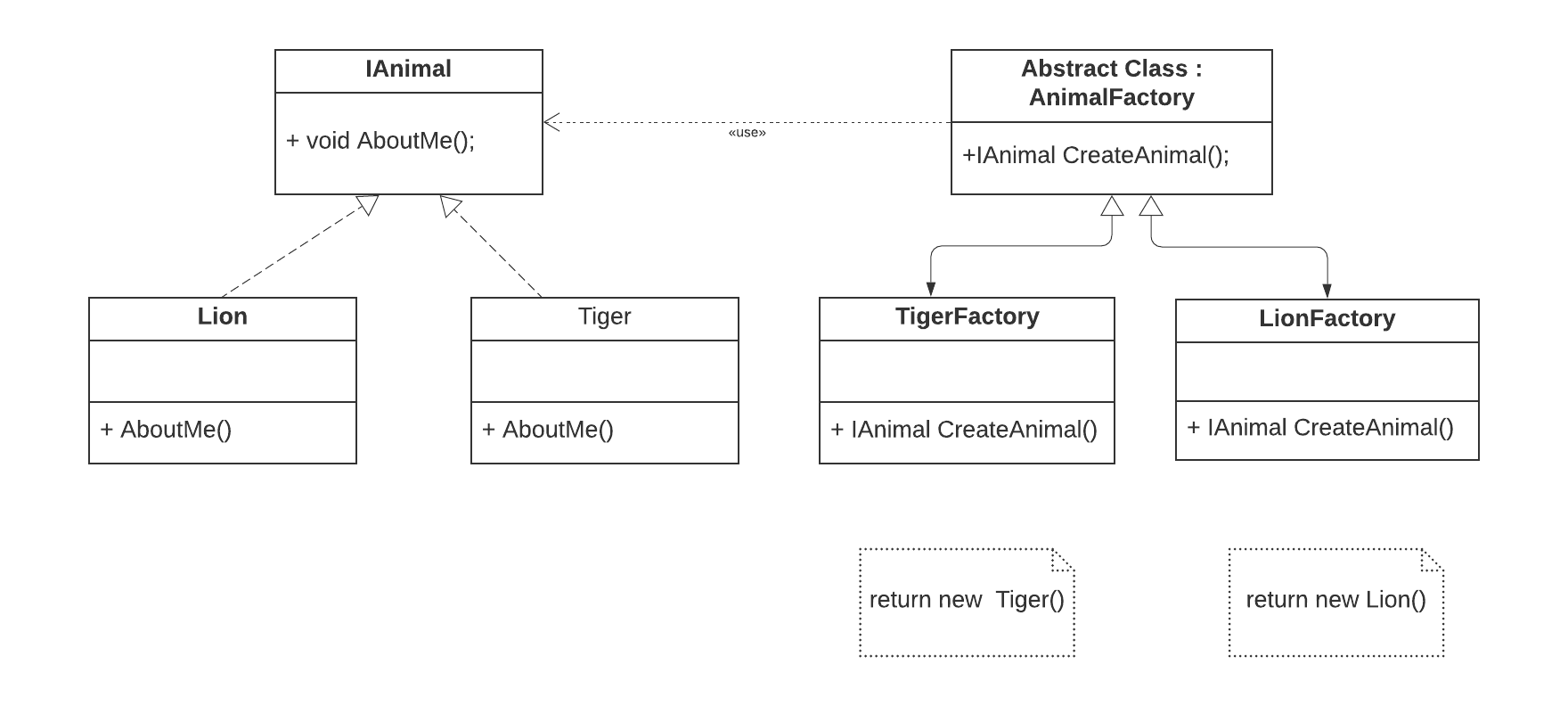
Họ và tên: Nguyễn Chi Tùng

MSV: He151309

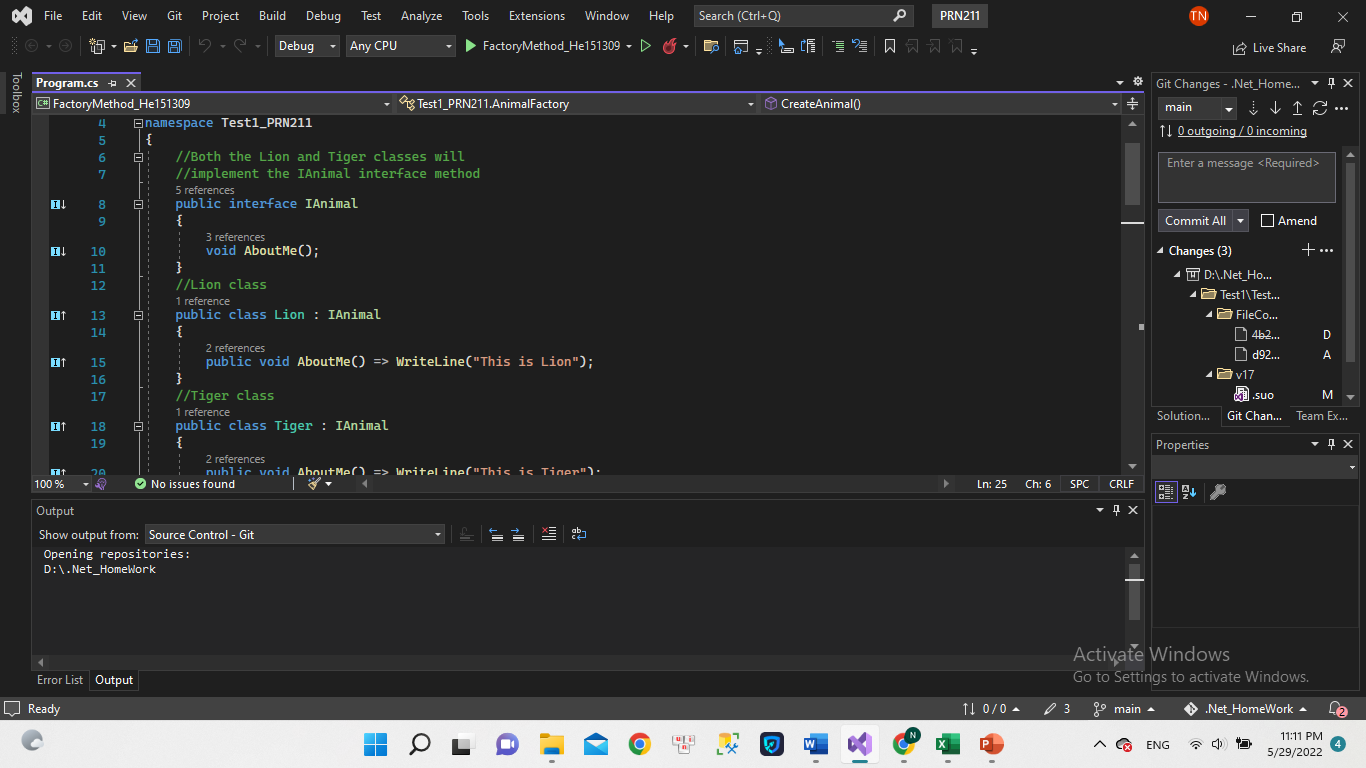
Môn: PRN211

**PROGRESS TEST (Factory Method)**

* We can see the general example through this diagram



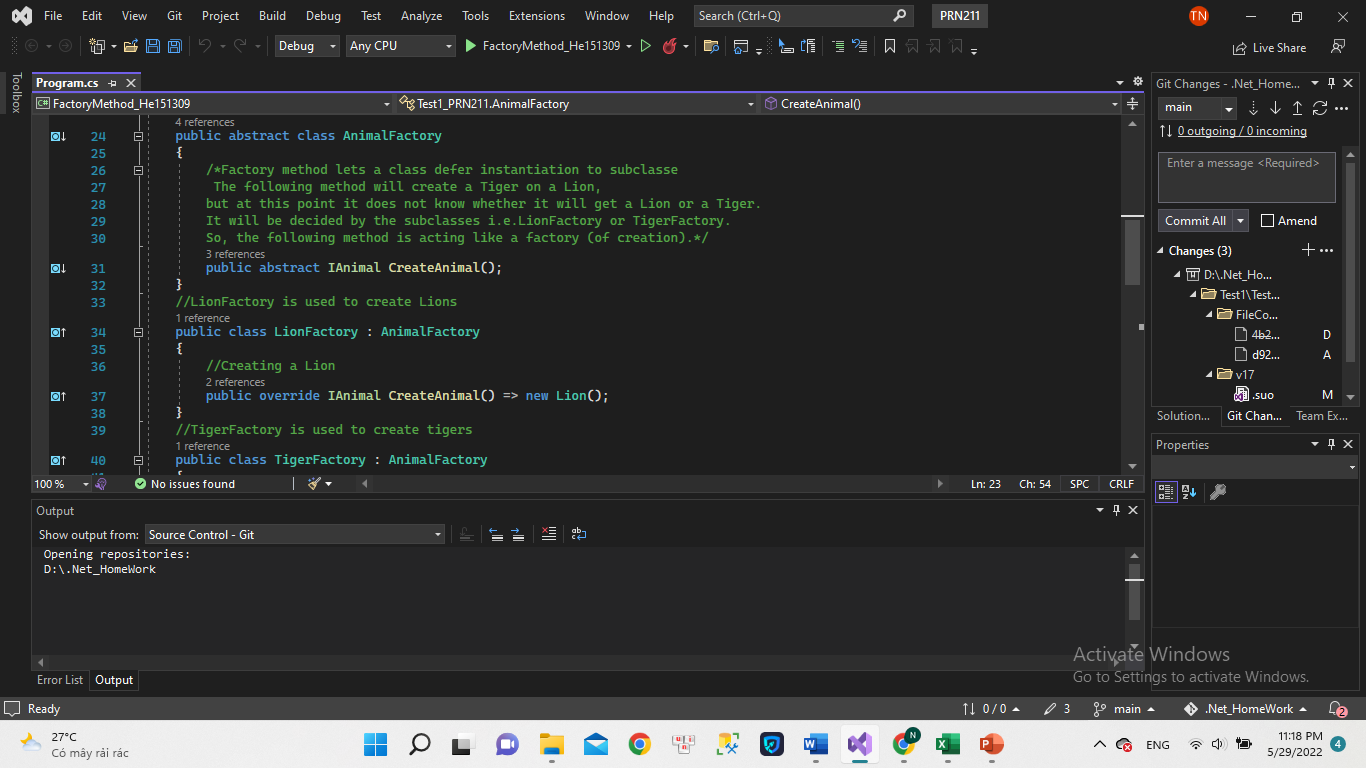
1. First we need to create an interface named IAnimal.



* This is an interface that contains only one non-returning function named AboutMe();
* Then we in turn instantiate 2 classes Tiger and Lion used to implement interface IAnimal -> Definition for function AboutMe() inside IAnimal class according to two different functions.

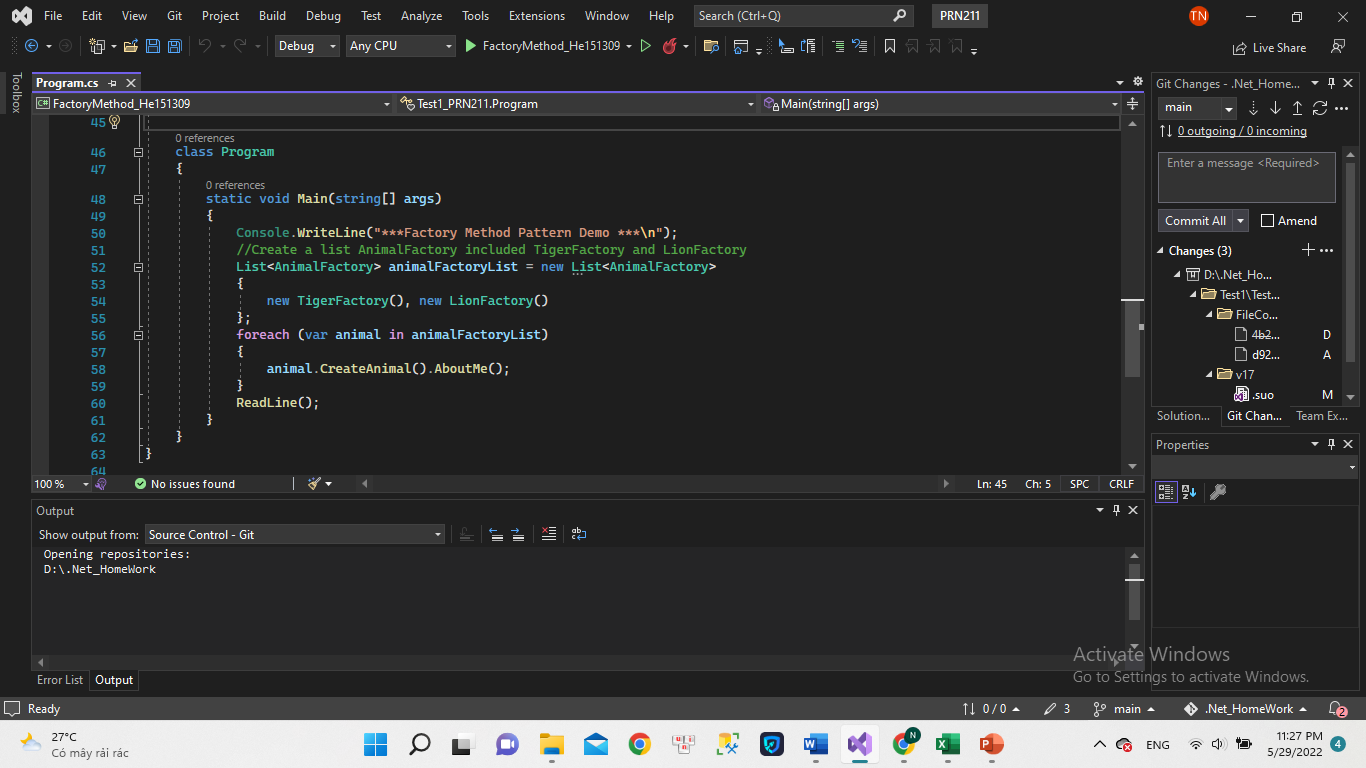
1. Next, we will create an abstract class named AnimalFactory. Inside it will contain an abstract function named CreateAnimal with a method that returns type IAnimal.

We will instantiate the two included TigerFactory and LionFactory classes. These two classes will inherit the AnimalFactory class and also have the task of implementing the CreateAnimal function in its parent class.



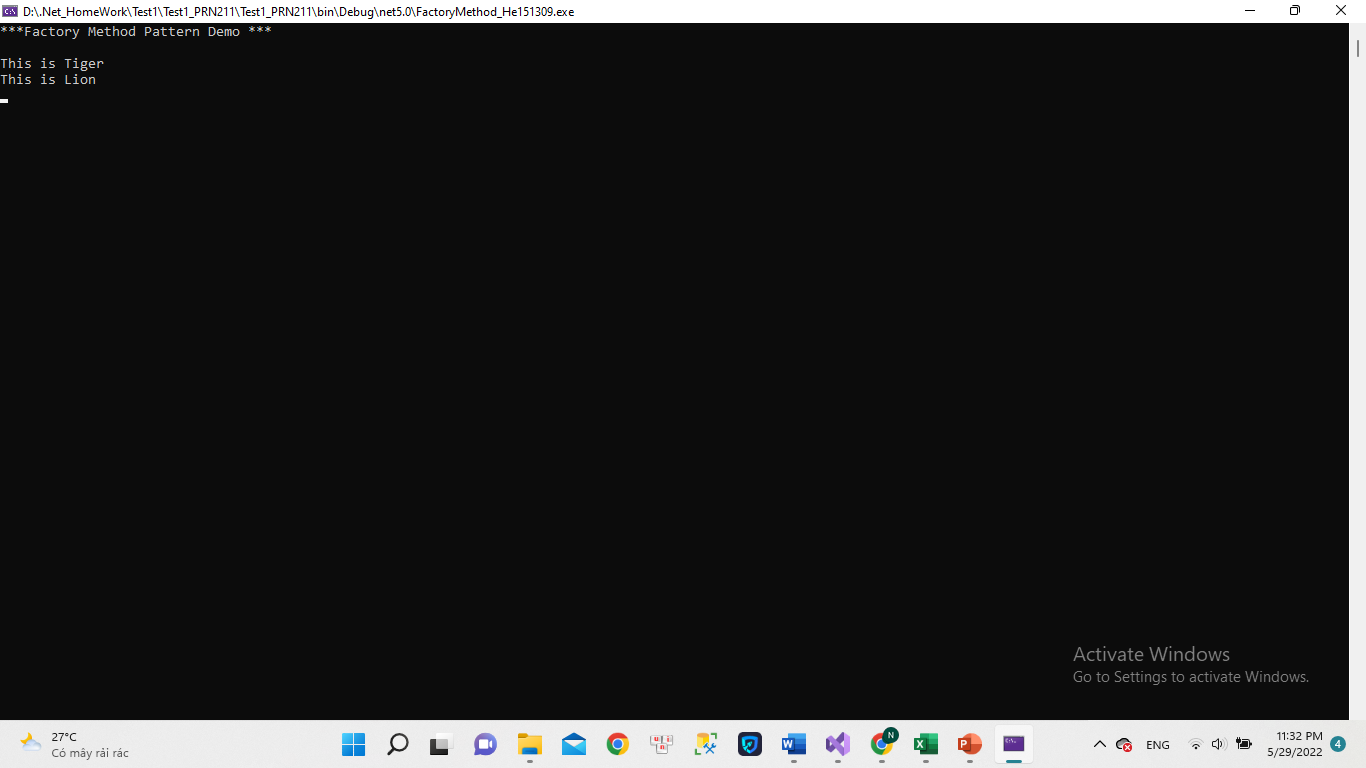
We can understand that Factory method lets a class defer instantiation to subclass the following method will create a Tiger on a Lion, but at this point it does not know whether it will get a Lion or a Tiger. It will be decided by the subclasses i.e.LionFactory or TigerFactory. So, the following method is acting like a factory (of creation).

1. Finally, run the test program in main function.



We will declare a list containing TigerFactory and LionFactory. To declare a Generic list, we must import the Generic library above. And inside we will call AboutMe() function to print the test case.

1. Result



We can see that the results are printed, but the images are the two statuses we put in the Lion and Tiger layers, respectively.