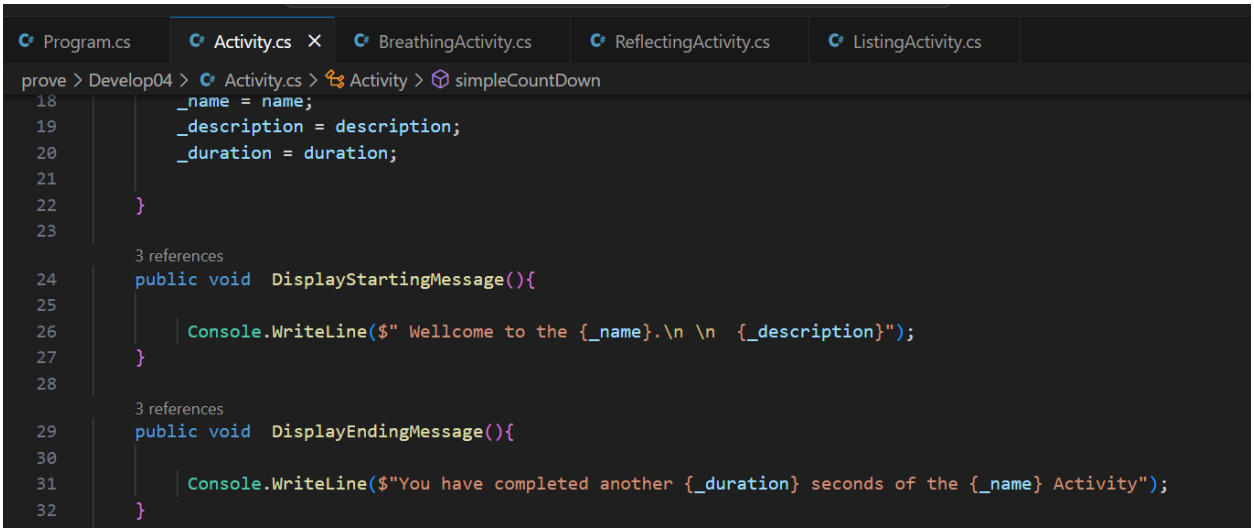
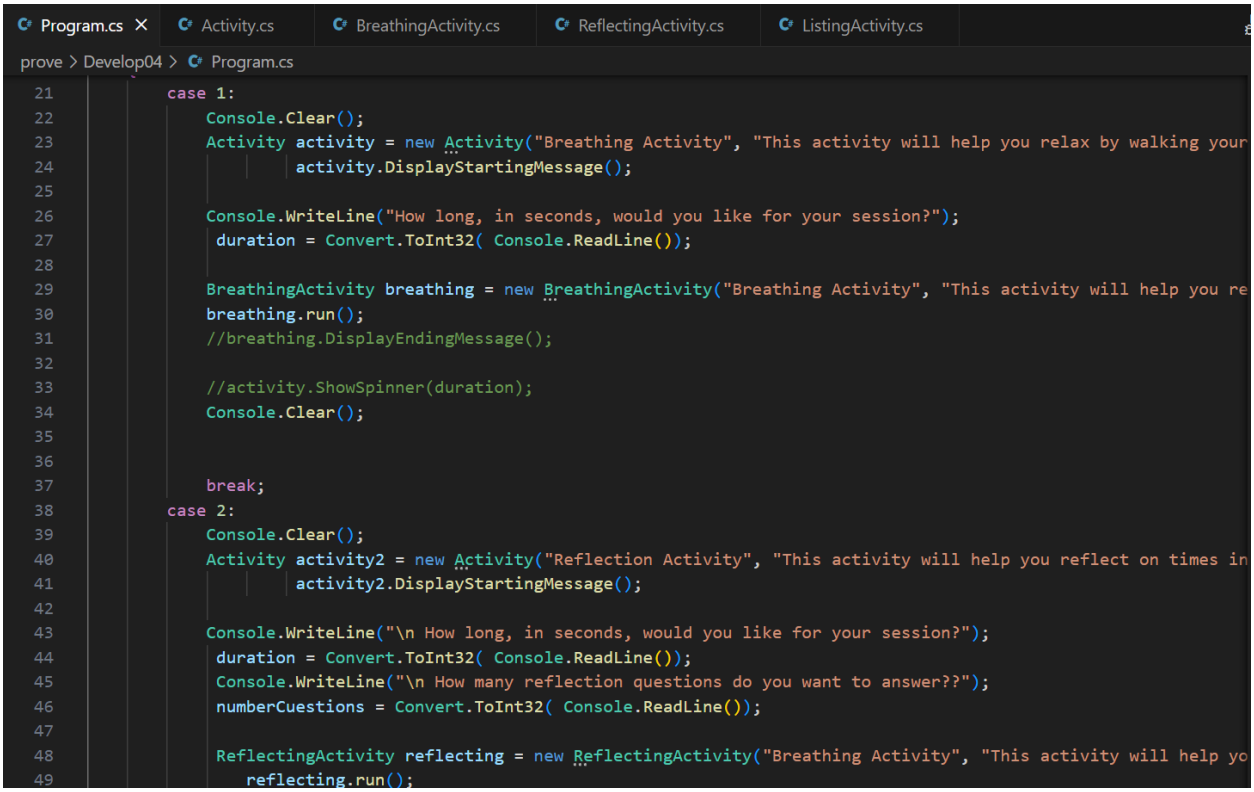
What is inheritance and why is it important?

- Explain the meaning of Inheritance
    - The basic concept of inheritance is to be able to obtain attributes or methods from other classes without having to write them.
- Highlight a benefit of Inheritance
    - In my own experience this week, one of the highlight benefits of inheritance is the reuse of code. We do not have to program the same code in each class, which will make the code more functional and easier to maintain.
- Provide an application of Inheritance
    - The best application I can give is the one I used this week with the Activity that had the function of displaying in the name of the activity for both Breathing, reflection and listing, the only thing I had to do is instantiate the Activity class along with its constructor To be able to do them, I didn't have to do them in every class.
- Use a code example of Inheritance from the program you wrote

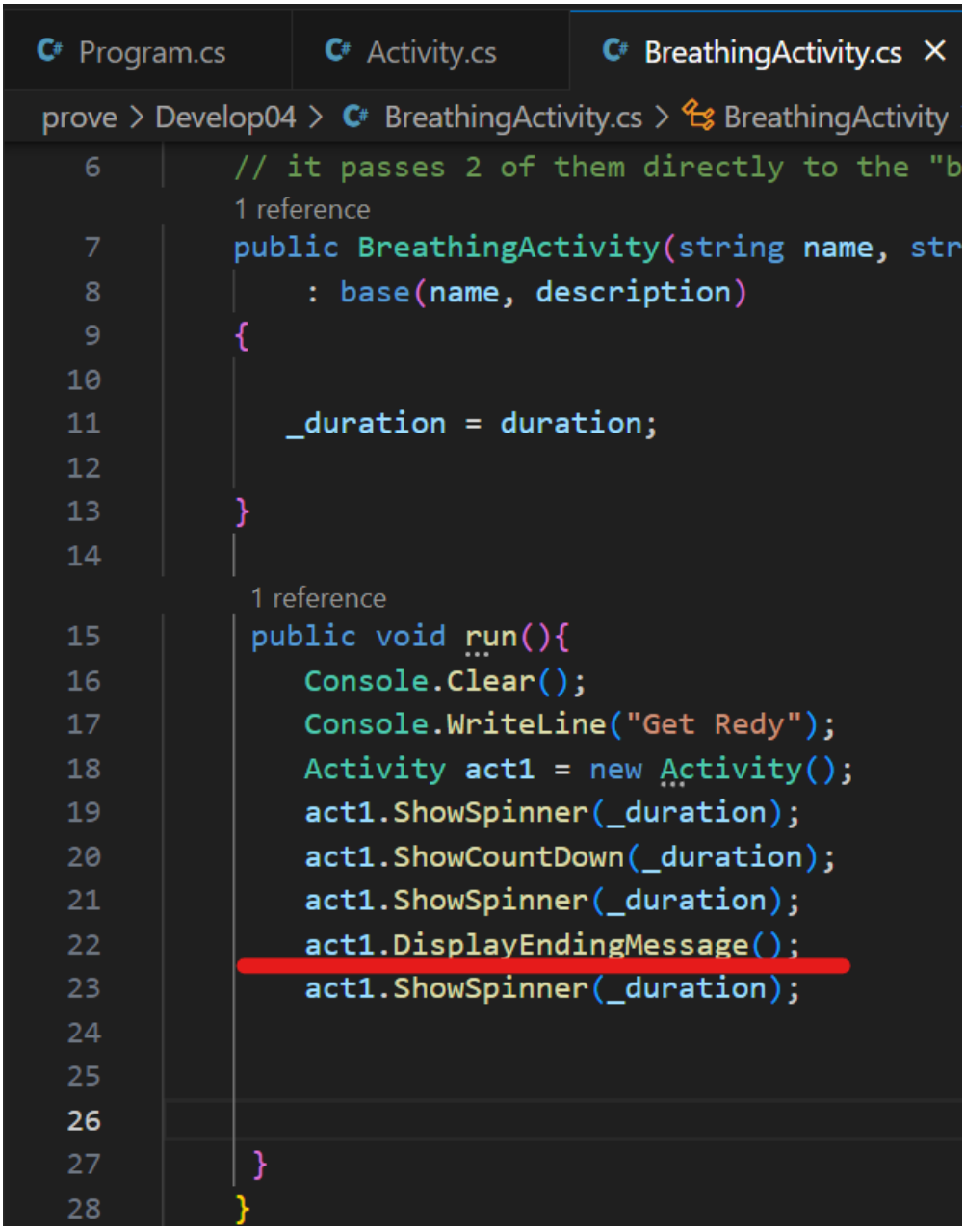
In the Activity class, the method was created to display the start and end message.



From here I make the instance of the class



from here I invoke it


Program.cs
Activity.cs
BreathingActivity.cs
prove > Develop04 > BreathingActivity.cs > BreathingActivity
// it passes 2 of them directly to the "b
1 reference
public BreathingActivity(string name, str
: base(name, description)
{
_duration = duration;
}
1 reference
public void run(){
Console.Clear();
Console.WriteLine("Get Redy");
Activity act1 = new Activity();
act1.ShowSpinner(_duration);
act1.ShowCountDown(_duration);
act1.ShowSpinner(_duration);
act1.DisplayEndingMessage();
act1.ShowSpinner(_duration);
}
}