SU - Assignment 1

Aditya Fadhillah (hjg708)

20. februar 2022

Report

Various concepts from C#

What is a class?
A class describes and assign all the attributes of objects, and it also designate the objects methods that implement the behavior of the object. So a class is what determine how an object behave and what kind of attributes the object hold. And a class achieve this by using fields and methods. As an example a bicycle class need to have two wheels, so one of its fields is wheel. And its method is to go on a solid ground, so a bicycle doesn't have a flying method.

What is an instance?
An instance is what you created when you declare an object of a class. So an instance is a specified member of a class that have been given values rather than variables. In a real life context might be that a student is a class, but I am an instance of that student class.

What is a field?
A field is a member of a class or an object, a field represents a memory location for storing a value. A field is used to define a variable in a class with its accessibility set to private. To access a field outside its class you need to use public method or property. The name of a student class is an example of a field.

What is a method?
A method is a member of a class that enable that class to do a specific task. A method comprises of returnType, methodName and method body. returnType specifies what kind of value that method return, so a string returnType returns a string value. methodName used as an identifier for that method. Method body contains the code for that specific task. toSpeak() can be a method to a student class.

What does it mean that a field or method is public or private?
Public: field or method is accessible by any other codes in the same class or even other classes that reference it.
Private: field or method is only accessible by codes in the same class.

Considerations

Getter and setter

In the first part of the DIKULecture assignment we needed to create a field name, at that tíme I did not create a getter and a setter for that name field. But in the next task we needed to make a public getter and a setter for that field name, this public method allows me to access the private field name. Because of this information I would automatically made a public getter and setter for my field just in case I need to access it. Because of the public getter and setter we can also make a class inherit another class by using (: base).

Code structure

I structure my code so that I put my field on the top of my class and then comes constructor, method and property. This helps me get an overview of my code and let me see whether or not I have made a getter and setter for my field.

Information method

The information method is located in lecture class ensures that both speaker and student class have access to it by declaring lecture.Information.

Text for method

In speaker and student class I added text to my method body to help understand what a certain method do, and that different circumstances yield different results. Like when a speaker want to speak they need to begin a lecture first to give their information or else a text will pop-up and that says "the speaker (Dorthe) cannot speak when not in a lecture". As seen from the response text the name Dorthe correspond to a speaker instance, this helps to see who was the person that did that action. I decide to add a text in the method body itself because I want to have it so I don't have to add text in the program class every time a different instance do something.

Broadcast method

The broadcast method is a bit confusing for me but in the end I decide so broadcast method need string variable as an input. Information field uses this string as its value. So the broadcast method sets the value of information field. For the speaker to give this information, they need to call the speak method. This method don't have any input but it outputs the string that was inputted in the broadcast method. So if the broadcast method doesn't receive any input or that the speak method is called alone, the speaker will simply return an empty string. And of course both of these methods check whether or not the speaker is in a lecture or not, before they store or give any information.

Yellow error message

There are two yellow error messages on my code, both of them reference to a non-nullable field 'lecture' must contain a non-null value when exiting constructor. Consider declaring the field as nullable. This error occurs because the code like it when a non-nullable field have null as its value, which is what happened here. I want it so that in the beginning speaker and student class is not in any lecture, in other word null. So lecture need to have the value null at some point so that my isInLecture can return false when they are not in a lecture.

Access the information of person instances

I have implemented an override string method on the person class to give me the information of a student and a speaker, by making this method on person class it automatically applies to all student and speaker instances. This method is not required in the assignment, but I want to familiarize myself with the override String method.