UML Class Diagram

https://csci-1301.github.io/about#authors

August 11, 2021 (09:36:30 AM)

Contents

Unified Modeling Language (UML) is a tool for visually representing programs. UML can represent many different types of diagrams. In this lab you will practice interpreting and creating one of them: a *class diagram*.

1 Interpreting a UML class diagram

1.1 Reading the diagram

Study the following diagram, then answer follow up questions:

```
|=========================================|
|                 Account                 |
|-----------------------------------------|
| - balance : decimal                     |
|-----------------------------------------|
| + GetBalance():decimal                  |
| + DisplayBalance():void                 |
| + AddFunds(amount:decimal):void         |
| + Withdraw(amount:decimal):void         |
|=========================================|
```

1. What is the name of this class?
2. How many attributes does this class have?
3. What is the data type of `balance`?
4. How many methods does this class have?
5. What is the significance of + and – in the diagram?
6. You will notice that there are two similar methods: `GetBalance` and `DisplayBalance`
    - based on the name can you interpret the behavior of these methods?
    - can you think of *why* we might need two such similar methods?

1.2 Implementing the class

Class diagram provides a concise way to represent attributes and methods, but it does not explain the implementation of the methods.

Knowing that:

1. `GetBalance` returns the current value of balance,
2. `DisplayBalance` displays the current balance at the screen formatted as currency,

   for example:

   ```
   Your current balance is $1,000,000.00 dollars!
   ```

3. `AddFunds` increments the current balance value by specified `amount`, and
4. `Withdraw` reduces balance by specified `amount`.

implement your version of this class in C#. For completeness, after you are done you should instantiate an object of the class and ensure it works as described.

2 Creating your own class diagram

In this next exercise you will practice drawing your own diagram, on paper.

1. Draw the UML class diagram of a `PreciseRectangle` class.
2. It should have two attributes: `width` and `length` of type `double`
3. It should have eight methods:
   - two setters, two getters (i.e., one for each attribute)
   - `ComputeArea` method to compute the area of a precise rectangle
   - `ComputePerimeter` method to compute the perimeter of a precise rectangle
   - `Swap` method to swap the length and the width of a precise rectangle
   - `Multiply` method to multiply the length *and* width of a precise rectangle by an ratio given in argument as an integer (`int`)

3 Pushing Further (Optional)

The following is an independent task, to widen your understanding of UML modelling concepts:

1. Class diagrams are just a special case of UML diagram. Have a look at https://en.wikipedia.org/wiki/Unified_Modeling_Language#Diagrams. In which category are class diagrams: behavior, or structure diagram?

2. Besides modelling attributes and methods, class diagrams can also represent relationships between classes. Have a look at https://en.wikipedia.org/wiki/Class_diagram for more examples of class diagrams and its uses.
3. Activity Diagram is another type of UML diagram for representing program actions. You will occasionally see activity diagrams in the lecture notes. Have a look at https://en.wikipedia.org/wiki/Activity_diagram and try to understand the example: “Activity diagram for a guided brainstorming process”.