Introduction to VB.net -OOP

Nasrin Khodapanah

VB.NET

Agenda

- OOP
  - Why?
  - Terms
- Exo

OOP

OOP

- ## Abstraction:

Normally, in a Class, variables used to hold data (like the radius of a cake) are declared as Private. Functions or property routines are used to access these variables. Protecting the data of an object from outside functions is called Abstraction or Data Hiding. This prevents accidental modification of data by functions outside the class.

- ## Encapsulation:

Putting all the data and related functions in a Class is called Encapsulation.

OOP

- ## Polymorphism:

Polymorphism provides for multiple implementations of the same method. For example, different objects can have a Save method, each of which performs different processing.

Polymorphism is the property in which a single object can take more than one form. For example, if you have a base class named Human, an object of Human type can be used to hold an object of any of its derived types. When you call a function in your object, the system will automatically determine the type of the object to call the appropriate function. For example, let us assume that you have a function named speak() in your base class. You derived a child class from your base class and overloaded the function speak(). Then, you create a child class object and assign it to a base class variable. Now, if you call the speak() function using the base class variable, the speak() function defined in your child class will work. On the contrary, if you are assigning an object of the base class to the base class variable, then the speak() function in the base class will work. This is achieved through runtime type identification of objects.

LINK

OOP

- Inheritance:

Reuse the interface of a class, but not its implementation. Visual Basic .NET provides for true implementation inheritance whereby you can reuse the implementation of a class.

Inheritance is the property in which, a derived class acquires the attributes of its base class. In simple terms, you can create or 'inherit' your own class (derived class), using an existing class (base class). You can use the Inherits keyword for this.

OOP

Why…

So far, we’ve only done Procedural programming.

(based on procedures and functions)

Complicated and hard when it comes to reusing the same code for something else.

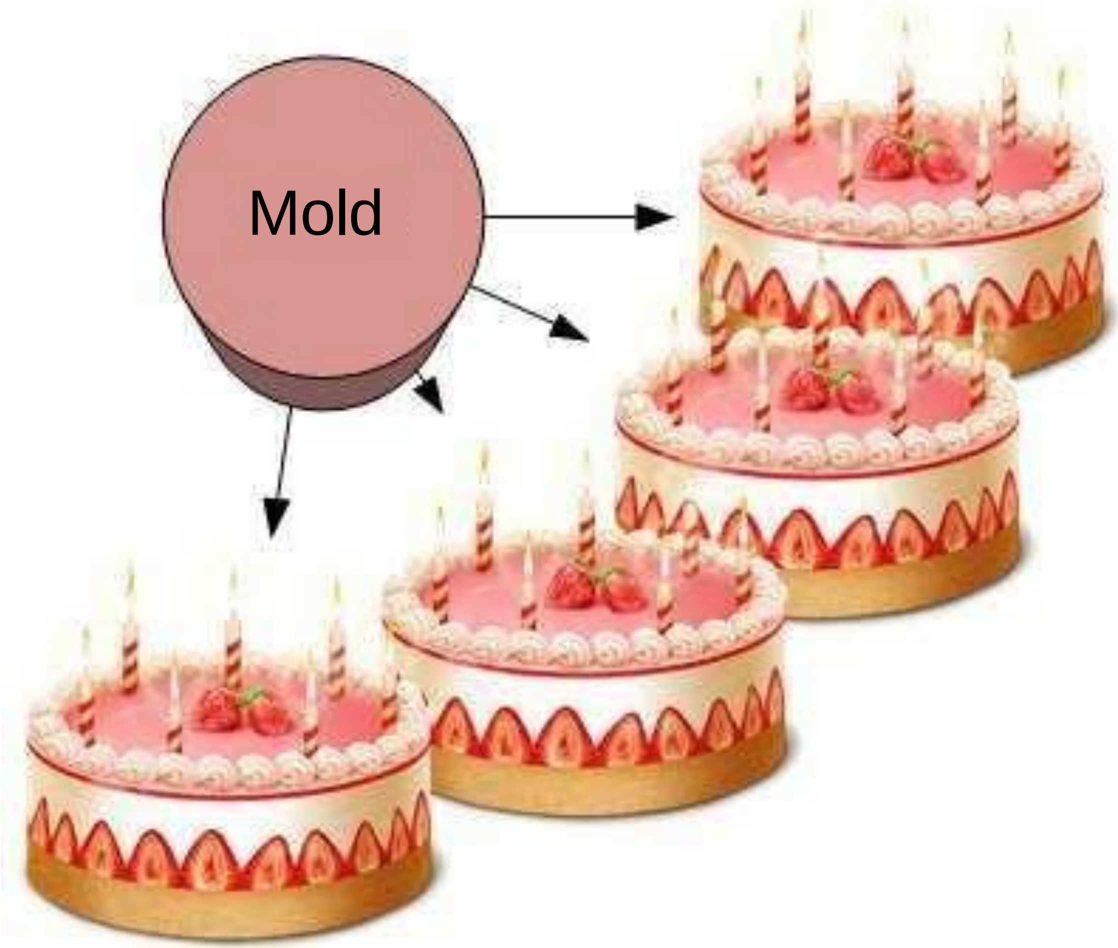
OOP

Objects

Grouping related functions and procedures

Ex: Cake

OOP



OOP

Access Type

The major access types are Public, Private, Friend and Protected. A Class may contain functions, variables etc., which can be either

Public or Private or Protected or Friend.

OOP

Access Type

If they are Public, they can be accessed by creating objects of the Class. Private and Protected members can be accessed only by the functions inside the Class. Protected members are much like Private members, but they have some special use while inheriting a Class.

OOP

Access Type

Friend members can be accessed only by elements of the same project, and not by the ones outside the current project.

OOP

Syntax:

```
Public Class MyClass
…
End Class
```

OOP

Constructor

This is the method that will be called when instantiating the object, when we do "= New MyClass“

The arguments are specify this way: "= New MyClass (consArg1, consArg2)".

Just like a function

OOP

Destructor

The destructor is particular, it is mainly used to release memory resources allocated to the object just before its destruction.

If a Class uses a database connection, the destructor will be used to close and release the connection

OOP

Destructor

In visual basic, Destructor is a special method of a class and it is useful in class to destroy the object or instances of classes. The destructor in Visual Basic will invoke automatically whenever the class instances become unreachable.

The following are the properties of destructor in a visual basic programming language.

- In visual basic, destructors can be used only in classes and a class can contain only one destructor.
- The destructor in class can be represented by using Finalize() method.
- The destructor in Visual Basic won't accept any parameters and access modifiers.
- The destructor will invoke automatically, whenever an instance of a class is no longer needed.
- The destructor is automatically invoked by the garbage collector whenever the class objects are no longer needed in an application.

useful link

OOP

Methods and Attribute

Methods are like functions private necessity is internal, nothing is visible from the outside.

When their attributes are public, visible from the outside accessible.

EXO 2