Unified Modeling Language

Agenda

- What is UML
- Why UML
- Use Case Diagram
- Class Diagram
- Class Relationships
- Multiplicity
- Examples
- Questions
- Resources

What is UML

- Provide users with an expressive visual modeling language.
- Provide extensibility and specialization mechanisms.
- Be independent of particular programming languages.
- Provide a formal basis for understanding
- Support higher-level development.
- Integrate best practices.

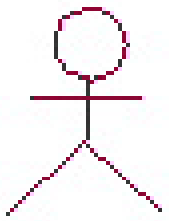
Why UML?

- Manage complex systems
- Visually convey object details and relationships
- Convey information is a structured and standard manner

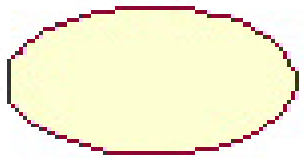
USE CASE

Use Case

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

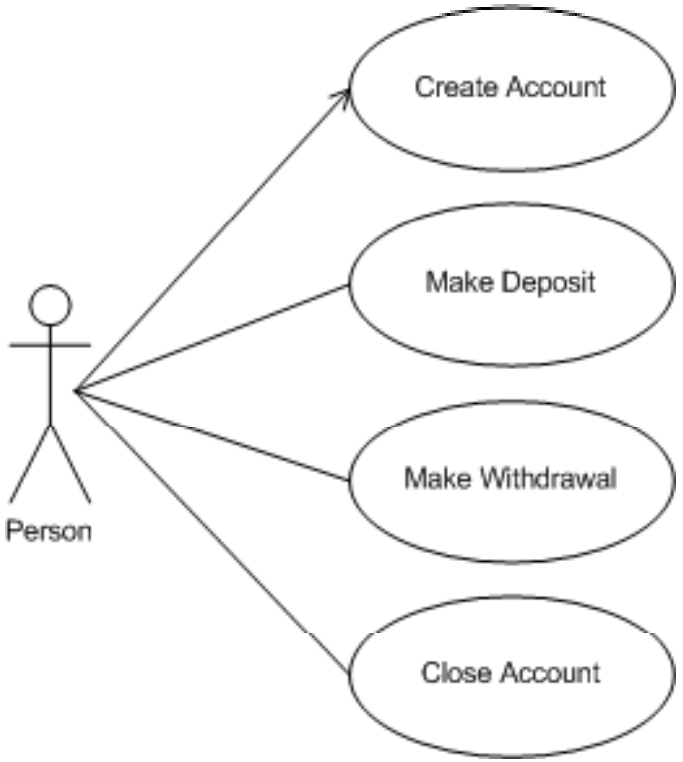


Actor



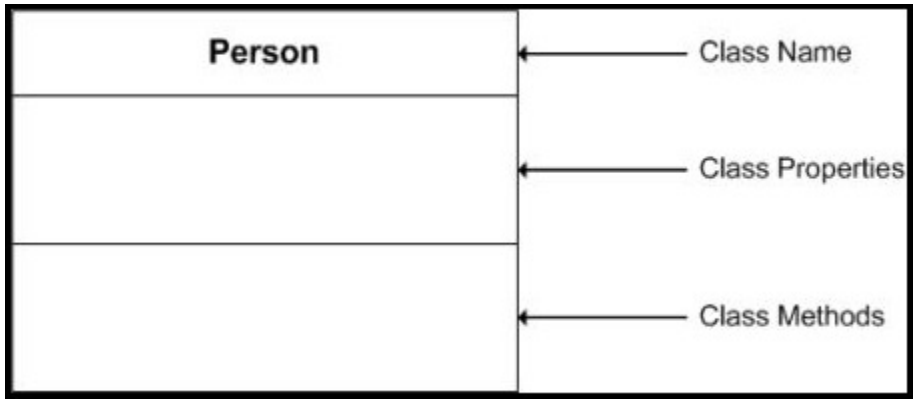
Use Case

Use Case

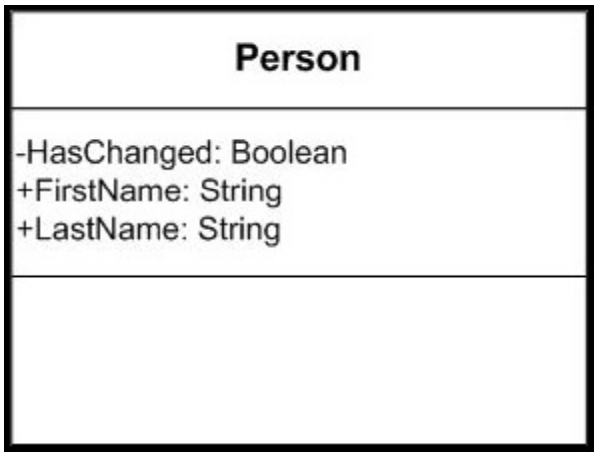

Create Account
Make Deposit
Make Withdrawal
Close Account
Person

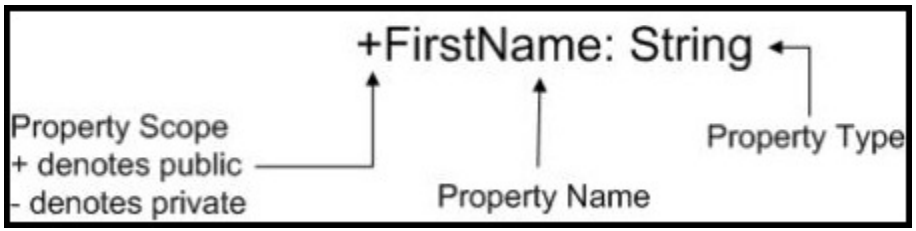
CLASS DIAGRAM

Class Diagram

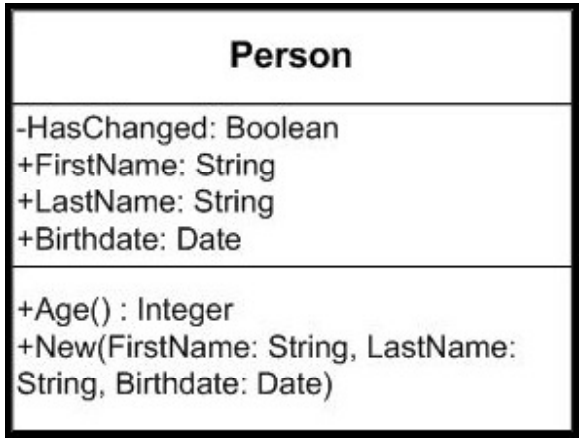


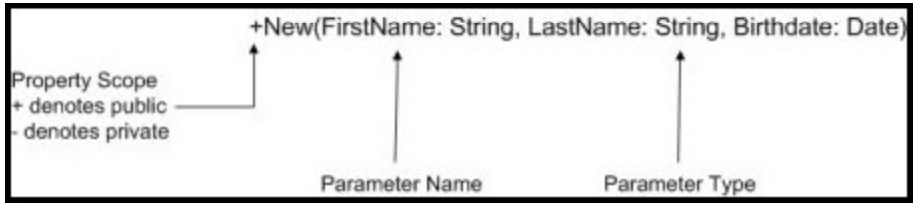
Class Diagram





Class Diagram





Class Diagram

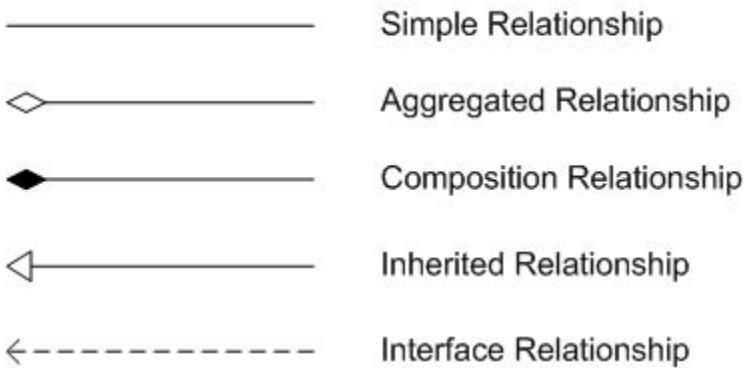
```
Public Class Person

    Private _HadChanged As Boolean

    Private _firstName As String
    Public Property FirstName() As String
        Get
            Return _firstName
        End Get
        Set(ByVal Value As String)
            _firstName = Value
        End Set
    End Property

    Private _lastName As String
    Public Property LastName() As String
        Get
            Return _lastName
        End Get
        Set(ByVal Value As String)
            _lastName = Value
        End Set
    End Property

    Private _birthDate As Date
    Public Property BirthDate() As Date
        Get
            Return _birthDate
        End Get
        Set(ByVal Value As Date)
            _birthDate – Value
        End Set
    End Property

    Public Function Age() As Integer

    End Function

    Public Sub New(ByVal FirstName As String, ByVal LastName As String, ByVal Birthdate As Date)

    End Sub

End Class
```

Class Diagram-Relationships

UML Relationships define the structure that exists between classes

Class Diagram-Relationships

There are five major relationship indicators



Class Diagram-Relationships

Simple Relationship


```
Public Class Man

    Private _spouse As Woman
    Public Property Spouse() As Woman
        Get
            Return _spouse
        End Get
        Set(ByVal Value As Woman)
            _spouse = Value
        End Set
    End Property

End Class
```

```
Public Class Woman

    Private _spouse As Man
    Public Property Spouse() As Man
        Get
            Return _spouse
        End Get
        Set(ByVal Value As Man)
            _spouse = Value
        End Set
    End Property

End Class
```

Class Diagram-Relationships

Aggregated Relationship


```
Public Class Car

    Private _WheelType As Wheel

End Class

Public Class Wheel

End Class
```

Class Diagram-Relationships

Composite Relationship


```
Public Class Company

    Private _Departments() As Department

    Private Class Department

    End Class

End Class
```

Class Diagram-Relationships

Inherited Relationship

Public Class BankAccount

End Class

Public Class CheckingAccount
    Inherits BankAccount

End Class

Class Diagram-Relationships

Interface Relationship

Public Interface Person

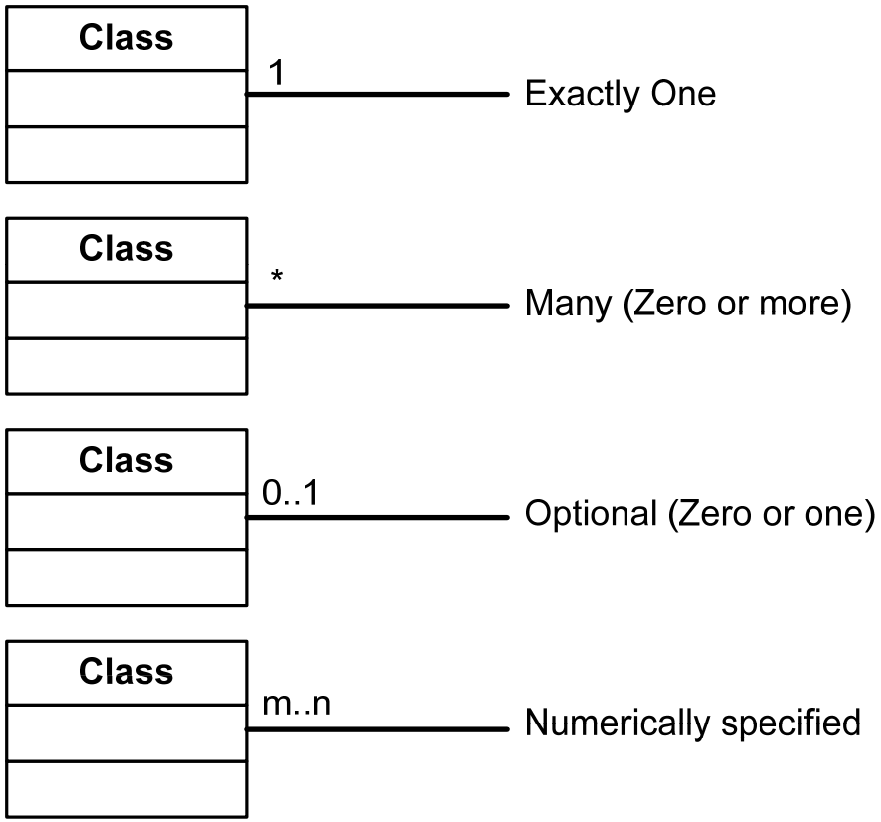
End Interface

Public Class Student
    Implements Person

End Class

Multiplicities

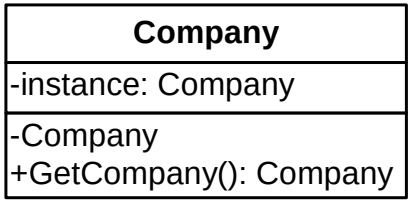
Class Diagram - Multiplicities



Examples

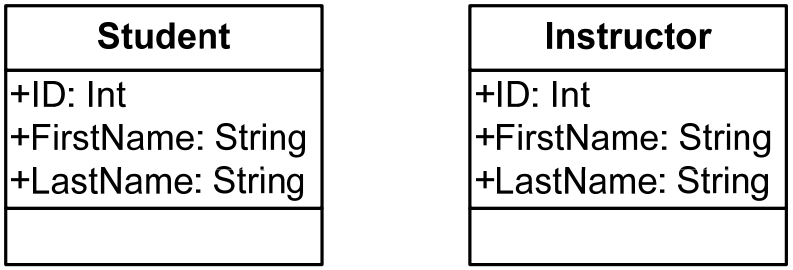
Examples

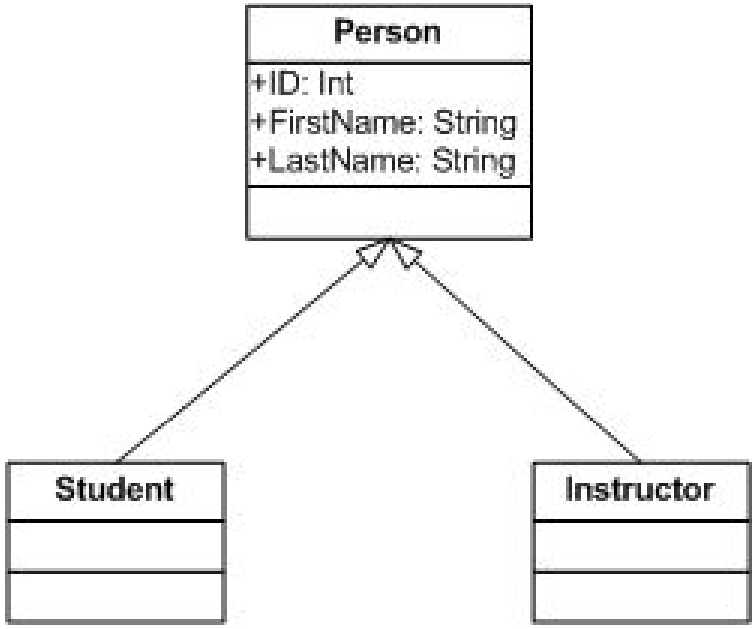
Singleton Pattern



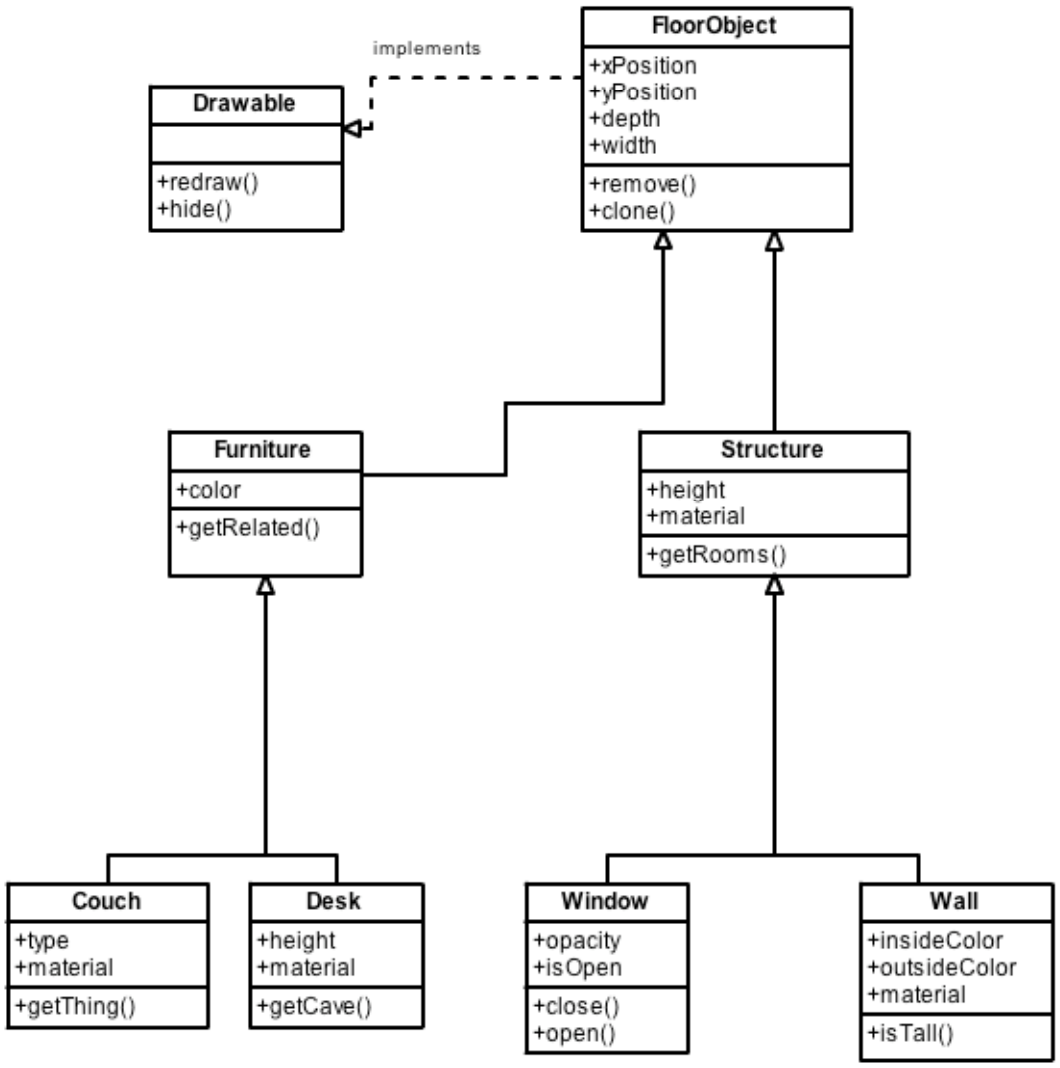

```
Public Class Company

    Private _Company As Company

    Private Sub New()
    End Sub

    Public Function GetCompany() As Company
        Return _Company
    End Function

End Class
```

Examples


Student
+ID: Int
+FirstName: String
+LastName: String
Instructor
+ID: Int
+FirstName: String
+LastName: String


Person
+ID: Int
+FirstName: String
+LastName: String
Student
Instructor

Examples


implements
FloorObject
+xPosition
+yPosition
+depth
+width
+remove()
+clone()
Drawable
+redraw()
+hide()
Furniture
+color
+getRelated()
Structure
+height
+material
+getRooms()
Couch
+type
+material
+getThing()
Desk
+height
+material
+getCave()
Window
+opacity
+isOpen
+close()
+open()
Wall
+insideColor
+outsideColor
+material
+isTall()

Questions

Resources

Tutorials
http://edn.embarcadero.com/article/31863
http://uml-tutorials.trireme.com/
http://thesoftwaredesigners.com/blog

UML Stencil Add-on for Visio
http://www.softwarestencils.com/uml/index.html

Free and Commercial UML Tools
http://en.wikipedia.org/wiki/List_of_UML_tools