UML Overview

Unified Modeling Language
UML – symbology for diagrams

- A note or comment
- Visibility:
  + public
  - private
  # protected
  ~ protected
- Stereotypes:
  <<abstract>>
  <<singleton>>
  <<interface>>
UML – symbology for static (class) diagrams

Class Name
-------------
instance attributes
Class Attributes
-------------
object methods
Class Methods

myObjectName:Class

Note: class method or Attributes are underlined. An instance of a class is written as the instance name : the class name and underlined.

Note: two different ways to show a class

inherits from (points to parent)

implements this interface

composition (exclusively “owns it”)
• contains it
• the diamond on the owner
• shows the multiplicity (0 or more)

aggregation
• references it
• knows about it, doesn’t own it
• the diamond is on the owner
• shows multiplicity (1 to 1 or more)

Points to an **Instance** of a class
Vehicle
-----------
theSteeringWheel: Wheel
Tire tires[4];
Door doors[4];
Engine engine;
-----------
int pressGasPedal(int force)
int turnWheel(int degree)
int pressBrake();
getCurrentVelocity(): int

F100Pickup
-----------
Bed bed;
Color externalColor;
Color interiorColor;
Seat seats[2];
-----------
int pressBrake();
boolean isDoorOpen(int doorNum);

Camero
---------
AlpineStereo stereo;
Color externalColor;
Color interiorColor;
Seat seats[4];
---------
int pressBrake();
boolean isDoorOpen(int doorNum);

UML style

Vs

My Java style for writing attributes/method and return types
Vehicle
--------------
- Wheel theSteeringWheel;
- Tire tires[4];
- Door doors[4];
- Engine engine;
--------------
# int pressGasPedal(int force)
+ int turnWheel(int degree)
 + int pressBrake();
~ int getCurrentVelocity();

F150Pickup
------------
- Bed bed;
- Color externalColor;
- Color interiorColor;
- Seat seats[2];

Camero
------------
- AlpineStereo stereo;
- Color externalColor;
- Color interiorColor;
- Seat seats[4];

boolean openDoor(int doorNumber);
boolean isDoorOpen(int doorNum);

Is A

Camero
------------
- AlpineStereo stereo;
- Color externalColor;
- Color interiorColor;
- Seat seats[4];

F150Pickup
------------
- Bed bed;
- Color externalColor;
- Color interiorColor;
- Seat seats[2];

boolean openDoor(int doorNumber);
boolean isDoorOpen(int doorNum);
PickupTruck
-------------
Wheel wheels[4];
Seat seats[2];

Is A

FordF150
-------------
CargoBed bed;
TowHitch towHitch;

Is A (also called Specialization)

Printer
----------
Print(int.int)

GreenPrinter
----------
Print(int.int)

Method Overriding

2 instances of a Class

billsFordF150Pickup

nafisasPickup
Polymorphic Classes Represented in UML

Polygon

| numberOfSides : int = 0 |
| area : float = 0.0 |

calculateArea(): float <<abstract>>

Triangle

| numberOfSides=3 |
| Triangle(float, float, float) |
| computeArea(): float |

Square

| numberOfSides=4 |
| Square(float) |
| computeArea(): float |
Computer class (UML)

<<abstract>>
Computer

<<interface>>
IUSBInterface

<<interface>>
ISerialInterface

<<interface>>
IVideoCardInterface

Implements

IDeviceTypes

Is A

NonmodularComputer

Class Def

Is An Instance Of

myNonmodularComputer: NonmodularComputer

Object Instance
ComputerV2 class (UML)

<<abstract>>
ComputerV2
-------------
- serialIF: ISerialInterface
- USB_IF IUSBInterface
- videoCardIF IVideoCardInterface
-------------
+ ComputerV2()

<<interface>> IUSBInterface

<<interface>> ISerialInterface

<<interface>> IVideoCardInterface

<<interface>> IDeviceTypes

SuperDeluxoComputer

Is An Instance Of

mySuperDeluxoComputer: SuperDeluxoComputer

uses composition instead of inheritance
Inheritance

<<interface>>
ISteerable
-------------------------
float Turn(float degree);

Implements A
Vehicle
-----------------
Wheel wheels[];
Seat seats[];

Is A
Motercycle
-----------------
Wheels wheels[2];
Seat seats[1];

Has A
HandleBar
-----------------
float turn(float degree);

Is A
SteeringWheel
-----------------
turn(float degree):float

Is A
Car
-----------------
Trunk trunk;
Wheels wheels[4];
Seat seats[4];

Has A
PickupTruck
-----------------
CargoBed bed;
Wheels[4]
Seats seats[2];

Note: UML allows two styles for writing methods and attributes. The java style and the more general way.)
Telephone class

<<abstract>>
Telephone

-Color color;
-String phoneNumber;
-String phoneType;
-boolean inAphoneCall=false;

getAConnection():boolean
boolean makeACall(String num);
boolean talk(String whatToSay);
String listen();
boolean hangUp();

<<interface>>
IDialerDevice

-------------------------
dial(String number): boolean;

<<interface>>
ISoundInputDevice

--------------------------------------------------------
getSoundIn(Sound soundReceived)

<<interface>>
ISoundOutputDevice

----------------------------------------------
putSoundOut(Sound sound)
changeVolume(float percentage)
mute(boolean muteOn);

UML style Vs My Java style

DeskPhone

-----------------------

CordlessPhone

-----------------------

Telephone

-------------------------

<<interface>>
IDialerDevice

-------------------------
dial(String number): boolean;

<<interface>>
ISoundInputDevice

--------------------------------------------------------
getSoundIn(Sound soundReceived)

<<interface>>
ISoundOutputDevice

----------------------------------------------
putSoundOut(Sound sound)
changeVolume(float percentage)
mute(boolean muteOn);
PayPhone and TV classes

```
<<abstract>>
Telephone
-------------

<<interface>>
IDialerDevice
-------------------------
   dial(number)

<<interface>>
ISoundInputDevice
-------------------------
   getSoundIn(Sound soundReceived)

<<interface>>
ISoundOutputDevice
-------------------------
   putSoundOut(Sound sound)
   changeVolume(float percentage)
   mute(boolean muteOn)

<<abstract>>
PayPhone
-------------------------
   amountDeposited;
   CoinSlot myCoinSlot;
   CoinReturn myCoinReturn;
-------------------------
   insertCoin(float coin);
   pressCoinReturn();

<<abstract>>
TV
-------------------------
```
Subclasses – abstract and concrete

<<abstract>> Telephone

<<abstract>> CellPhone

<<abstract>> Fax<Machine

Blackberry

<<abstract>> SpeakerPhone

myBlackberry

<<interface>> IAbilityToDial

<<interface>> IAbilityToInputSound

<<interface>> IAbilityToOutputSound

<<interface>> IAbilityToConnectToWiring

Notation shows an instance of a class