

Object Oriented Programming

LESSON 01

Concept Object

Outline

1. What is Object?
2. Object examples
3. Encapsulation Concept
4. What is Class?
5. Example using Class and Object

Overview

In this chapter, you are going to learn about

- Know Object
- Know how to use Object
- Know Encapsulation in Object oriented world
- Know Class
- Know how to use Class

Learning content



1. What is Object?
 - Definition
 - Properties of object
 - Example
2. Object examples
 - Classroom
 - Vehicle
 - Fruit
3. Encapsulation concept
 - Definitions
 - Advantages
 - Example
4. What is Class?
 - Inheritance
 - Advantages of Inheritance
 - Object vs Procedural approach
5. Example using Class and Object
 - Animated Smileys
 - Animated Smileys Classes
 - Animated Smileys Code

Pre-Test



Question	Possible answers	Correct Answer	Feedback of the question
Which one is in group of fruit?	1. Rock 2. Land 3. Grape	3. Grape	Rock is hard and not belong to tree. Land is where we grow tree. Grape is something produced by tree

Pre-Test



Question	Possible answers	Correct Answer	Feedback of the question
Which one is not visible in Jackfruit?	1. Yellow inside 2. Taste sweet 3. Leaf	2. Taste sweet	Yellow color we know by seeing it. Leaf we can see on the tree. But, taste we can not see, we know when test it.

Pre-Test



Question	Possible answers	Correct Answer	Feedback of the question
Which one is part of PC?	1. Table 2. CPU 3. UPS	3. CPU	PC can be placed on a place other than table. PC can plug direct to electric source without UPS. But CPU is required for PC to operate.

1.1. What is Object?

- All identifiable entity models, concrete or abstract, handled by the software application
 - tangible thing
Ex: town, vehicle, student, a button on the screen
 - a conceptual thing
Ex: date, meeting, booking schedule



1.2. Properties of Object

- A unique identity (to distinguish one object from another)
- An internal state given by values of variables (or attributes)
 - Attributes describe the state of the object at a given time
Ex: patient measures 1.82 m and weights 75 Kg
 - Attributes are typed and named
Ex: float height; float weight;
- A Behavior (action capacities of object) given by functions or sub-programs, called methods (or operations)
 - Methods define the behavior of the object (what it can do, what it can be done ...) and his reactions to external stimulation
Ex: a student takes an exam, etc ...
 - Methods implement the algorithms that can be invoked on this object



1.2. Properties of Object

- Object = data + algorithms
 - An object is the collection of data (variables or attributes) and processes (methods) associated
- Example: red pen
 - Data:
 - Color: red
 - Mark: haknuman
 - Made: in Cambodia
 - Description: easy to use on any plain surface
 - Process:
 - Draw
 - Lock
 - Unlock
 - Write
 - Spray



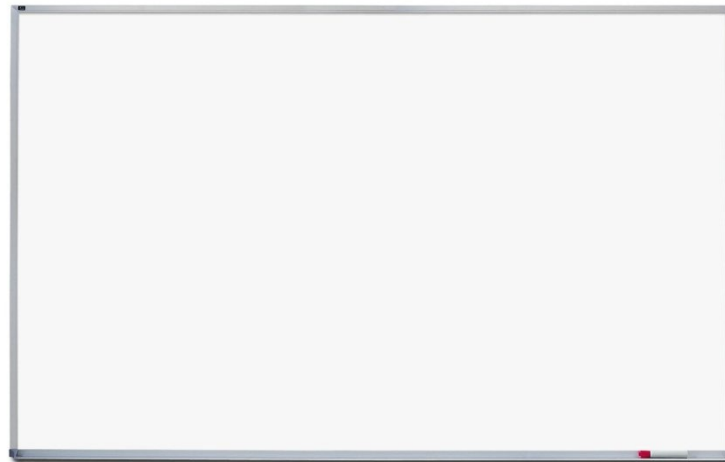
1.3. Example

- Computer
 - Dell Latitude 620D red belongs to Mr. Sreng
 - TOSHIBA color black belongs to Mr. Sreng
- Things
 - Thida's bag
 - Kagna's note book
 - My father's bicycle is now very old one.
 - The white
- Animals
 - A Dog next door is named John
 - My cat is named BlackCool, he is black the whole body except his nose is white. He is very special mouse catcher agent.

2.1. Classroom



- Whiteboard at room 306-F is very clean, and easy to write and erase.
- A Teacher's Table in room 210-F is very old and color gray
- A PC code 209F-05 is not working well
- A mouse model LogiTech in room 206-F is in good condition



2.2. Vehicle



- Motorbike:
 - model Wave alpha,
 - red color,
 - machine code 2131-2312-313111,
 - wheel size 16 inches
- Car:
 - model Toyota Siena,
 - red color,
 - machine code 6131-2612-361211,
 - wheel size 20 inches
- ...

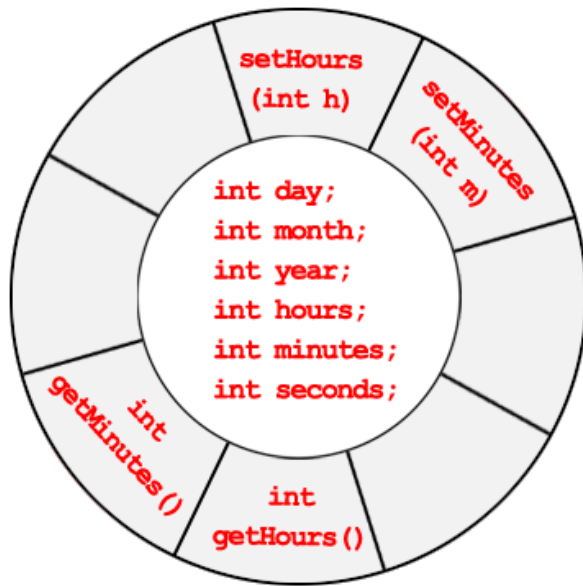
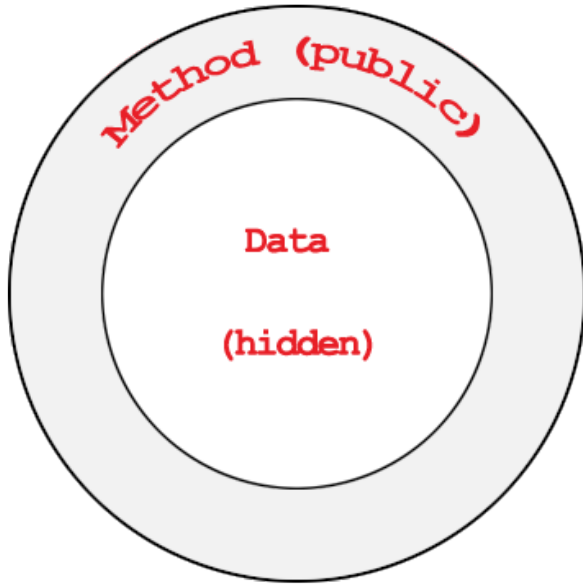


2.3. Fruit



- Jackfruit
 - Made in Cambodia
 - Weight: 5 kilograms
 - Type: Normal
 - Color: green
 - Condition: good
- Papaya
 - Made in Cambodia
 - Weight: 2 kilograms
 - Type: Normal
 - Color: yellow
 - Condition: good



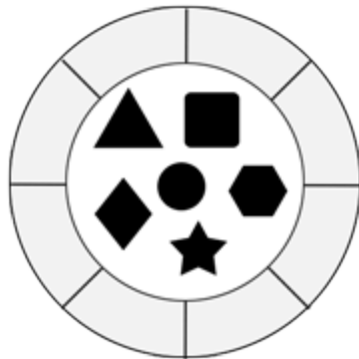


3.1. Definition of Encapsulation

- An **Object = Data + algorithms**
An object is a group of data (variables or attributes) and its treatments (methods)
- Principle of encapsulation
 - Access to data (state) of the object can not be done only through methods
 - Data is private (hidden)
 - Public methods define the interface (what let other to see) of object
- Example: Clock Object

3.2. Advantage of Encapsulation

- Data structures change does not affect programs that use the object
- Objects interact and communicate with each other by sending messages
- Public methods of an object correspond to messages that can be send
- Messages are characterized by:
 - Target object (receiver) Message
 - Name of the method to trigger
 - Parameters of this method



Object named myClock

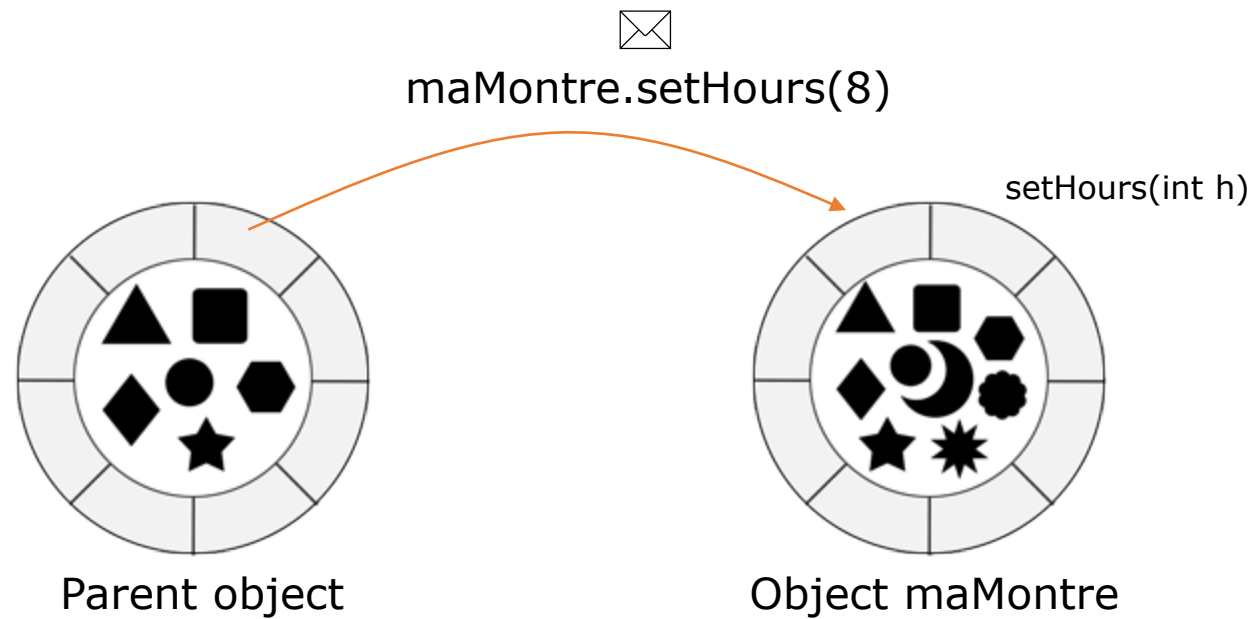
```
myClock.setHours(3);  
myClock.setMinutes(45);  
myClock.setSeconds(2);
```

```
myClock.getMonth();  
myClock.getYear();
```


3.3. Example

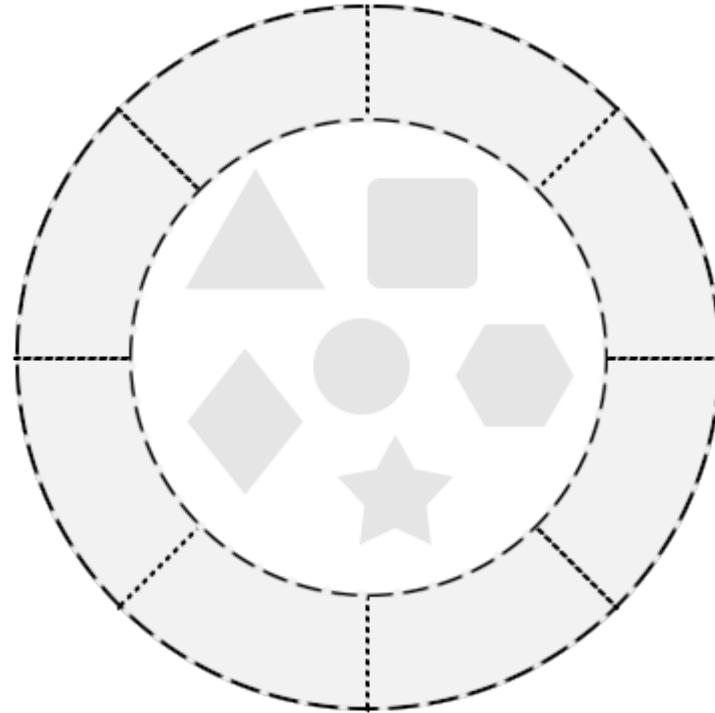


- Example demonstrate how messages communication between objects



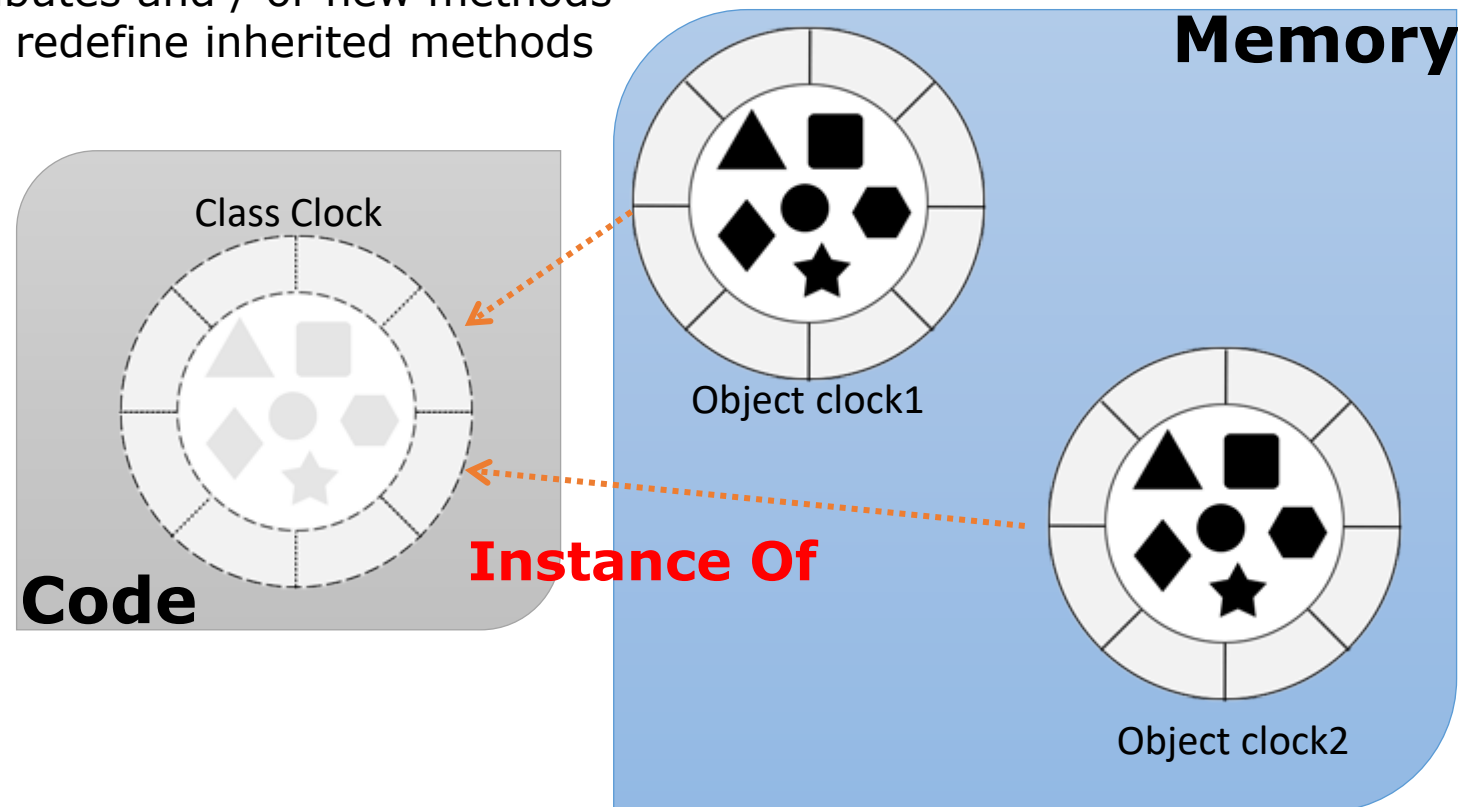
4. What is Class?

- Objects (instances) are created (instantiated) from "molds" the classes
- Class = Schema / mold / object model, it describes:
 - Private part
 - internal data structure (attributes)
 - bodies of methods (algorithms)
 - Public part (interface)
 - names and methods parameters
- Class = objects generator
by instantiation, we can create objects
that obey this pattern / mold / model



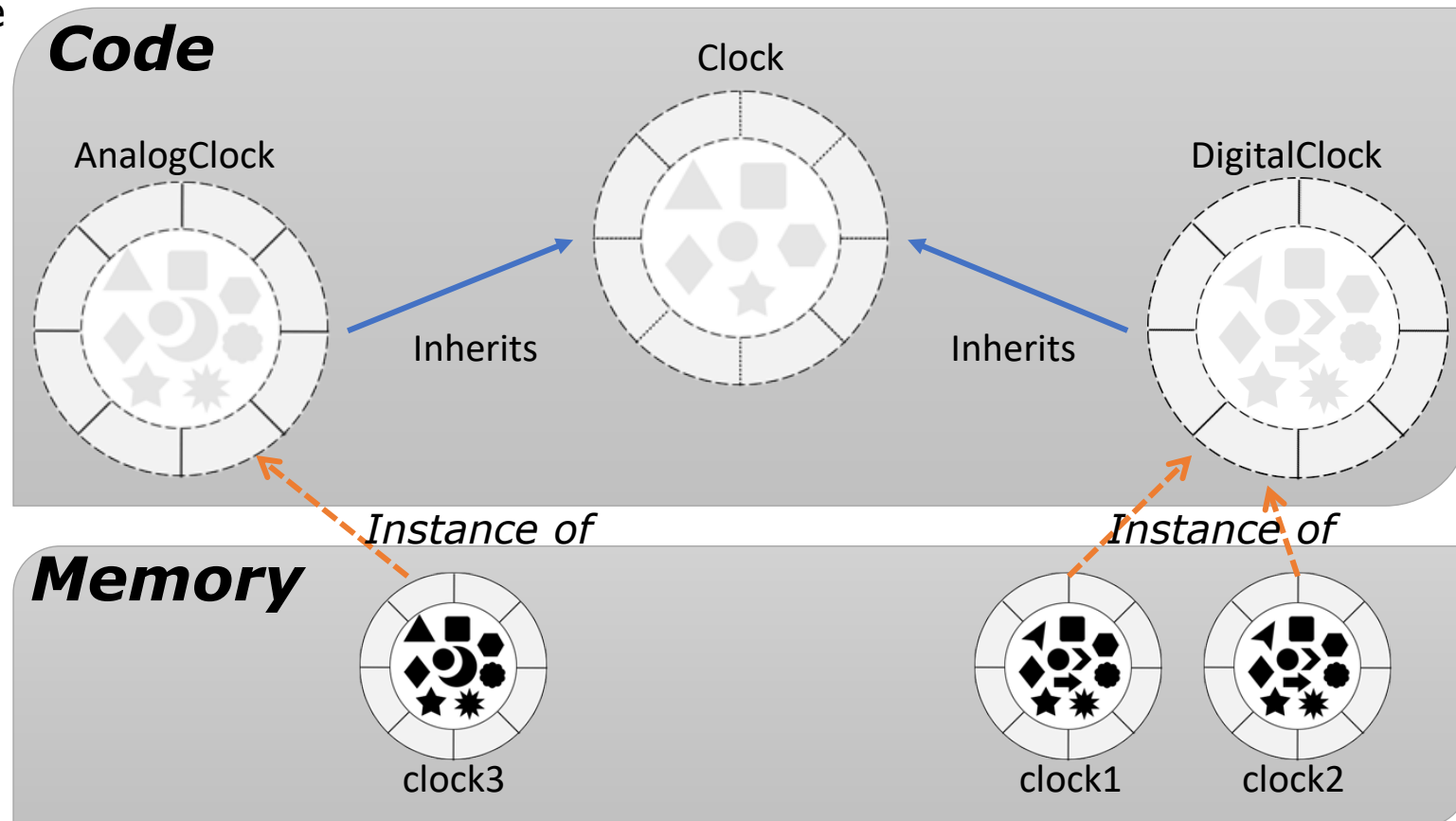
4.1. Inheritance

- Class = refinement / specialization of an existing class
- They form a class hierarchy, wherein each class:
 - Inherits attributes and methods of its ancestors / super-classes
 - Adds new attributes and / or new methods
 - May modify or redefine inherited methods



4.2. Advantages of Inheritance

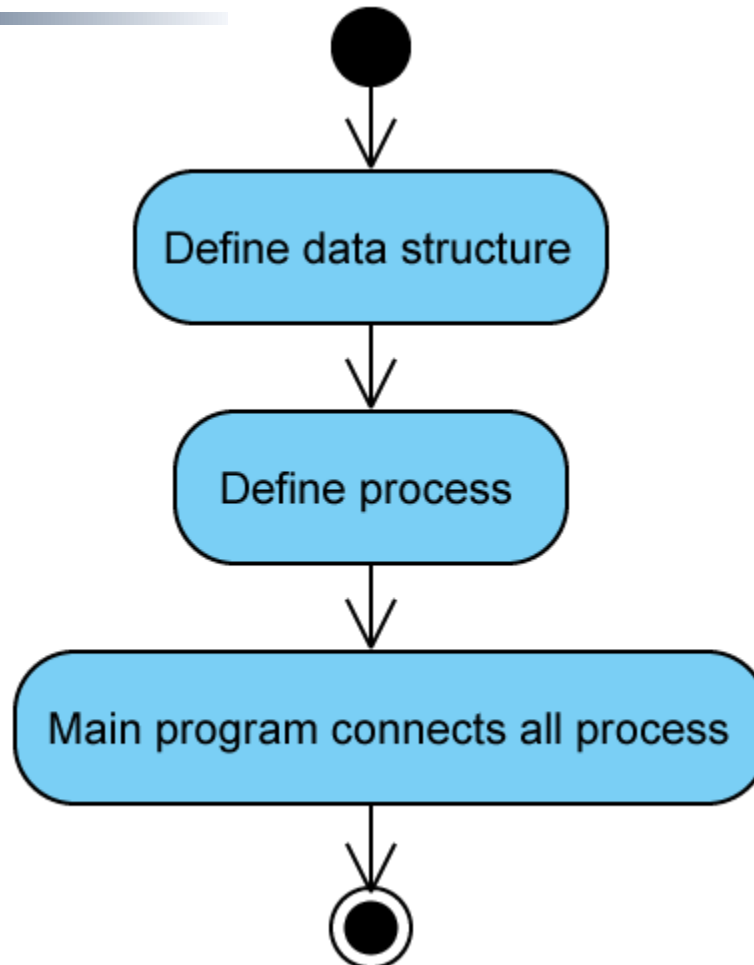
- Code Reuse
- No need to reinvent the wheel each time
- Extensible & Flexible



4.3. Object VS Procedural Approach



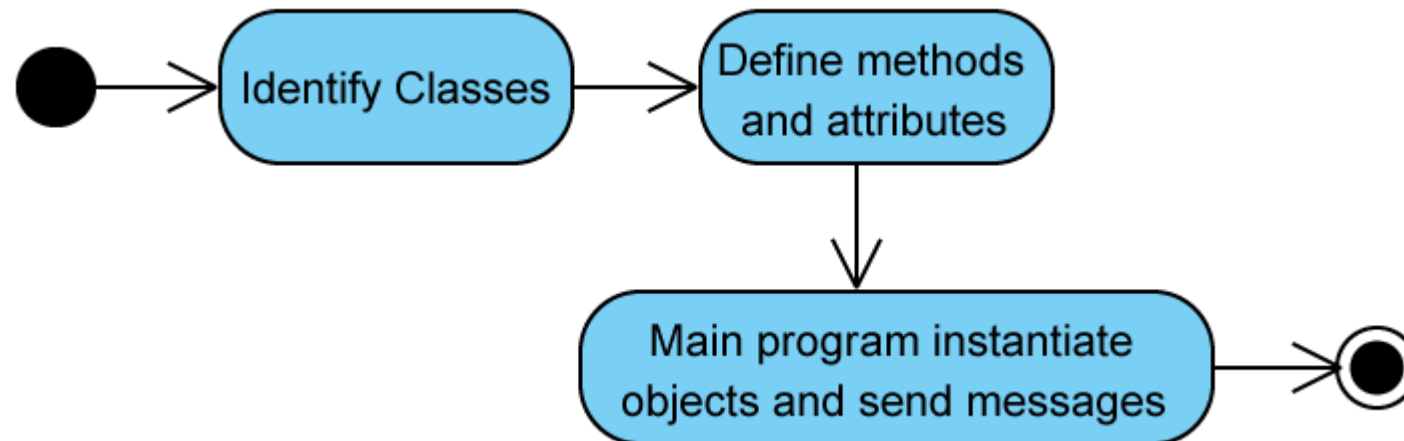
- Procedural approach
 - Define data structures
 - Define treatments (process)
 - Top-down analysis
 - The main program connects treatments (process)



4.3. Object VS Procedural Approach



- Object approach
 - Identify classes
 - For each class
 - Define the public interface (method signatures)
 - Define its implementation (attributes, method bodies)
 - The main program:
 - creation (instantiation) of objects in memory
 - launches execution by sending messages to the created objects
 - These messages can cause sending other messages and / or the creation of other objects

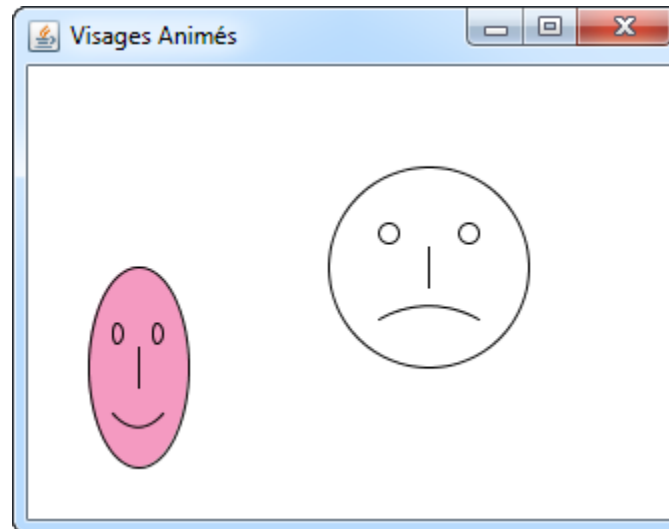


5. Example using Class and Object



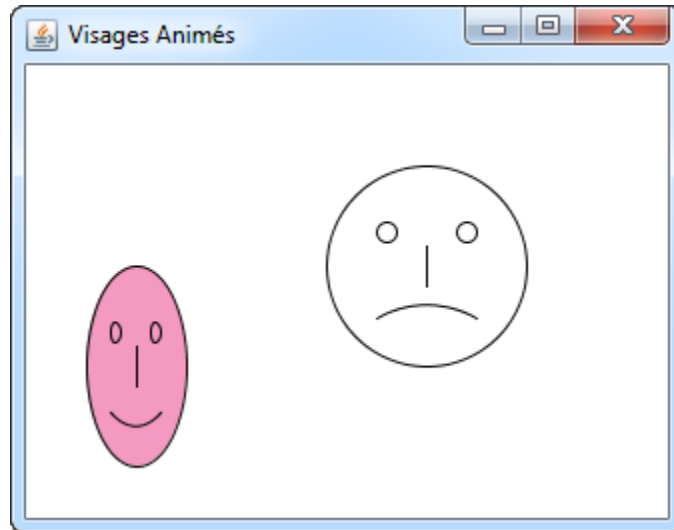
In this example will include

- Java Graphical User Interface (GUI)
- Some small java codes
- Demonstration desktop application
- Source code



5.2. Animated Smiley Classes

- Analyzed the GUI and find:
 - Objects?
 - Action capacity of these objects: What do they do?
 - Object attributes: what are their properties?



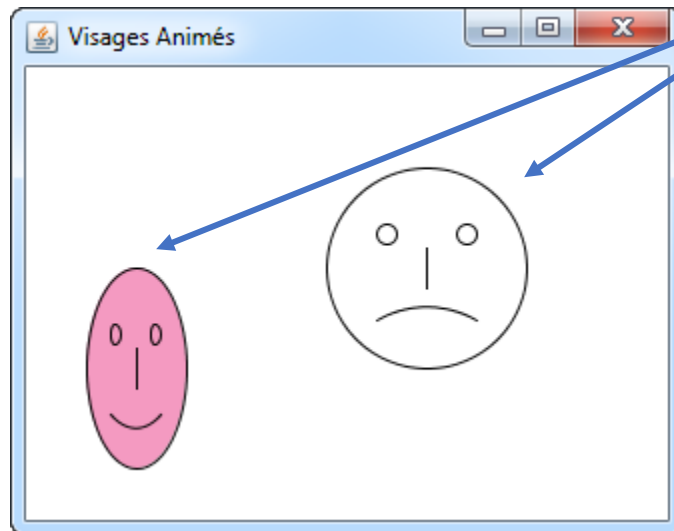
Window

- Action capacities
 - Exit
 - Iconify
 - Bring to front
 - ...
- Attributes
 - Position
 - Width, height
 - ...

Using existing class in Java (JFrame)

5.2. Animated Smiley Classes

- Analyzed the GUI and find:
 - Objects?
 - Action capacity of these objects: What do they do?
 - Object attributes: what are their properties?



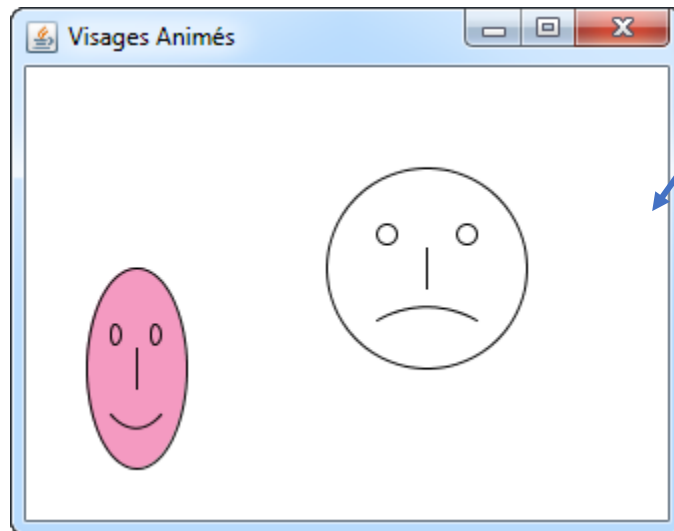
Smileys

- Action capacities
 - Smile
 - Advance
 - Change direction
 - Display
 - ...
- Attributes
 - Color
 - Position
 - Width, Height

Create a new class (Smiley)

5.2. Animated Smiley Classes

- Analyzed the GUI and find:
 - Objects?
 - Action capacity of these objects: What do they do?
 - Object attributes: what are their properties?



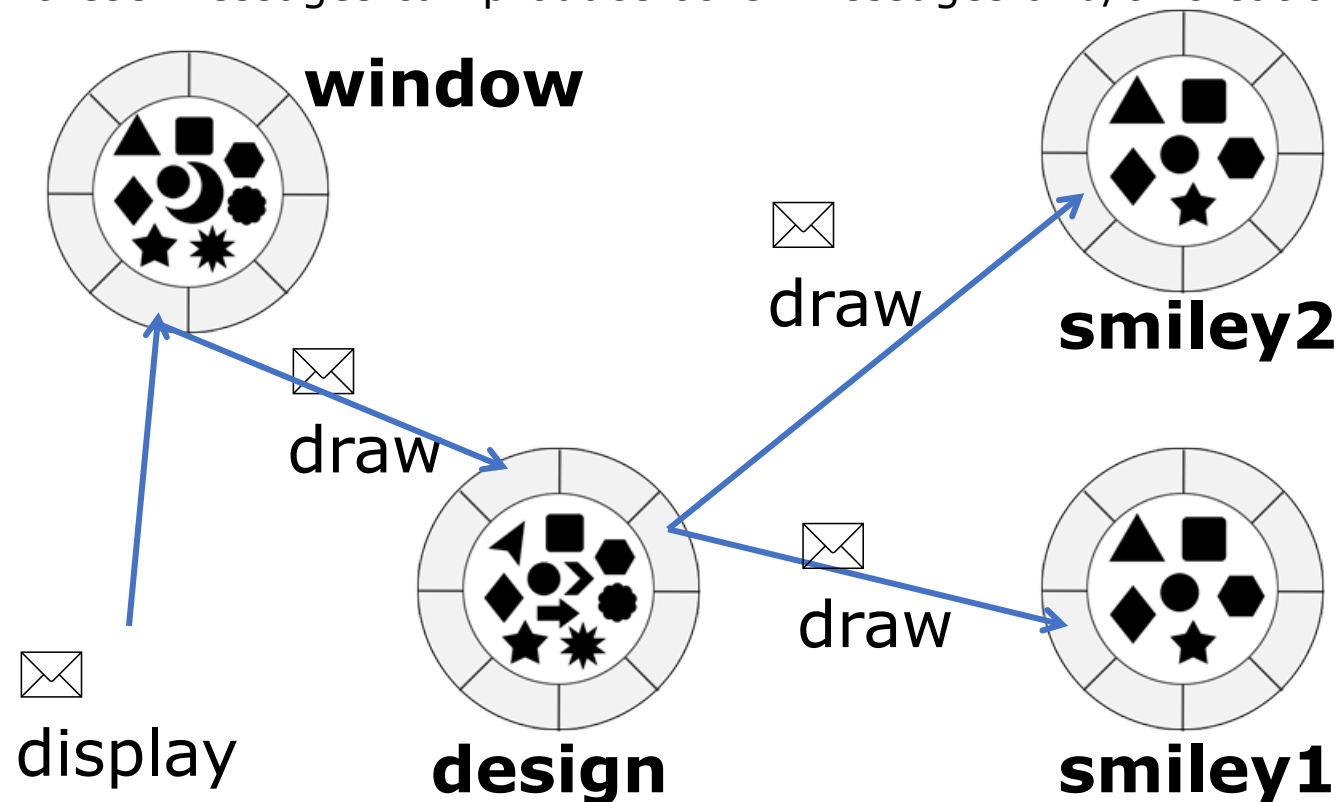
Design zone

- Action capacities
 - Erase
 - Display objects it has
 - ...
- Attributes
 - Window that it is in
 - Background color
 - ...

Create a new class (Design) by reusing (inheritance) existing class in Java (JPanel)

5.2. Animated Smiley Classes

- An Object oriented application comport by :
 - Creating (instantiating) objects in memory
 - Launch execution by sending messages to created objects
 - these messages can produce other messages and/or creation of other objects



5.3. Animated Smiley Codes

- Java Code

Creation of
objects in
application

```
Window window = new Window();
Design design = new Design();
window.setContentPane(design);
window.afficher();

Smiley smiley1 = new Smiley();
Smiley smiley2 = new Smiley(150, 50, 100, 100,
                           Color.white);
```

Send messages
to objects

```
smiley2.sad();
design.ajouterObjet(smiley1);
design.ajouterObjet(smiley2);
design.draw();
while(true){
    smiley1.move();
    smiley1.changeDirection();
    smiley2.move();
    smiley2.changeDirection();
    window.repaint();
    delay(100);
}
```

Test



Question	Possible answers	Correct Answer
1. Object is:	<ul style="list-style-type: none">a) Any visible thingb) An Identifiable entity model handled by softwarec) Any movable thing on earth	
2. Completing blank field:	Class is mold/...../..... model; it describes attributes and methods for object	

Test



Question	Possible answers	Correct Answer
3. Choose a name that is not mentioned in this lesson:	<ul style="list-style-type: none">a) Objectb) Classc) Inheritsd) Instance ofe) Time	
4. Inheritance is:	<ul style="list-style-type: none">a) Refinement of another existing classb) Specialization of a new classc) Relationship that one class belongs to another	
5. Inheritance inherits:	<ul style="list-style-type: none">a) Attributes of child classb) Attributes of super classc) Methods of super classd) Methods of child class	

Practice



No.	Exercise	Solution
1.	Setup Eclipse environments	
2,	Create HelloJava class	
3,	Create HelloSomeone class	

Summarize



- Objects are all identifiable entity models, concrete or abstract, handled by the software application.
- Class is schema/mold/object model.
- We can Instantiate many objects from one class.
- Inheritance inherits all attributes and methods of super class.
- Java provides implements OOP.

Reference



- “Object Oriented Programming” by Ph. Genoud – Université Joseph Fourier, 2006
- “Building Skills in Object-Oriented Design” by Steven F. Lott, 2009
- “Exercices en Java” 2nd edition by Groupe Eyrolles, 2006
- “Java Programming – Introductory” by Joyce Farrell, 1999
- “Java Examples in a Nutshell” 3rd Edition by David Flanagan, 2004
- “Programmer en Java” 5th edition by Groupe Eyrolles, 2008
- <https://docs.oracle.com/javase/tutorial/java/javaOO/>