Java un langage Orienté Objets

L'approche Objets

Philippe Genoud

Philippe.Genoud@imag.fr

Le langage Java
un langage orienté-objet

- **Qu'est-ce qu'un objet ?**
  - **Modélise** toute entité identifiable, concrète ou abstraite, manipulée par l'application logicielle
    - une chose tangible
      ex: ville, véhicule, étudiant, un bouton sur l'écran
    - une chose conceptuelle
      ex: date, réunion, planning de réservation
  - **Réagit** à certains messages qu'on lui envoie de l'extérieur; la façon dont il réagit détermine le **comportement** de l'objet.
  - Ne réagit pas toujours de la même façon à un même message; sa réaction dépend de **l'état dans lequel il se trouve**.

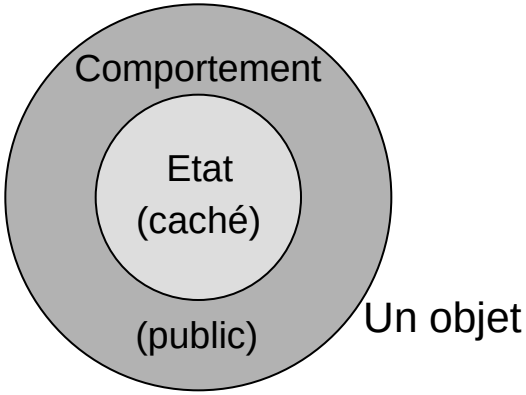
Le langage Java
un langage orienté-objet

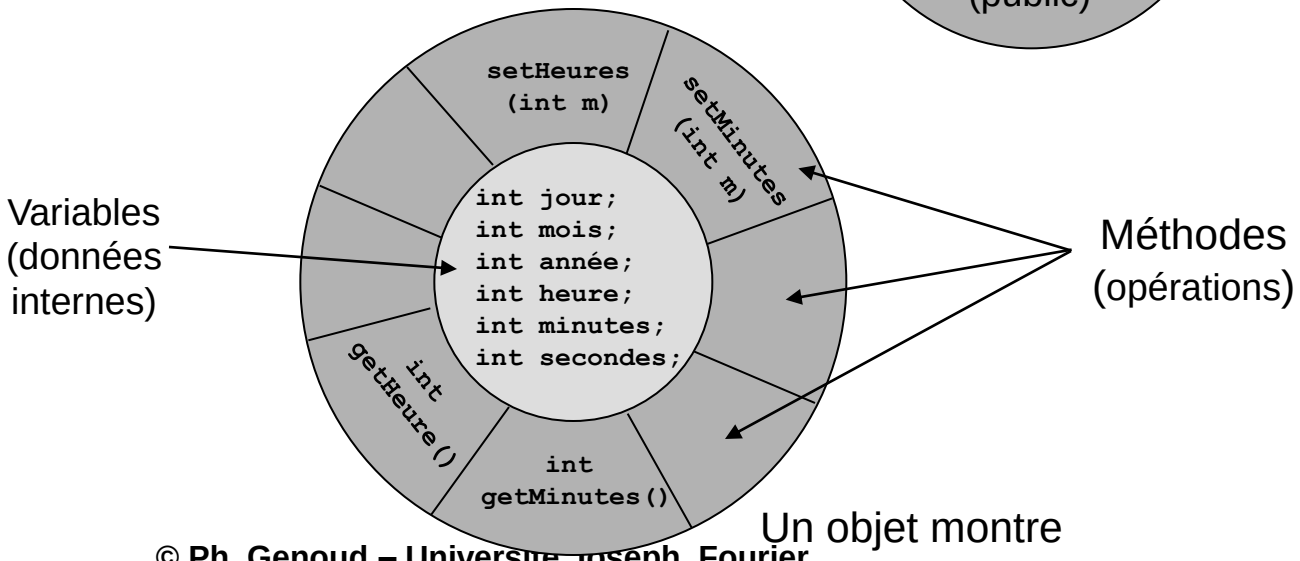
- **Un objet possède :**
  - Une **identité unique** (permet de distinguer un objet d'un autre)
  - Un **état interne** donné par des valeurs de **variables (ou attributs)**
    - Attributs décrivent l'état de l'objet à un instant donné
      - ex: patient mesure 1,82 m et pèse 75 Kg
    - Attributs sont typés et nommés
      - ex: `float hauteur; float poids;`
  - Un **comportement** (capacités d'action de l'objet) donné par des fonctions ou sous-programmes, appelés **méthodes** (ou opérations).
    - Méthodes définissent le comportement de l'objet (ce qu'il peut faire, comment il peut le faire…) et ses réactions aux stimulations externes
      - ex: un étudiant passe un examen, etc…
    - Méthodes implémentent les algorithmes invocables sur cet objet

Le langage Java
un langage orienté-objet

- **Un objet = données + algorithmes**
  - Un objet est le regroupement de données (variables ou attributs) et des traitements (méthodes) associées

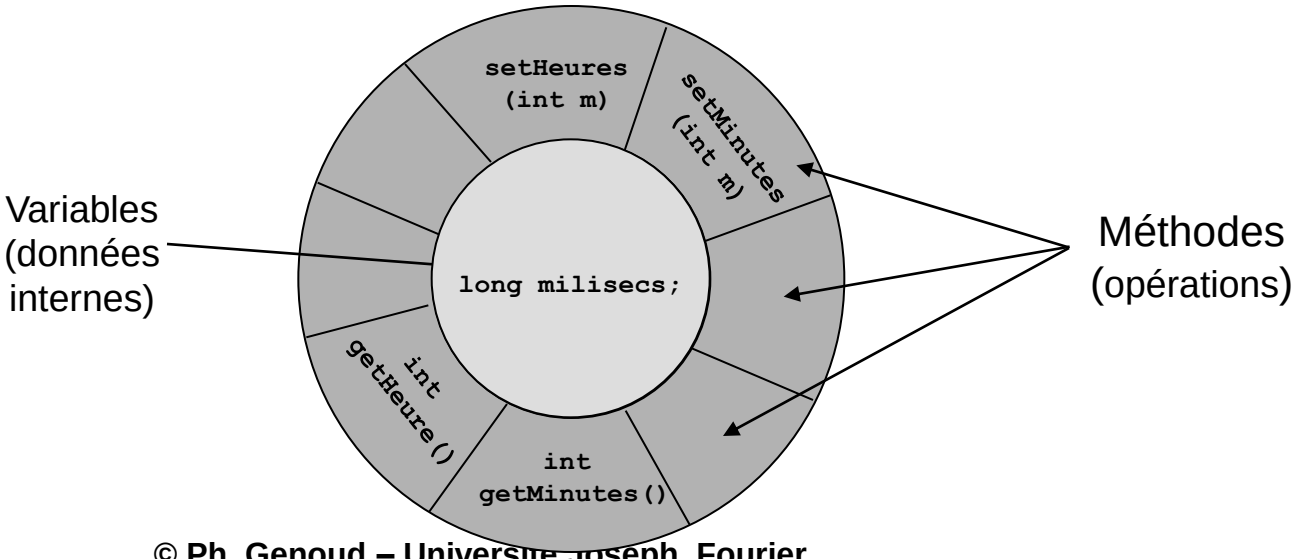
- **Principe d'encapsulation**
  - L'accès aux données (état) de l'objet ne peut être fait qu'au travers des méthodes.
  - Les données sont **privées** (cachées)
  - Les méthodes **publiques** définissent **l'interface** de de l'objet





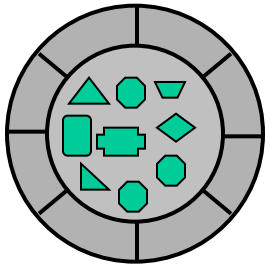
Le langage Java
un langage orienté-objet

- **Intérêt de l'encapsulation**
  - Modification des structures de données n'affecte pas les programmes qui utilisent l'objet.



Le langage Java
un langage orienté-objet

- Les objets **interagissent** et **communiquent** entre eux par l'envoi de **messages**
- **Les méthodes publiques d'un objet correspondent aux messages que l'on peut lui envoyer**
- Les messages sont caractérisés par
  - *objet cible (récepteur) du message*
  - *nom de la méthode à déclencher*
  - *paramètres de cette méthode*



objet ***maMontre***

maMontre.setHeure(14);

Envoi d'un message à l'objet identifié par maMontre

maMontre.setMinutes(45);

Objet récepteur du message

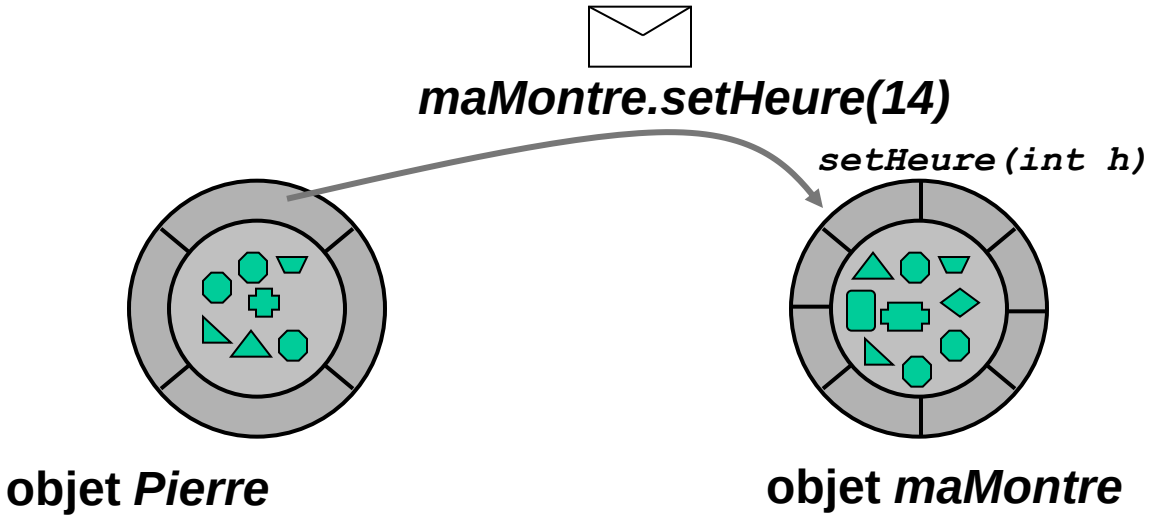
Méthode invoquée

Paramètre du message

maMontre.getMois();

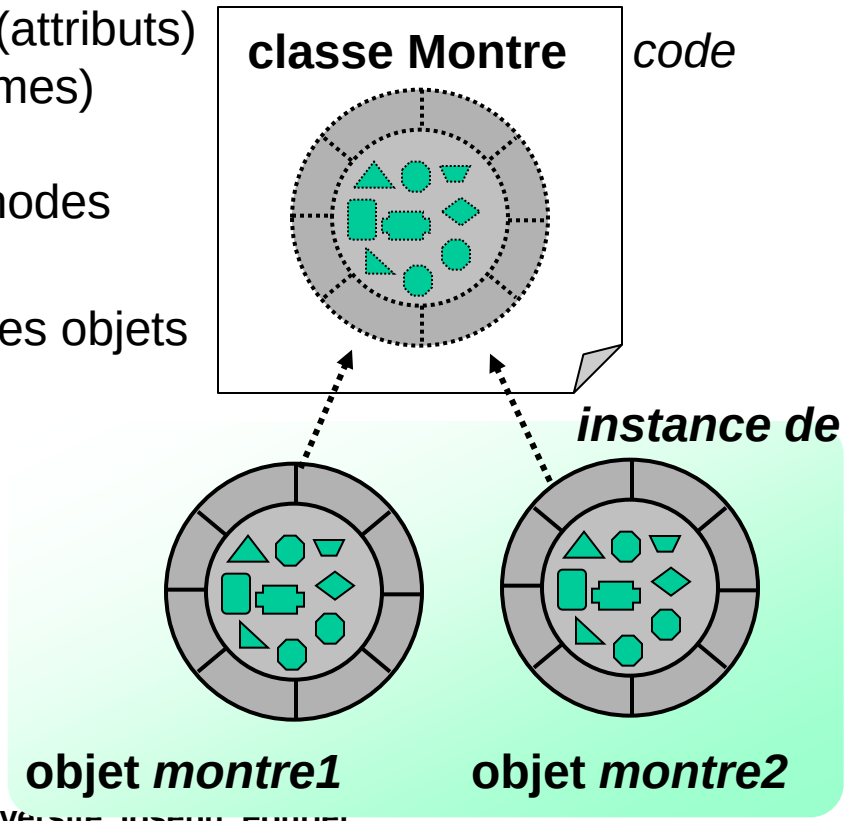
Le langage Java
un langage orienté-objet

- Les objets **s'envoient des messages entre eux**



Le langage Java
un langage orienté-objet

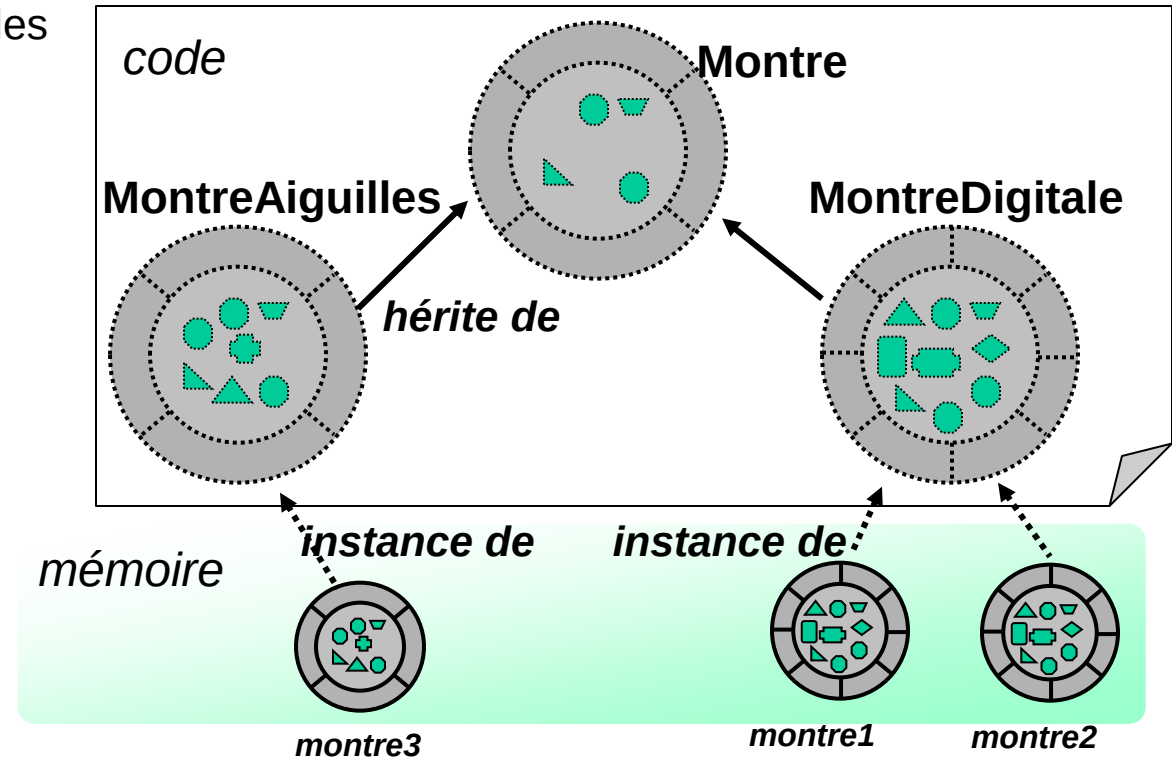
- Les objets (instances) sont créés (**instanciés**) à partir de "moules" : les **classes**
- Classe = schéma/moule/modèle d'objets, elle décrit :
  - **partie privée**
    - structure de données interne (attributs)
    - corps des méthodes (algorithmes)
  - **partie publique** (interface)
    - noms et paramètres des méthodes
- Classe = générateur d'objets
  par **instanciation**, on peut fabriquer des objets obéissant à ce schéma/moule/modèle



Le langage Java
un langage orienté-objet

- Classe = raffinement/spécialisation d'une classe existante
- Elles forment une **hiérarchie de classes**, où chaque classe :
  - **hérite** des attributs et méthodes de ses ancêtres/**super-classes**
  - ajoute de nouveaux attributs et/ou de nouvelles méthodes
  - peut modifier ou redéfinir les méthodes héritées

- **Intérêt héritage :**
  - Réutilisation du code
  - Pas besoin de réinventer la roue à chaque fois



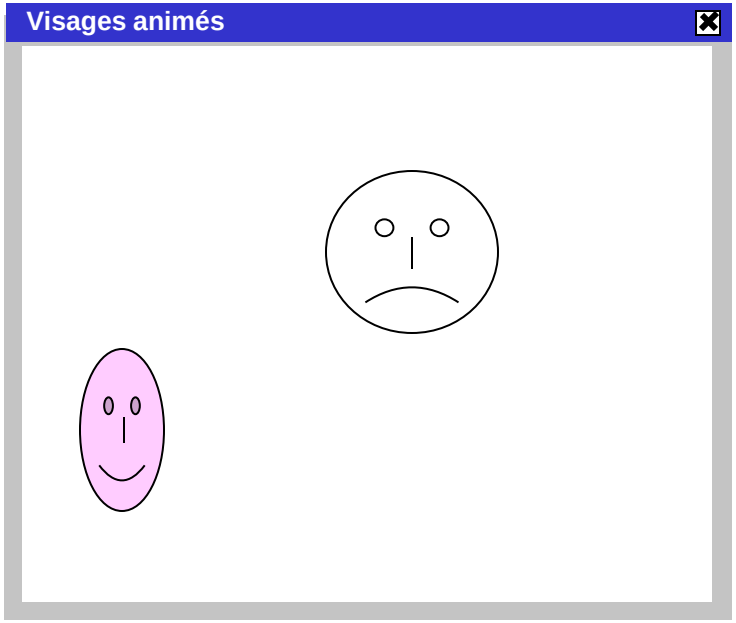
Le langage Java
un langage orienté-objet

- **Approche procédurale (C)**
  - Définir les structures de données
  - Définir les traitements
    - Analyse descendante
  - Le programme principal enchaîne les traitements.

- **Approche Objet**
  - Identifier les classes
  - Pour chaque classe
    - Définir son interface publique (signature des méthodes)
    - Définir son implémentation (attributs, corps des méthodes)
  - Le programme principal :
    - création (instanciation) d'objets en mémoire
    - lance exécution par envoi de messages aux objets créés
    - ces messages peuvent provoquer d'autres envois de messages et/ou la création d'autres objets

Le langage Java
un langage orienté-objet

- **Exemple : les visages animés** Demo
  - Quels sont les concepts/les objets composant le jeu ? → Classes
  - Que doivent savoir faire ces objets ? → Méthodes
  - Quelles sont leurs propriétés ? → Attributs

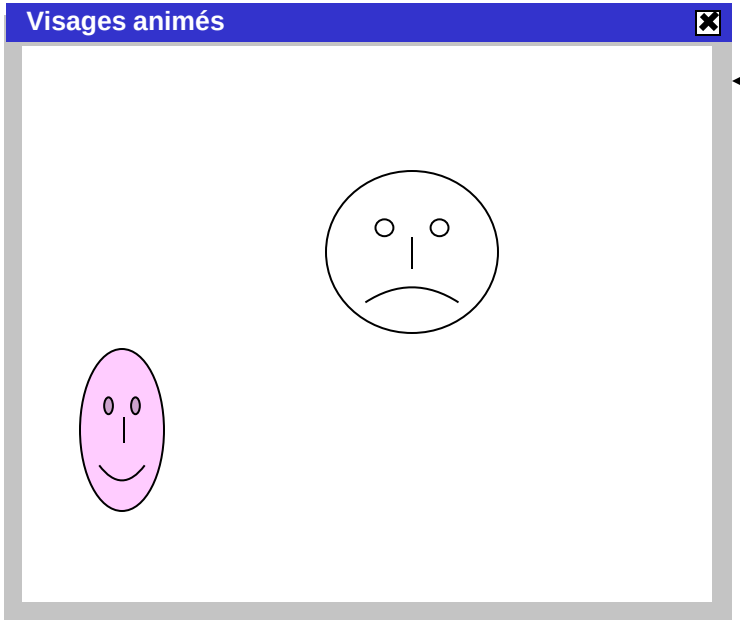


Dans une fenêtre[1] de titre Visages Animés deux "visages" se déplacent et rebondissent lorsqu'ils touchent les bords de la fenêtre. L'utilisateur peut déplacer, agrandir, réduire la fenêtre, la placer au premier plan... Les visages peuvent être ronds ou ovales. Il sont dessinés de manière schématique, deux ronds représentant les yeux, un trait représentant le nez, un arc de cercle la bouche. Lorsqu'un visage touche l'un des bords de la fenêtre, l'expression du visage est modifiée (trois expressions sont définies : triste (bouche vers le bas), heureux (bouche vers le haut), neutre (bouche = trait horizontal). Un visage est rose l'autre blanc…

[1] La fenêtre est composée de deux parties : son cadre et son intérieur (une zone banche) Pour être plus précis on devrait dire que les visages se déplacent et rebondissent dans cette zone.

Le langage Java
un langage orienté-objet

- **Exemple : les visages animés**
- Objets ?
- Capacité d'action des objets : que font-ils ?
- Attributs des objets : quelles sont leurs propriétés ?

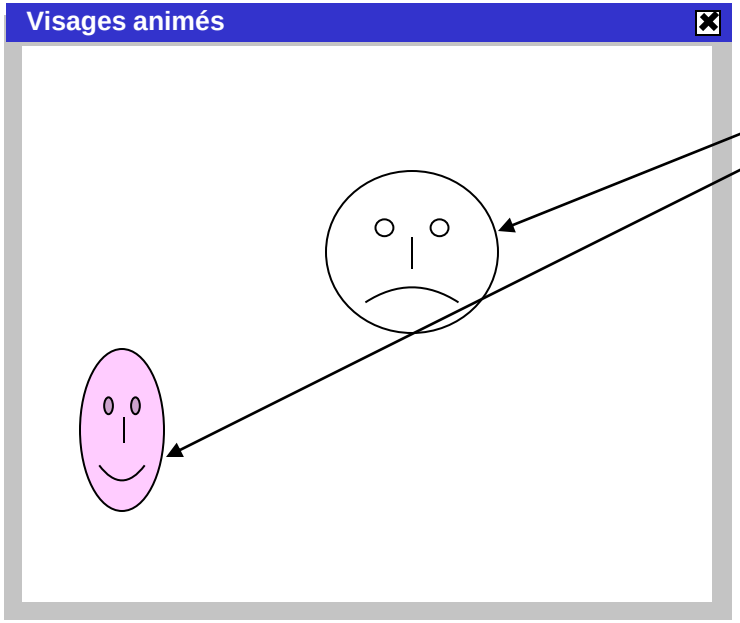


Fenêtre
- capacités d'action
  - se fermer
  - s'iconifier
  - passer au premier plan
  - ...
- attributs
  - position
  - largeur, hauteur
  - ...

Utilisation d'une classe existante dans Java (JFrame)

Le langage Java
un langage orienté-objet

- **Exemple : les visages animés**
- Objets ?
- Capacité d'action des objets : que font-ils ?
- Attributs des objets : quelles sont leurs propriétés ?



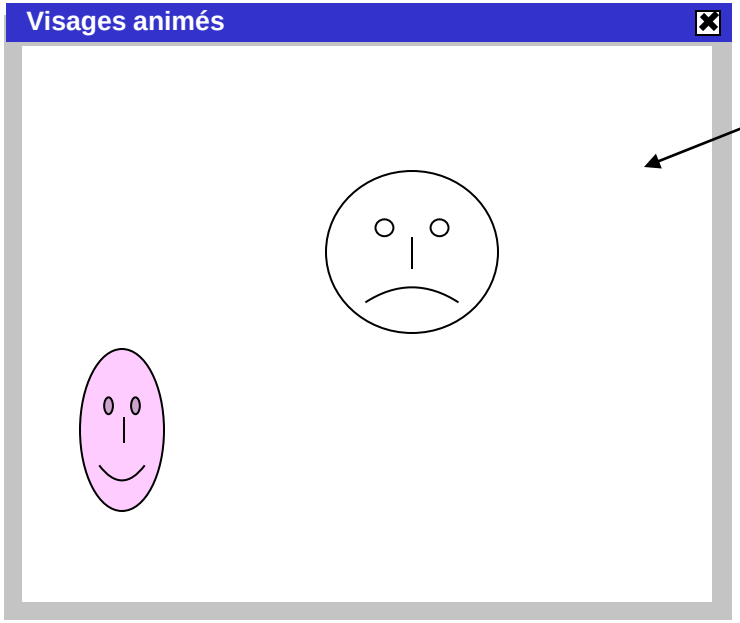
Visages

- capacités d'action
  - sourire
  - faire la tête
  - avancer
  - changer de direction
  - s'afficher
- attributs
  - couleur
  - position
  - largeur, hauteur
  - ...

Ecriture d'une nouvelle classe (`Visage`)

Le langage Java
un langage orienté-objet

- **Exemple : les visages animés**
- Objets ?
- Capacité d'action des objets : que font-ils ?
- Attributs des objets : quelles sont leurs propriétés ?



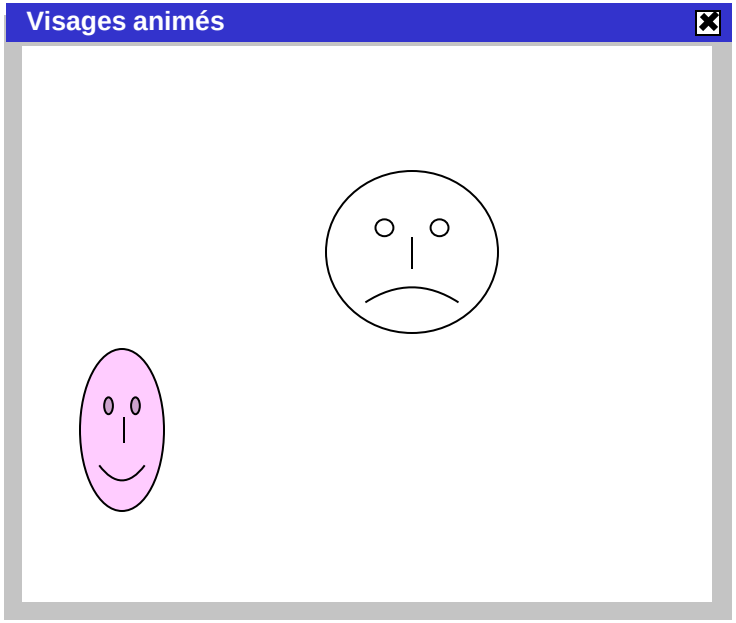
Zone de dessin

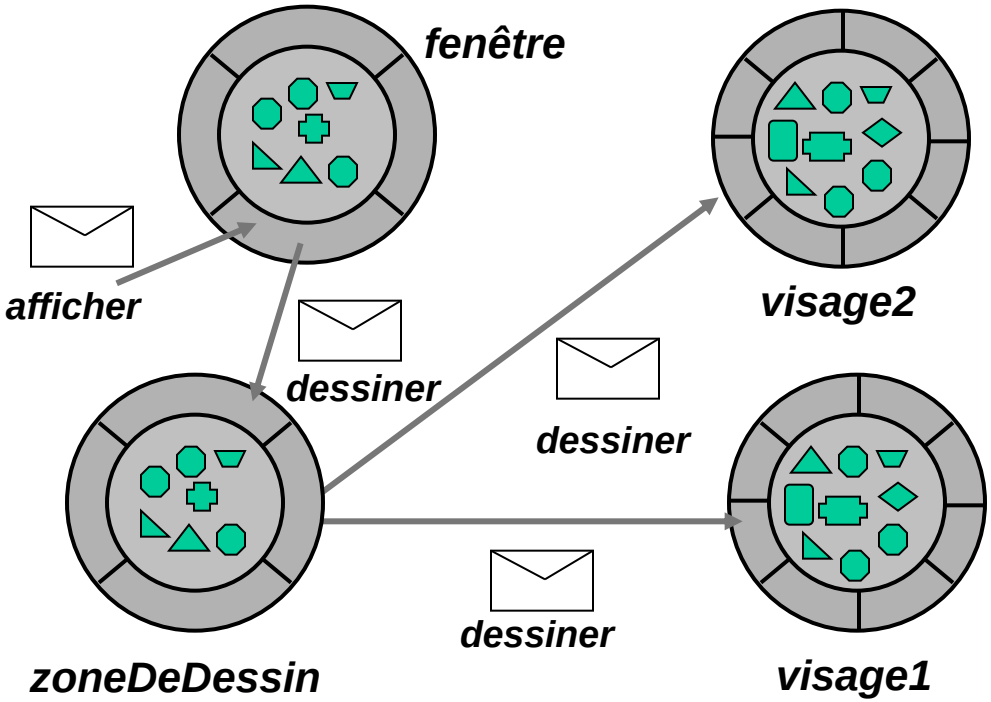
- capacités d'action
  - s'effacer
  - afficher les objets qu'elle contient
  - ...
- attributs
  - fenêtre où elle se trouve
  - couleur du fond
  - ...

Ecriture d'une nouvelle classe (`Dessin`) en réutilisant (héritage) une classe existante dans Java (`JPanel`)

Le langage Java
un langage orienté-objet

- Une application orientée objet consiste en :
  - *création (instanciation) d'objets en mémoire*
  - *lancer exécution par envoi de messages aux objets créés*
    - ces messages peuvent provoquer d'autres envois de messages et/ou la création d'autres objets





Le langage Java

un langage orienté-objet : Exemple de programme Java

Création des objets de l'application

Envoi de messages aux objets

```
class DemoVisagesAnimes
{
  public static void main(String[] argv)
  {
      JFrame fenêtre = new JFrame("Titre de la fenêtre");
      // Création d'un objet de type Dessin. Cet objet est destiné à
      // contenir les objets graphiques gérés par l'application.
      Dessin dessin = new Dessin();
      // Insertion de objet de type Dessin dans la fenêtre de l'appli.
      fenêtre.add(dessin);
      // Affichage de la fenêtre
      fenêtre.show();
      //  ajout, modification des objets visage
      Visage v1 = new Visage();
      Visage v2 = new Visage(200,100,200,200,5);
      dessin.ajouterObjet(v1);
      dessin.ajouterObjet(v2);
      v2.pleurer();
      dessin.repaint(10);
      while (true) {
           v1.deplacer();
           v2.deplacer();
           dessin.repaint(10);
      }
  }
}
```