

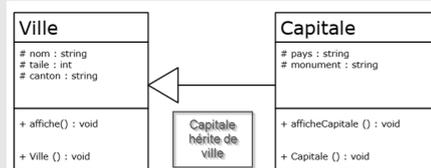
POO

Enca Regrouper caractéristiques et fonctionnalité et cacher le fonctionnement interne

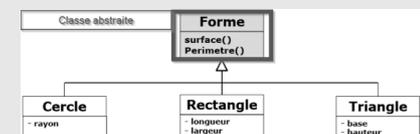
Héritage Comme l'héritage génétique d'un humain, accepte le positif et le négatif. C'est un gain de temps en réutilisant des objets.

Poly Dispositifs interchangeable, par héritage on peut substituer les fonctionnalités.

Diagramme de classe UML



Classes abstraites



Classes abstraites

```

public abstract class Shape {
    //Méthode abstraites
    public abstract double permet-
    er();
    public abstract double area();
}
    
```

Classe principale du projet

```

package demoappli;
public class DemoAppli {
    public static void main
    (String[] args) {
        //Instanciation de la classe
        Fenetre
        Fenetre ma fenetre =new
        Fenetre();
    }
}
    
```

Classe Fenêtre

```

package demoappli;
import javax.swing.JFrame;
public class Fenetre extends
JFrame{
    public Fenetre()
    {
        //Appel du constructeur par
        défaut de la classe JFrame
        super();
        //Titre de la fenetre
        this.setTitle("Mon application
        graphique");
        //Dimensions de ma Fenetre
        this.setSize(1200, 600);
        //Positionnement de ma Fenetre
        this.setLocationRelativeTo(-
        null);
        //Comportement de ma Fenetre à
        la fermeture
        this.setDefaultCloseOperation(-
        JFrame.DISPOSE_ON_CLOSE);
        //Créer un Panneau
        Panneau monPanneau =
        new Panneau();
        //Ajouter un Panneau au
        ContentPane
        this.setContentPane(m-
        onPanneau);
        //Ma Fenetre est visible
        this.setVisible(true)
    }
}
    
```

Aide Mémoire

Ajout de bouton

Border Layout

GridLayout

Box Layout

Boutons réactifs

Classe écoutant le bouton

Classe, constructeur

Classe, constructeur (cont)

```

public Ville (String nomVille,
int taille Ville, String canton-
Ville){
    nom=nomVille;
    ...
}
//Méthodes
void affiche (){
    System.out.println("Caractér-
    istiques de " + nom);
}
}
    
```

Classe panneau

```

package demoappli;
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JPanel;
public class Panneau extends
JPanel{
    //Constructeur
    public Panneau ()
    {
        //Appel du constructeur par
        défaut de la classe JPanel
        super();
    }
    //Redéfinition de la méthode
    paintComponent
    public void paintComponent
    (Graphics g){
        //Couleur du trait
        g.setColor(Color.blue);
        //Dessin des différentes formes
    }
}
    
```

```
public class Ville {
    //Attributs
    protected String nom;
    protected int taille;
    protected String canton;
    //Constructeur par défaut
    public Ville(){
        nom="";
        taille=0;
        ...
    }
    //Constructeur complémentaire
```



By **Alysee**
cheatography.com/alysee/

Not published yet.
Last updated 5th June, 2019.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>