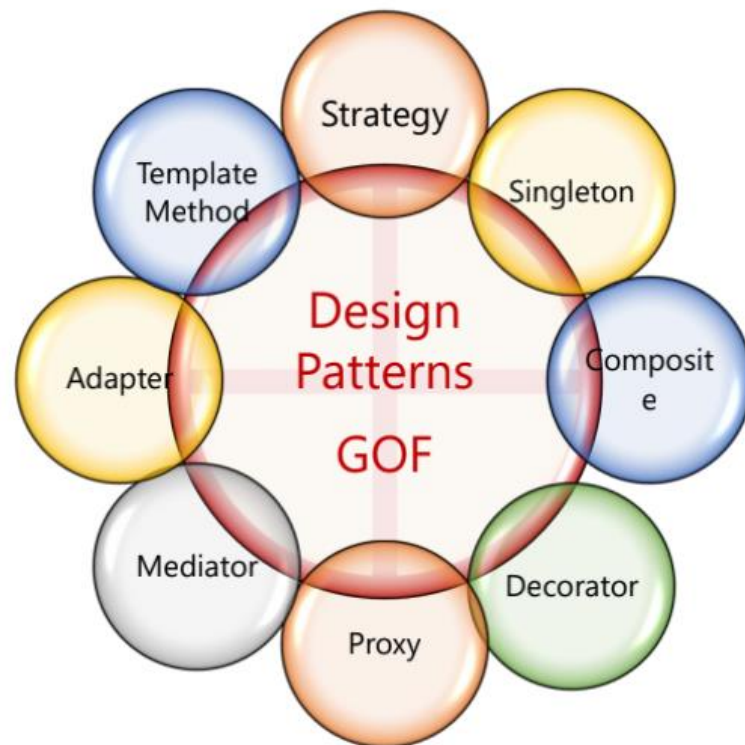


## State Pattern



Réalisé par : REGOUG Zakia

GLSID3

Année universitaire : 2023-2024

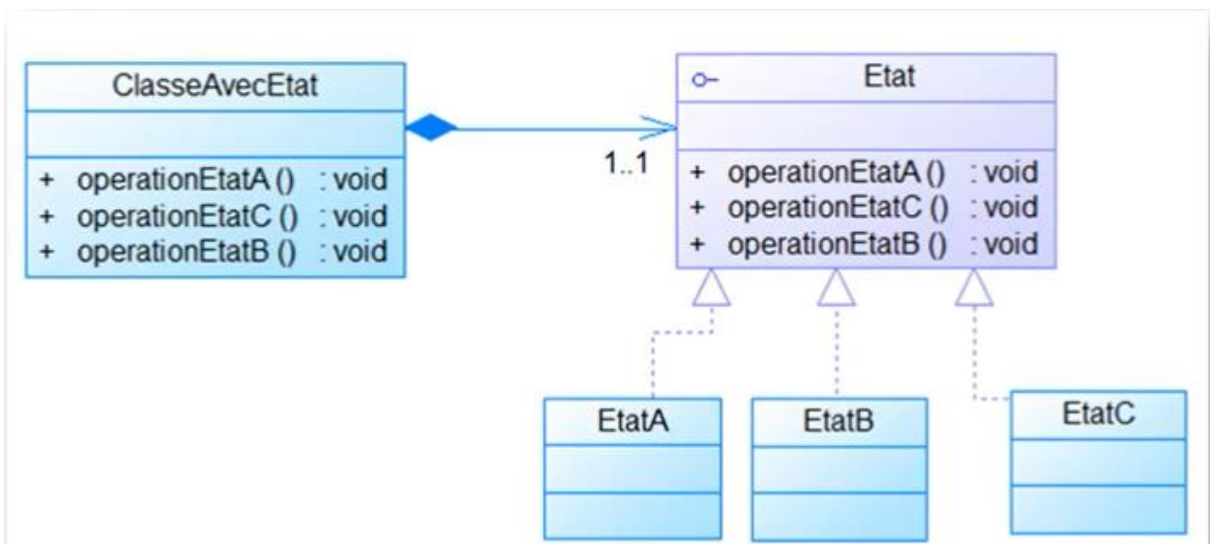
## Catégorie :

Comportement

## Objectif :

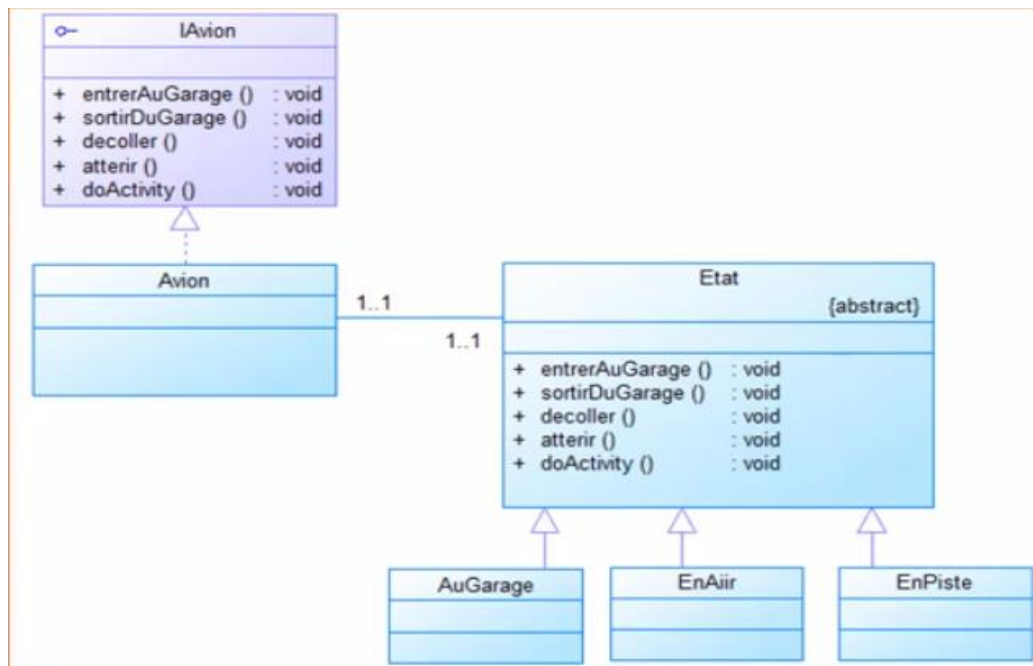
Changer le comportement d'un objet selon son état interne.

## Diagramme de classe :



## Exemple d'application :

- Un avion peut être dans l'un des trois états suivants :
  - soit dans le garage,
  - soit sur la piste,
  - soit en l'air.
- Lorsqu'il est dans le garage, la méthode **sortirDuGarage** permet de passer dans l'état "**sur la piste**".
- Lorsqu'il est sur la piste, la méthode **entrerAuGarage** permet de passer dans l'état "**dans le garage**".
- La méthode **décoller** permet de passer dans l'état "**en l'air**".
- Lorsqu'il est en l'air, la méthode **atterrir** permet de passer dans l'état "**sur la piste**".
- Les autres combinaisons Etats - Méthodes génèrent une erreur comme par exemple, invoquer décoller lorsque l'avion est dans le garage.



## Interface Avion :

```
package state;
```

```
public interface Avion {
    void entrezAuGarage();
    void sortirDuGarage();
    void decoller();
    void atterrir();
    void doActivity();
}
```

## Classe AvionImpl

```
package state;
```

```
public class AvionImpl implements Avion{
    private State state=new EtatEnPiste();
    @Override
    public void entrezAuGarage() {
        state.exentrezAuGarage();
    }

    @Override
    public void sortirDuGarage() {
        state.exsortirDuGarage();
    }

    @Override
    public void decoller() {
        state.exdecoller();
    }

    @Override
    public void atterrir() {
        state.exatterrir();
    }
}
```

```

@Override
public void doActivity() {
    state.exdoActivity();
}

abstract class State{
    abstract void exentrezAuGarage();
    abstract void exsortirDuGarage();
    abstract void exdecoller();
    abstract void exatterir();
    abstract void exdoActivity();
}

class EtatEnPiste extends State{

    @Override
    void exentrezAuGarage() {
        state=new EtatAuGarage();
        System.out.println("transition : piste => Garage");
    }

    @Override
    void exsortirDuGarage() {
        System.out.println("Impossible, Je suis deja en piste");
    }

    @Override
    void exdecoller() {
        state=new EtatEnAir();
        System.out.println("transition : piste => Air");
    }

    @Override
    void exatterir() {
        System.out.println("Impossible, je suis en piste");
    }

    @Override
    void exdoActivity() {
        System.out.println("En piste");
    }
}

class EtatAuGarage extends State{

    @Override
    void exentrezAuGarage() {
        System.out.println("Je suis deja au garage");
    }

    @Override
    void exsortirDuGarage() {
        state=new EtatEnPiste();
        System.out.println("transition : Garage => Piste");
    }

    @Override
    void exdecoller() {
        System.out.println("Impossible, je suis au garage");
    }
}

```

```

        @Override
        void exatterir() {
            System.out.println("Impossible, je suis au garage");
        }

        @Override
        void exdoActivity() {
            System.out.println("Au garage");
        }
    }
}

class EtatEnAir extends State{

    @Override
    void exentrezAuGarage() {
        System.out.println("Impossible je suis en Air");
    }

    @Override
    void exsortirDuGarage() {
        System.out.println("Impossible je suis en Air");
    }

    @Override
    void exdecoller() {
        System.out.println("Je suis deja en air");
    }

    @Override
    void exatterir() {
        state=new EtatEnPiste();
        System.out.println("transition : Air => Piste");
    }

    @Override
    void exdoActivity() {
        System.out.println("En Air");
    }
}

}

```

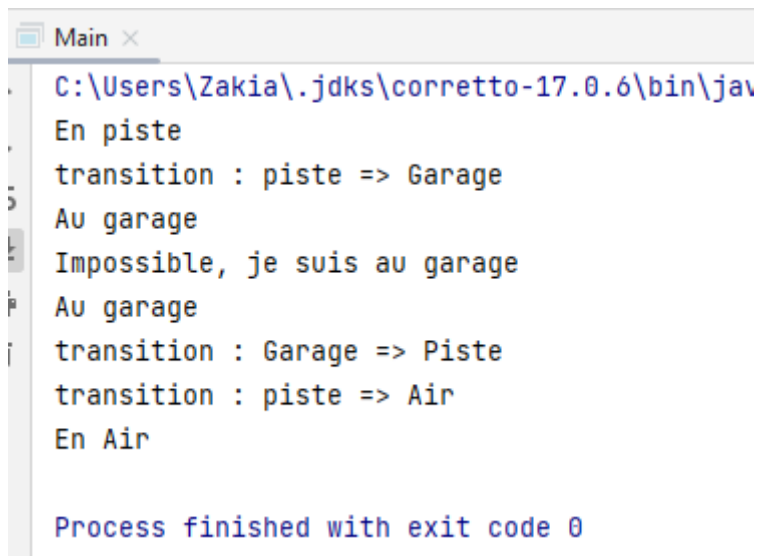
## Main

```

public class Main {
    public static void main(String[] args) {
        Avion avion=new AvionImpl();
        avion.doActivity();
        avion.entrezAuGarage();
        avion.doActivity();
        avion.atterir();
        avion.doActivity();
        avion.sortirDuGarage();
        avion.decoller();
        avion.doActivity();
    }
}

```

## Affichage :



The screenshot shows a console window titled 'Main' with a close button. The output text is as follows:

```
C:\Users\Zakia\.jdk\corretto-17.0.6\bin\jav
En piste
transition : piste => Garage
Au garage
Impossible, je suis au garage
Au garage
transition : Garage => Piste
transition : piste => Air
En Air

Process finished with exit code 0
```