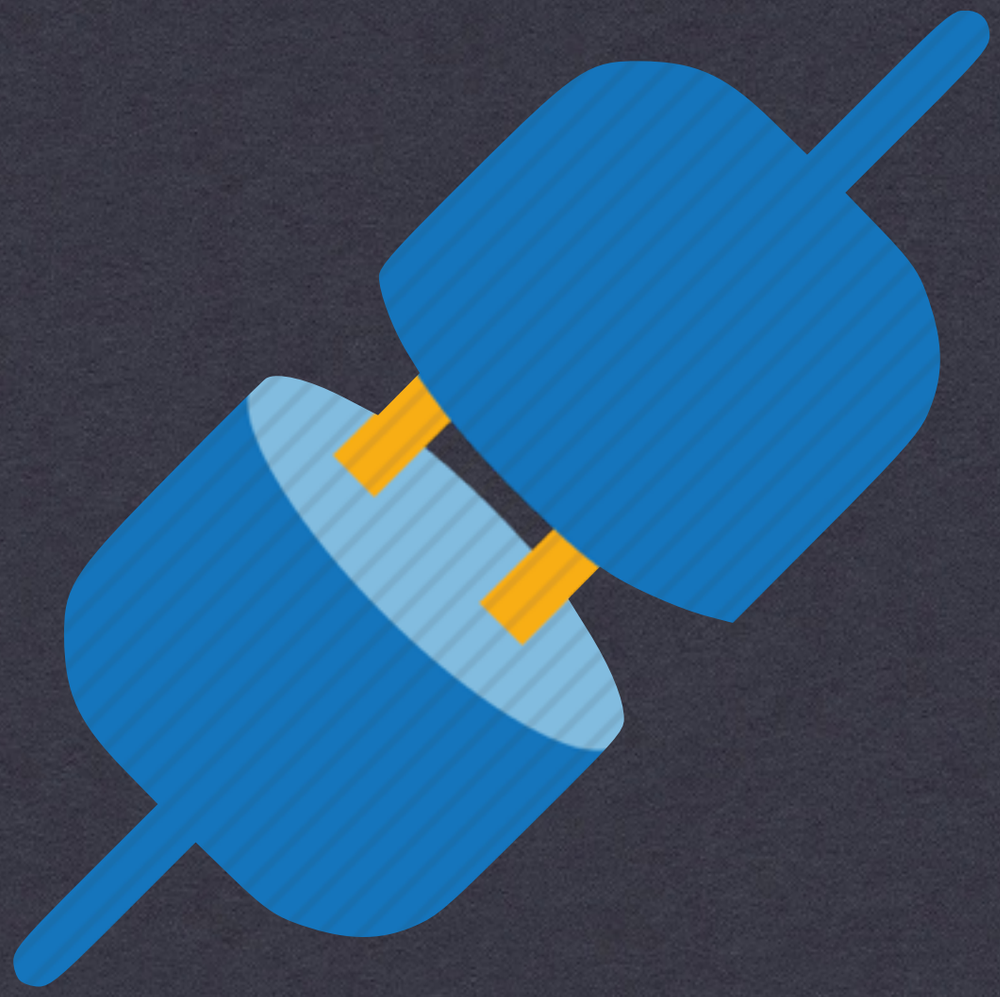


DESIGN PATTERN ADAPTER

COURS MAIL 26/01/2015

AXEL RICHIER

THERMOMETRE

En Celsius


```java
package designPattern;

public class ThermometreCelsius {

    double temperatureEnCelsius;

    public ThermometreCelsius() {
    }

    public double getTemperature() {
        return this.temperatureEnCelsius;
    }

    public void setTemperature(double temperatureEnCelsius) {
        this.temperatureEnCelsius = temperatureEnCelsius;
    }

}
```

← un attribut

← getter

← setter

THERMOMETRE

En Celsius


```java
package designPattern;

public class ThermometreCelsius {

    double temperatureEnCelsius;

    public ThermometreCelsius() {
    }

    public double getTemperature() {
        return this.temperatureEnCelsius;
    }

    public void setTemperature(double temperatureEnCelsius) {
        this.temperatureEnCelsius = temperatureEnCelsius;
    }

}
```

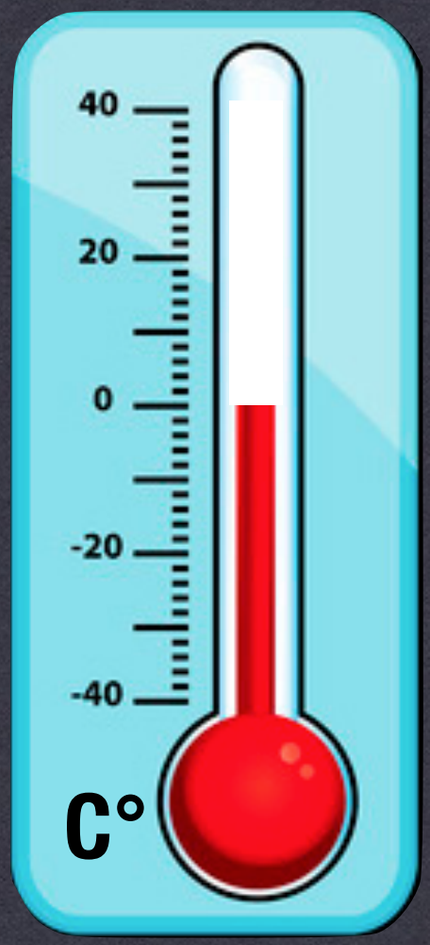
← un attribut

← getter

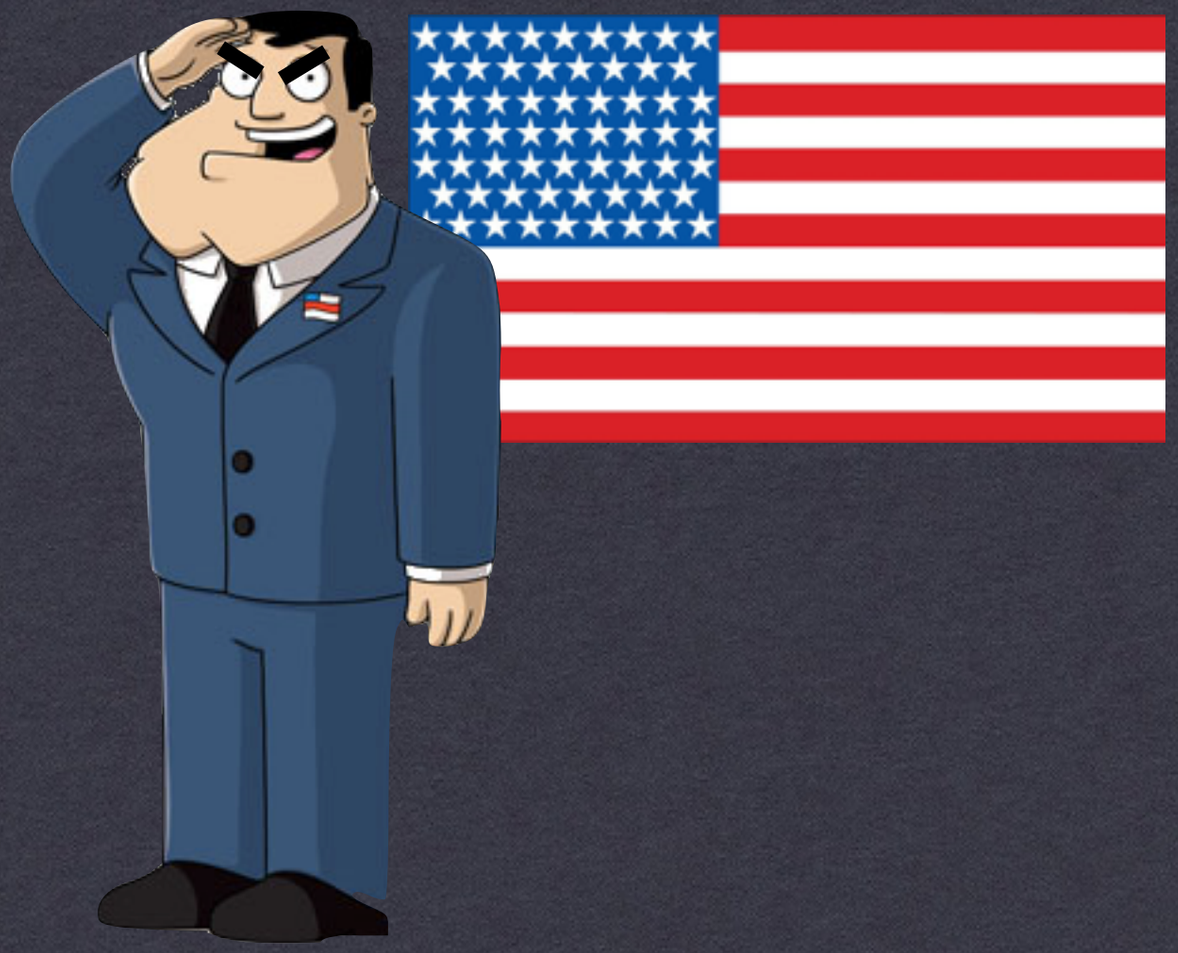
← setter

THERMOMETRE...

Non universel..

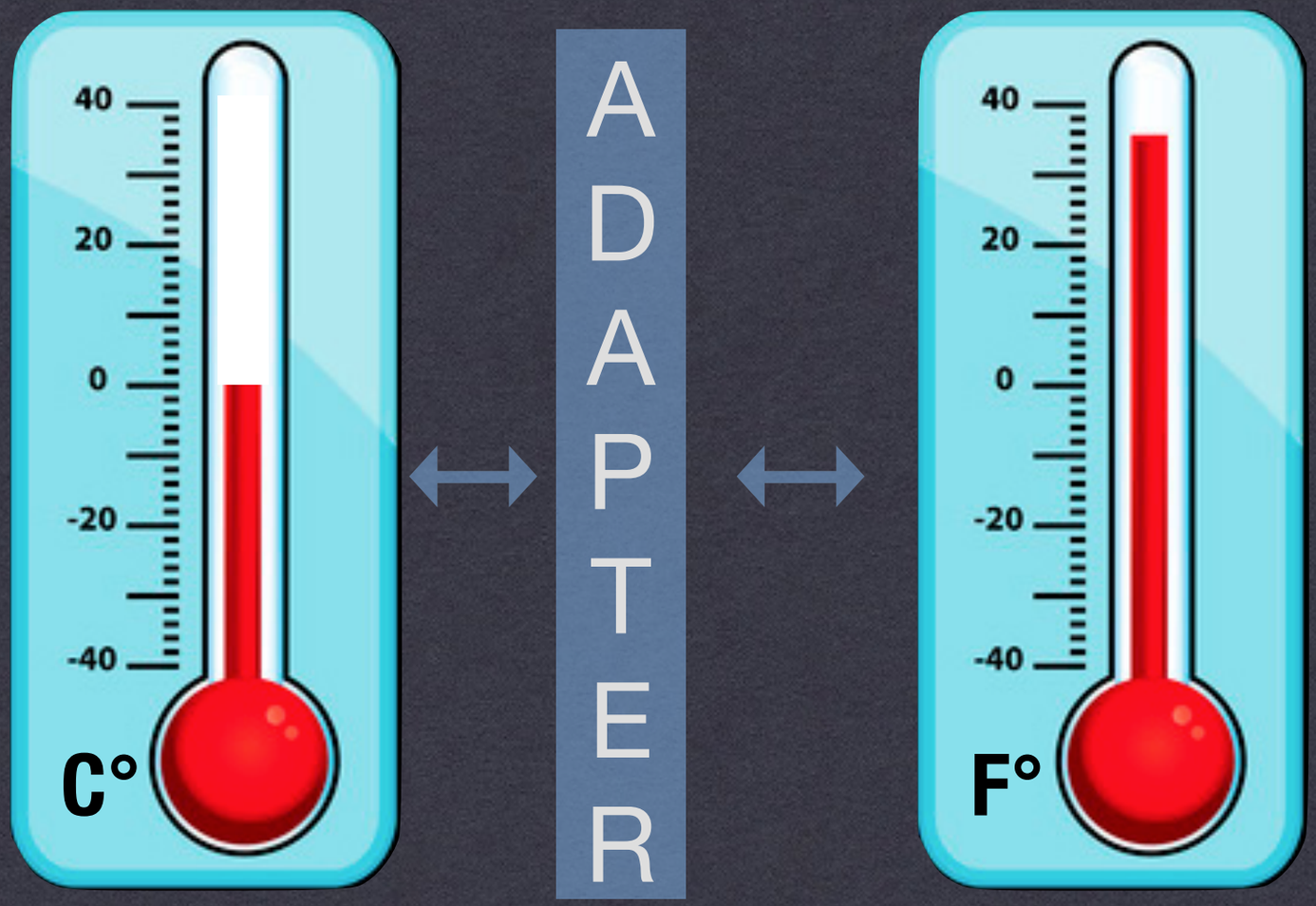


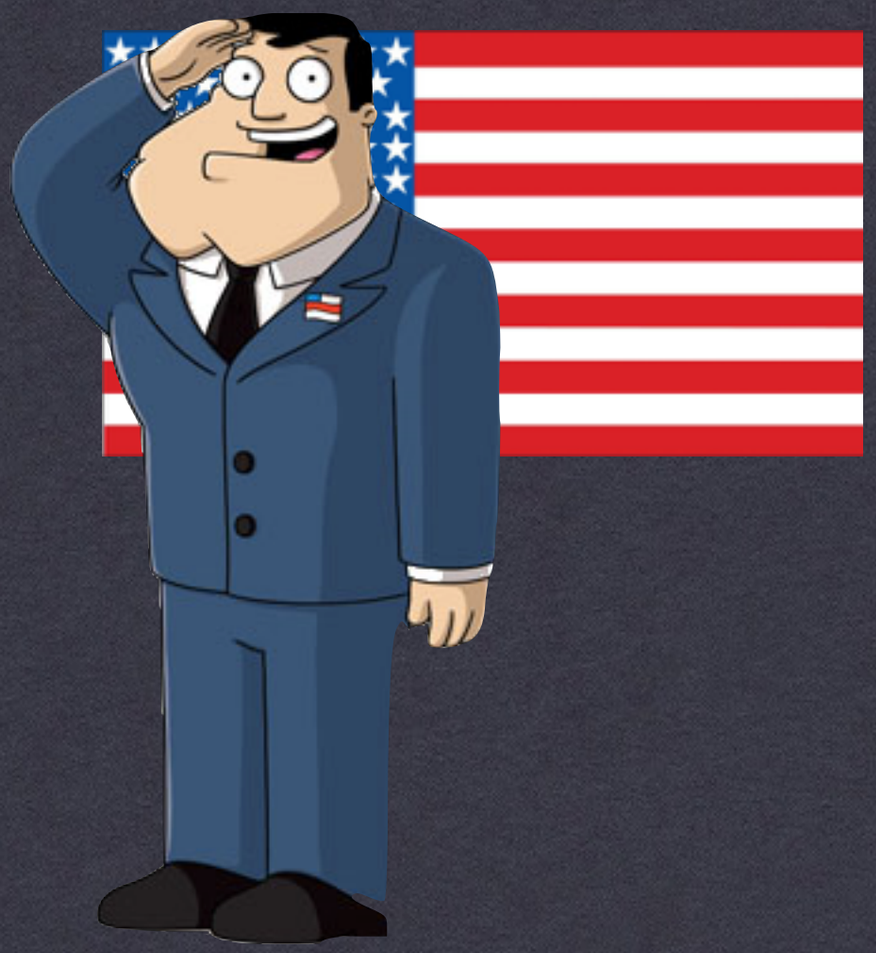
? ? ?



THERMOMETRE

Il faut trouver un moyen de convertir la température, sans changer la classe initiale, ni créer un nouveau thermomètre





THERMOMETRE

On crée une INTERFACE avec les méthodes d'affichage de la température en Celsius ET en Farenheit

```java
package designPattern;

public interface TemperatureInfo {

    public double getTemperatureEnFarenheit();

    public void setTemperatureEnFarenheit(double temperatureEnFarenheit);

    public double getTemperatureEnCelsius();

    public void setTemperatureEnCelsius(double temperatureEnCelsius);

}
```

THERMOMETRE

On crée une CLASSE implémentant l'interface.
Cette classe joue le rôle de l'adaptateur, et possède des méthodes pour convertir les Celsius en Farenheit (et inversement)

```java
package designPattern;
//classe Adapter
public class TemperatureAdaptateur extends ThermometreCelsius implements TemperatureInfo {
    @Override
    public double getTemperatureEnCelsius() {
        return temperatureEnCelsius;
    }
    @Override
    public double getTemperatureEnFarenheit() {
        return celsiusToFarenheit(temperatureEnCelsius);
    }
    @Override
    public void setTemperatureEnCelsius(double temperatureInC) {
        this.temperatureEnCelsius = temperatureInC;
    }
    @Override
    public void setTemperatureEnFarenheit(double temperatureInF) {
        this.temperatureEnCelsius = farenheitToCelsius(temperatureInF);
    }
    private double farenheitToCelsius(double f) {
        return ((f - 32) * 5 / 9);
    }
    private double celsiusToFarenheit(double c) {
        return ((c * 9 / 5) + 32);
    }
}
```

getter Celsius

getter Farenheit

setter Celsius

setter Farenheit

méthodes de conversion

THERMOMETRE

On peut instance UN SEUL THERMOMETRE et employer des méthodes en celsius ou en farenheit

```
TemperatureInfo Temperature = new TemperatureAdaptateur();

Temperature.setTemperatureEnCelsius(0);
Temperature.getTemperatureEnCelsius()); // Renvoie 0
Temperature.getTemperatureEnFarenheit()); // Renvoie 32


Temperature.setTemperatureEnFarenheit(0);
Temperature.getTemperatureEnCelsius()); // Renvoie -17.7777
Temperature.getTemperatureEnFarenheit()); // Renvoie 0
```

La classe initiale est intouchée et aucune classe thermometreFarenheit n’est nécessaire.