```java
import conversion.Conversion;
import conversion.Devise;
import conversion.DeviseExistanteException;
import conversion.DeviseInconnueException;

/*
 * Created on 20 sept. 2005
 *
 */

public class TestConversion {
	public static void main(String[] args) throws DeviseInconnueException, D
eviseExistanteException {
		double s = 100;
		Devise d = Conversion.devise("Franc");
		System.out.println(s + " " + d.nom() + " = " + d.conversionEnEuro
(s) + " Euros");
		Conversion.ajouterDevise("Dollard", 100);
	}
}
```



```java
package conversion;

import java.util.ArrayList;
import java.util.List;

/**
 * La classe Conversion permet de convertir une somme en Euro.
 */

public class Conversion {

	/*
	 * Cette version permet de rajouter dynamiquement de nouvelles devises.
	 *
	 * Certains problèmes de la version précédente sont toujours présents :
	 *
	 * utilisation de Error() plutôt qu'une exception ;
	 *
	 * problème d'efficacité de la méthode rechercheDevise(String) ;
	 *
	 * il n'existe que des devises à taux variable. Il est donc possible de
	 * modifier le taux de conversion, alors que ce devrait être parfois
	 * interdit.
	 */

	private static final String[] NOMS_DEVISES = { "Franc", "Mark", "Dollard",
			"Yen" };

	private static final double[] TAUX_CHANGE = { 6.55957, 1.95583, 1.29,
			139.67 };

	private static final List<Devise> DEVISES = new ArrayList<Devise>();

	static {
		for (int i = 0; i < NOMS_DEVISES.length; ++i) {
			try {
			ajouterDevise(NOMS_DEVISES[i], TAUX_CHANGE[i]);
			} catch (DeviseExistanteException e) {
					throw new InternalError(e.getMessage());
			}
		}
	}

	public static Devise devise(String nom) throws DeviseInconnueException {
		for (Devise d : DEVISES) {
			if (d.nom().equals(nom)) {
				return d;
			}
		}
		throw new DeviseInconnueException(nom);
	}

	public static boolean deviseTraitee(String nom) {
		try {
			devise(nom);
			return true;
		} catch (DeviseInconnueException e) {
			return false;
		}
	}

	public static void ajouterDevise(String nom, double tauxDeChange) throws
```

```
DeviseExistanteException {
                try {
                        Devise d = devise(nom);
                        throw new DeviseExistanteException(d);
                } catch (DeviseInconnueException e) {
                        DEVISES.add(new Devise(nom, tauxDeChange));
                }
        }

        public static Devise[] devisesTraitees() {
                return DEVISES.toArray(new Devise[DEVISES.size()]);
        }

        public static void main(String[] args) throws DeviseInconnueException {
                double valeurAConvertir = Double.valueOf(args[0]).doubleValue();
                String nomDevise = args[1];

                if (deviseTraitee(nomDevise)) {
                        System.out.println(devise(nomDevise).conversionEnEuro(
                                        valeurAConvertir)
                                        + " Euros");
                } else {
                        Devise[] devises = devisesTraitees();
                        System.out.print("Devise inconnue (devises traitées =");
                        for (int i = 0; i < devises.length; ++i) {
                                System.out.print(" " + devises[i].nom());
                        }
                        System.out.println(")");
                }
        }
}
```



```
package conversion;
/**
 * La classe Devise permet de créer des objets représentant la monnaie d'un
 * pays. Chaque devise est caractérisée par son nom et possède comme donnée son
 * taux de change en Euro. La classe fournit également des méthodes permettant
 * de convertir une somme depuis la monnaie vers l'Euro et inversement. Le taux
 * de change en Euro peut être modifié.
 */

public class Devise {
        private final String nom;

        private double tauxDeChangeEnEuro;

        Devise(String nom, double tauxDeChangeEnEuro) {
                this.nom = nom;
                this.tauxDeChangeEnEuro = tauxDeChangeEnEuro;
        }

        public String nom() {
                return nom;
        }

        public double tauxDeChangeEnEuro() {
                return tauxDeChangeEnEuro;
        }

        public void changerTauxDeChangeEnEuro(double nouveauTaux) {
                tauxDeChangeEnEuro = nouveauTaux;
        }

        public double conversionEnEuro(double somme) {
                return somme / tauxDeChangeEnEuro;
        }

        public double conversionDepuisEuro(double somme) {
                return somme * tauxDeChangeEnEuro;
        }

        public boolean equals(Object o) {
        return o instanceof Devise && nom.equals(((Devise)o).nom);
        }

        public String toString() {
                return "Devise " + nom + " (taux de change en Euro = "
                                + tauxDeChangeEnEuro + ")";
        }
}
```

```java
package conversion;

public class DeviseExistanteException extends Exception {

	/**
	 *
	 */
	private static final long serialVersionUID = 1L;

	private Devise d;

	public DeviseExistanteException(Devise d) {
		super("Devise " + d.nom() + " dM−^Nja existante");
		this.d = d;
	}

	public Devise devise() {
		return d;
	}

}
```



```java
package conversion;

public class DeviseInconnueException extends Exception {

	/**
	 *
	 */
	private static final long serialVersionUID = 1L;

	private String nom;

	public DeviseInconnueException(String nom) {
		super("Devise " + nom + " inconnue");
		this.nom = nom;
	}

	public String nom() {
		return nom;
	}

}
```