

```

/*-----*/
/*
-- NOM DU CSU (class)           : complexe.java
-- AUTEUR DU CSU                : P. Pignard
-- VERSION DU CSU              : 1.0a
-- DATE DE LA DERNIERE MISE A JOUR : 21 décembre 2001
-- ROLE DU CSU                 : bibliothèque de fonctions sur les complexes
--
--
-- FONCTIONS EXPORTEES DU CSU   :
--
--
-- FONCTIONS LOCALES DU CSU    :
--
--
-- NOTES                       : notation interne polaire
--
*/
/*-----*/

```

```

public class Complexe {

    protected double rho;
    protected double théta;

    public static final Complexe I    = new Complexe(1, Math.PI / 2);
    public static final Complexe UN  = new Complexe(1, 0);
    public static final Complexe ZÉRO = new Complexe(0, 0);

    protected double modPI (double V) {
        while (V > Math.PI)
            V = V - 2 * Math.PI;
        while (V < - Math.PI)
            V = V + 2 * Math.PI;
        return V;
    }

    public Complexe (double rho, double théta) {
        this.rho = rho;
        this.théta = modPI(théta);
    }

    public double argument () {
        return théta;
    }

    public double module () {
        return rho;
    }

    public double réelle () {

```

```

    return rho * Math.cos(théta);
}

public double imaginaire () {
    return rho * Math.sin(théta);
}

public Complexe conjugué () {
    return new Complexe(rho, - théta);
}

public Complexe opposé () {
    return new Complexe(rho, théta + Math.PI);
}

public Complexe produit (Complexe V) {
    return new Complexe(rho * V.module(), théta + V.argument());
}

public Complexe divise (Complexe V) {
    return new Complexe(rho / V.module(), théta - V.argument());
}

public Complexe racinne () {
    return new Complexe(Math.sqrt(rho), théta / 2);
}

public String toString () {
    return "(" + Double.toString(rho) + ", " + Double.toString(théta) + ")";
}

public Complexe cartésien (double réel, double imaginaire) {
    double théta = 0;
    double rho = 0;

    if ((imaginaire == 0) && (réel != 0)) {
        rho = réel;
        théta = (réel > 0) ? 0 : Math.PI;
    }
    if ((imaginaire != 0) && (réel == 0)) {
        rho = imaginaire;
        théta = (imaginaire > 0) ? Math.PI / 2 : -Math.PI / 2;
    }
    if ((imaginaire != 0) && (réel != 0)) {
        rho = Math.sqrt(réel * réel + imaginaire * imaginaire);
        théta = Math.atan(imaginaire / réel);
    }
    return new Complexe(rho, théta);
}

public Complexe somme (Complexe V) {

```

```
    return cartésien(réelle() + V.réelle(), imaginaire() + V.imaginaire());  
}  
  
public Complexe différence (Complexe V) {  
    return cartésien(réelle() - V.réelle(), imaginaire() - V.imaginaire());  
}  
  
public boolean equalTo (Complexe V) {  
    return (réelle() == V.réelle()) && (imaginaire() == V.imaginaire());  
}  
}
```