

```

package telephonie;
import telephonie.util.Date;

public class OperateurTarifVariable implements Operateur {

    private static final int TARIF_BAS = 15;
    private static final int TARIF_HAUT = 45;
    private static final int TARIF_MOYEN = 30;

    private int nbConnexionsActives;
    private int nbMaxConnexions;

    public OperateurTarifVariable(int nbMaxConnexions) {
        this.nbConnexionsActives = 0;
        this.nbMaxConnexions = nbMaxConnexions;
    }

    public int getDureeComptabilisee(Connexion c) {
        return c.dureeConnexion();
    }

    public int getTarifUnitaire(Connexion c) {
        int h = c.heureDebutConnexion();
        if ( h < 8 || h > 20) {
            return OperateurTarifVariable.TARIF_BAS;
        }
        else if ( h > 12 && h < 14) {
            return OperateurTarifVariable.TARIF_HAUT;
        }
        else {
            return OperateurTarifVariable.TARIF_MOYEN;
        }
    }

    public Connexion seConnecter(Date debut, ModeDePaiement m)
        throws OperateurSatureException, ModeDePaiementInvalideException {
        if (this.nbConnexionsActives == this.nbMaxConnexions) {
            throw new OperateurSatureException();
        }
        if (! m.valide()) {
            throw new ModeDePaiementInvalideException();
        }
        this.nbConnexionsActives = this.nbConnexionsActives + 1;
        return new Connexion(this, debut, m);
    }

    public void seDeconnecter(Connexion c, Date fin)
        throws PasDeConnexionException {
        if (c.getOperateur() != this) {
            throw new PasDeConnexionException();
        }
        c.finConnexion(fin);
        c.mode().debiter(this.getDureeComptabilisee(c), this.getTarifUnitaire(c));
        this.nbConnexionsActives = this.nbConnexionsActives - 1;
    }
}

```

```

package telephonie;
import telephonie.util.Date;

public class OperateurTarifFixe implements Operateur {

    private static final float COEFF_REDUCTION = (5f/6f);
    private static final int DUREE_AVANT_REDUCTION = 5;
    private static final int TARIF = 30;

    private int nbConnexionsActives;
    private int nbMaxConnexions;

    public OperateurTarifFixe (int nbMaxConnexions) {
        this.nbConnexionsActives = 0;
        this.nbMaxConnexions = nbMaxConnexions;
    }

    public int getDureeComptabilisee(Connexion c) {
        if (c.dureeConnexion() < OperateurTarifFixe.DUREE_AVANT_REDUCTION) {
            return c.dureeConnexion();
        }
        else {
            return (int) (c.dureeConnexion() * OperateurTarifFixe.COEFF_REDUCTION);
        }
    }

    public int getTarifUnitaire(Connexion c) {
        return OperateurTarifFixe.TARIF;
    }

}

public Connexion seConnecter(Date debut, ModeDePaiement m)
    throws OperateurSatureException, ModeDePaiementInvalideException {
    if (this.nbConnexionsActives >= this.nbMaxConnexions) {
        throw new OperateurSatureException();
    }
    if (! m.valide()) {
        throw new ModeDePaiementInvalideException();
    }
    this.nbConnexionsActives = this.nbConnexionsActives + 1;
    return new Connexion(this, debut, m);
}

public void seDeconnecter(Connexion c, Date fin)
    throws PasDeConnexionException {
    if (c.getOperateur() != this) {
        throw new PasDeConnexionException();
    }
    c.finConnexion(fin);
    c.mode().debiter(this.getDureeComptabilisee(c), this.getTarifUnitaire(c));
    this.nbConnexionsActives = this.nbConnexionsActives - 1;
}
}

```

```

package telephonie;
import telephonie.util.Date;

public class OperateurTarifXXXXX implements Operateur {

    private int nbConnexionsActives;
    private int nbMaxConnexions;

    public OperateurTarifXXXXX (int nbMaxConnexions) {
        this.nbConnexionsActives = 0;
        this.nbMaxConnexions = nbMaxConnexions;
    }

    public int getDureeComptabilisee(Connexion c) {

    }

    public int getTarifUnitaire(Connexion c) {

    }

    public Connexion seConnecter(Date debut, ModeDePaiement m)
        throws OperateurSatureException, ModeDePaiementInvalideException {
        if (this.nbConnexionsActives >= this.nbMaxConnexions) {
            throw new OperateurSatureException();
        }
        if (! m.valide()) {
            throw new ModeDePaiementInvalideException();
        }
        this.nbConnexionsActives = this.nbConnexionsActives + 1;
        return new Connexion(this, debut, m);
    }

    public void seDeconnecter(Connexion c, Date fin)
        throws PasDeConnexionException {
        if (c.getOperateur() != this) {
            throw new PasDeConnexionException();
        }
        c.finConnexion(fin);
        c.mode().debiter(this.getDureeComptabilisee(c), this.getTarifUnitaire(c));
        this.nbConnexionsActives = this.nbConnexionsActives - 1;
    }
}

```

```

package telephonie;
import telephonie.util.Date;

public class SuperOperateur implements Operateur {

    protected int nbConnexionsActives;
    protected int nbMaxConnexions;

    public SuperOperateur(int nbMaxConnexions) {
        this.nbConnexionsActives = 0;
        this.nbMaxConnexions = nbMaxConnexions;
    }

    public int getDureeComptabilisee(Connexion c) {
        // A PRECISER PLUS TARD
    }

    public int getTarifUnitaire(Connexion c) {
        // A PRECISER PLUS TARD
    }

    public Connexion seConnecter(Date debut, ModeDePaiement m)
        throws OperateurSatureException, ModeDePaiementInvalideException {
        if (this.nbConnexionsActives >= this.nbMaxConnexions) {
            throw new OperateurSatureException();
        }
        if (! m.valide()) {
            throw new ModeDePaiementInvalideException();
        }
        this.nbConnexionsActives = this.nbConnexionsActives + 1;
        return new Connexion(this, debut, m);
    }

    public void seDeconnecter(Connexion c, Date fin)
        throws PasDeConnexionException {
        if (c.getOperateur() != this) {
            throw new PasDeConnexionException();
        }
        c.finConnexion(fin);
        c.mode().debiter(this.getDureeComptabilisee(c), this.getTarifUnitaire(c));
        this.nbConnexionsActives = this.nbConnexionsActives - 1;
    }
}

```

```
package telephonie;
```

```
public class OperateurTarifVariable extends SuperOperateur {  
  
    private static final int TARIF_BAS = 15;  
    private static final int TARIF_HAUT = 45;  
    private static final int TARIF_MOYEN = 30;  
  
    public OperateurTarifVariable(int nbMaxConnexions) {  
        super (nbMaxConnexions) ;  
    }  
  
    public int getDureeComptabilisee(Connexion c) {  
        return c.dureeConnexion();  
    }  
  
    public int getTarifUnitaire(Connexion c) {  
        int h = c.heureDebutConnexion();  
        if ( h < 8 || h > 20) {  
            return OperateurTarifVariable.TARIF_BAS;  
        }  
        else if ( 12 < h && h < 14) {  
            return OperateurTarifVariable.TARIF_HAUT;  
        }  
        else {  
            return OperateurTarifVariable.TARIF_MOYEN;  
        }  
    }  
}
```

```
package telephonie;
```

```
public class OperateurTarifFixe extends Operateur {  
  
    private static final float COEFF_REDUCTION = 5/6;  
    private static final int DUREE_AVANT_REDUCTION = 5;  
    private static final int TARIF = 30;  
  
    public OperateurTarifFixe(int nbMaxConnexions) {  
        super(nbMaxConnexions);  
    }  
  
    public int getDureeComptabilisee(Connexion c) {  
        if (c.dureeConnexion() < OperateurTarifFixe.DUREE_AVANT_REDUCTION) {  
            return c.dureeConnexion();  
        }  
        else {  
            return (int) (c.dureeConnexion() * OperateurTarifFixe.COEFF_REDUCTION);  
        }  
    }  
  
    public int getTarifUnitaire(Connexion c) {  
        return OperateurTarifFixe.TARIF;  
    }  
}
```

```

package telephonie;
import telephonie.util.Date;

public abstract class AbstractOperateur implements Operateur {

    protected int nbConnexionsActives;
    protected int nbMaxConnexions;

    public AbstractOperateur(int nbMaxConnexions) {
        this.nbConnexionsActives = 0;
        this.nbMaxConnexions = nbMaxConnexions;
    }

    // déclarations en fait non nécessaires car induites par l'interface
    public abstract int getDureeComptabilisee(Connexion c);
    public abstract int getTarifUnitaire(Connexion c);

    public Connexion seConnecter(ModeDePaiement m)
        throws OperateurSatureException, ModeDePaiementInvalideException {
        if (this.nbConnexionsActives == this.nbMaxConnexions) {
            throw new OperateurSatureException();
        }
        if (! m.valide()) {
            throw new ModeDePaiementInvalideException();
        }
        this.nbConnexionsActives = this.nbConnexionsActives + 1;
        return new Connexion(this,m);
    }

    public void seDeconnecter(Connexion c) throws PasDeConnexionException {
        if (c.getOperateur() != this) {
            throw new PasDeConnexionException();
        }
        c.mode().debiter(this.getDureeComptabilisee(c),this.getTarifUnitaire(c));
        this.nbConnexionsActives = this.nbConnexionsActives - 1;
    }
}

```