

```

/**
 *
 * @author Dominique Bouthinon
 * Voiture avec ABS
 */
public class Abs extends Freinage
{
    private static final int prixABS = 2000 ;

    /**
     * initialise Abs avec uneVoiture
     */
    public Abs(Voiture uneVoiture)
    {
        super(uneVoiture) ;
    }

    /**
     * retourne le prix de la Voiture avec ABS
     */
    public int prix()
    {
        return Abs.prixABS + this.getVoiture().prix();
    }

    /**
     * retourne le String decrivant la voiture avec ABS
     */
    public String toString()
    {
        return this.getVoiture() + " + ABS (" + Abs.prixABS
+ "€)";
    }
}
// fin classe Abs
import java.util.ArrayList;

/**
 * @author Dominique Bouthinon
 * gere une cours de voitures (Motorisation)
 */
public class Course
{
    private ArrayList<Motorisation> voitures ;

    /**
     *
     */

```

```

public Course()
{
    this.voitures = new ArrayList<Motorisation>() ;
}

/**
 * ajoute une VoitureMotorisee
 */
public void inscrireVoitureMotorisee(Motorisation voiture)
{
    this.voitures.add(voiture) ;
}

/**
 * gere une course d'une duree d secondes
 */
public void simulerCourse(float d)
{
    System.out.println("\n\nla course est partie !") ;

    for (int t = 1 ; t <= d ; t++)
    {
        System.out.println(t + "
seconde(s) :)") ;

        for (int i = 0 ; i <
this.voitures.size() ; i ++ )
        {
            Motorisation m =
this.voitures.get(i) ;
            float distance =
m.distanceParcourue(t) ;

            System.out.println(m.getModele()
+ " a parcouru " + distance + "km") ;
        }
    }

    System.out.println("la course est finie") ;
}
}
//fin class Course

/**
 * @author Dominique Bouthinon
 * Voiture avec moteur Diesel
 */

```

```

public class Diesel extends Motorisation
{
    private static final int prixMotorisationDiesel = 1500 ;

    /**
     * initialise Diesel avec uneVoiture
     */

    public Diesel(Voiture uneVoiture, float uneMasseMoteur,
float uneForceMoteur)
    {
        super(uneVoiture, uneMasseMoteur, uneForceMoteur) ;

    }

    /**
     * retourne le prix de la Voiture motorisee en essence
     */
    public int prix()
    {
        return Diesel.prixMotorisationDiesel +
this.getVoiture().prix();
    }

    /**
     * retourne le String decrivant la Motorisation diesel
     */
    public String toString()
    {
        return this.getVoiture() + super.toString() + "
diesel (" + Diesel.prixMotorisationDiesel + "€)";
    }
}
// fin classe Essence

/**
 *
 * @author Dominique Bouthinon
 * Voiture avec moteur a essence
 */

public class Essence extends Motorisation
{
    private static final int prixMotorisationEssence = 1000 ;

    /**
     * initialise Essence avec uneVoiture
     */

    public Essence(Voiture uneVoiture, float uneMasseMoteur,
float uneForceMoteur)
    {

```

```

        super(uneVoiture, uneMasseMoteur, uneForceMoteur) ;
    }

    /**
     * retourne le prix de la Voiture motorisee en essence
     */
    public int prix()
    {
        return Essence.prixMotorisationEssence +
this.getVoiture().prix();
    }

    /**
     * retourne le String decrivant la Motorisation essence
     */
    public String toString()
    {
        return this.getVoiture() + super.toString() + "
essence (" +
        Essence.prixMotorisationEssence + "€)";
    }
}
// fin classe Essence

/**
 * @author Dominique Bouthinon
 *
 * gere une Ferrari avec de multiples options
 */
public class Ferrari extends Voiture
{
    /**
     * initialise la Ferrari avec unModele
     */
    public Ferrari(int unNumero, String unModele, float
uneMasse)
    {
        super(unNumero, unModele, uneMasse) ;
    }

    /**
     *
     * @return le prix de la Ferrari
     */
    public int prix()
    {
        return 250000;
    }
}
// fin classe Voiture

```

```

/**
 * @author Dominique Bouthinon
 *
 * Decorateur de la classe Voiture gerant le freinage
 */
public abstract class Freinage extends Voiture
{
    private Voiture v ;
    // la Voiture a decorer avec la motorisation

    /**
     * Initialise la Motorisation avec
     */
    public Freinage(Voiture uneVoiture)
    {
        super() ;
        this.v = uneVoiture ;
    }

    /**
     * Redefinition de prix pour obliger les classes
     filles a donner
     * les instructions adequates
     */
    public abstract int prix() ;

    /**
     * retourne la Voiture qui est decoree
     */
    public Voiture getVoiture()
    {
        return v ;
    }

    public String getModele()
    {
        return this.getVoiture().getModele() ;
    }
}

/**
 *

```

```

 * @author Dominique Bouthinon
 * Voiture avec freinage magnetique
 */

public class FreinageMagnetique extends Freinage
{
    private static final int prixFreinageMagnetique = 2000 ;

    /**
     * initialise Abs avec uneVoiture
     */
    public FreinageMagnetique(Voiture uneVoiture)
    {
        super(uneVoiture) ;
    }

    /**
     * retourne le prix de la Voiture avec ABS
     */
    public int prix()
    {
        return FreinageMagnetique.prixFreinageMagnetique +
this.getVoiture().prix();
    }

    /**
     * retourne le String decrivant la voiture avec ABS
     */
    public String toString()
    {
        return this.getVoiture() + " + freinage magnetique
(" + FreinageMagnetique.prixFreinageMagnetique + "€)";
    }

    public String getModele()
    {
        return this.getVoiture().getModele() ;
    }
}
// fin classe Freinage Magnetique

/**
 * @author Dominique Bouthinon
 *
 * Decorateur de la classe Voiture gerant la motorisation
 */
public abstract class Motorisation extends Voiture
{
    private float masseMoteur ;
    private float forceMoteur ;

```

```

private Voiture v ;
// la Voiture a decorer avec la motorisation

/**
 * Initialise la Motorisation avec
 */

public Motorisation(Voiture uneVoiture, float
uneMasseMoteur, float uneForceMoteur)
{
    super() ;
    this.masseMoteur = uneMasseMoteur ;
    this.forceMoteur = uneForceMoteur ;
    this.v = uneVoiture ;
}

/**
 * Redefinition de prix pour obliger les classes
filles a donner
 * les instructions adequates
 */
public abstract int prix() ;

/**
 *
 */
public float getMasseMoteur()
{
    return this.masseMoteur ;
}

/**
 *
 */
public float getForceMoteur()
{
    return this.forceMoteur ;
}

/**
 *
 */
public float getMasse()
{
    return this.getMasseMoteur() +
this.getVoiture().getMasse() ;
}

```

```

/**
 * retourne la Voiture qui est decoree
 */
public Voiture getVoiture()
{
    return v ;
}

/**
 *
 */
public float acceleration()
{
    return this.getForceMoteur() /
(this.getVoiture().getMasse() + this.getMasseMoteur()) ;
}

/**
 * distance parcourue au temps t
 */
public float distanceParcourue(float t)
{
    return 0.5f * this.acceleration() * t * t ;
}

/**
    retourne le modele de la voiture de base
("interne")
 */
public String getModele()
{
    return this.getVoiture().getModele() ;
}

/**
 * retourne la description de la motorisation
 */
public String toString()
{
    return " moteur : poids = " + this.getMasseMoteur()
+ "t, force = " +
    this.getForceMoteur() + "N" ;
}
}

```



```

        this.masseVoiture = v.getMasse() ;
    }

    /**
     * initialise la Voiture avec unModele
     */
    public Voiture(int unNumero, String unModele, float
uneMasseVoiture)
    {
        this.numero = unNumero ;
        this.modele = unModele ;
        this.masseVoiture = uneMasseVoiture ;
    }

    /**
     *
     * @return le prix de la Voiture
     */
    public abstract int prix() ;

    /**
     * retourne le numero de la voiture
     */
    public int getNumero()
    {
        return this.numero ;
    }

    /**
     * retourne le modele de la voiture
     */
    public String getModele()
    {
        return this.modele ;
    }

    /**
     * retourne la masse de la voiture */
    public float getMasse()
    {
        return this.masseVoiture ;
    }

    /**
     * retourne le String decrivant la Voiture
     */
    public String toString()
    {
        return "modele = " + this.getModele() + " n°" +

```

```

this.getNumero() + " poids = " + this.getMasse() + "t (" +
        this.prix() + "€)" ;
    }
}
// fin classe Voiture

```