

Soit le code suivant :

```
public class Pair {
    public double x;
    public double y;
    public Pair( double x, double y ){
        this . x = x;
        this . y = y;
    }
    public abstract double longueur();
}
```

```
public class Point extends Pair {
    public Point( double x, double y ){}
    public double longueur(){ return 0.0; }
}
```

```
public class Vecteur extends Pair {
    public Vecteur( double x, double y ){}
    public double longueur(){ return Math.sqrt( x * x + y * y ); }
}
```

```
1: public class TestPair {
2:     public static void main( String [] argv ){
3:         Pair t[] = new Pair[3];
4:         t[0] = new Point( 1.0, 2.0 );
5:         t[1] = new Pair( 0.0, 1.0 );
6:         t[2] = new Vecteur( 0.0, 0.0 );
7:     }
8: }
```

1. Quel(s) déclaration de classe devrait(ent) utiliser 'abstract' :

`Pair`

2. Quel(s) ligne(s) du main donne(nt) une erreur (en corrigeant les 'abstract' du numéro 1) :

`5 : new sur une classe abstraite.`

3. Écrivez le code pour le constructeur de Point :

`super(x, y);`

4. Écrivez le code pour la méthode `toString` de la classe `Point`. Cette méthode doit construire une `String` plaçant les `x`, `y` du point entre parenthèse et séparé d'une virgule. Par exemple : `(4, 2.5)`.

`return "(" + x + ", " + y + ")";`