

Master 2 LITL - Programmation pour le TAL

Java : visibilité des variables, valeurs et références

Franck Sajous/CLLE-ERSS



<http://fsajous.free.fr/>

Variables locales

```
public class TestVisibilite
{
    public void faitUnTruc ()
    {
        String maChaine = "truc";
        System.out.println(maChaine);
    }

    public void faitUnMachin ()
    {
        System.out.println(maChaine);
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.faitUnTruc();
        tv.faitUnMachin();
    }
}
```

Variables locales

```
public class TestVisibilite
{
    public void faitUnTruc ()
    {
        String maChaine = "truc";
        System.out.println(maChaine);
    }

    public void faitUnMachin ()
    {
        System.out.println(maChaine);
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.faitUnTruc();
        tv.faitUnMachin();
    }
}
```

**maChaine cannot be
resolved to a variable**

Déclaration de variable et masquage de visibilité

```
public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc ()
    {
        maChaine = "truc";
    }

    public void afficheChaine ()
    {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine();
        tv.faitUnTruc();
        tv.afficheChaine();
    }
}
```

Déclaration de variable et masquage de visibilité

```
public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc ()
    {
        maChaine = "truc";
    }

    public void afficheChaine ()
    {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine(); → affiche : machin
        tv.faitUnTruc();
        tv.afficheChaine(); → affiche : truc
    }
}
```

Déclaration de variable et masquage de visibilité

```
public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc () public void faitUnTruc ()
    {                          {
        maChaine = "truc";    String maChaine = "truc";
    }                          }

    public void afficheChaine ()
    {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine(); → affiche : machin
        tv.faitUnTruc();
        tv.afficheChaine(); → affiche : truc
    }
}
```

Déclaration de variable et masquage de visibilité

```
public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc () public void faitUnTruc ()
    {                          {
        maChaine = "truc";    String maChaine = "truc";
    }                          }

    public void afficheChaine ()
    {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine();
        tv.faitUnTruc();
        tv.afficheChaine();
    }
}
```

→ affiche : machin

→ affiche : ~~true~~ machin

Déclaration de variable et masquage de visibilité

```

public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc () {
        maChaine = "truc";
    }

    public void afficheChaine () {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine();
        tv.faitUnTruc();
        tv.afficheChaine();
    }
}

```


affiche : machin
affiche : ~~true~~ machin

Déclaration de variable et masquage de visibilité

```

public class TestVisibilite
{
    private String maChaine;

    public void faitUnTruc ()
    {
        maChaine = "truc";
    }

    public void afficheChaine ()
    {
        System.out.println("maChaine = " + maChaine);
    }

    public void setChaine (String str)
    {
        maChaine = str;
    }

    public static void main(String[] args)
    {
        TestVisibilite tv = new TestVisibilite();
        tv.setChaine("machin");
        tv.afficheChaine();
        tv.faitUnTruc();
        tv.afficheChaine();
    }
}

```

Duplicate local variable maChaine

~~```

 String maChaine = "truc";
 public void faitUnTruc ()
 {
 String maChaine = "truc";
 String maChaine = "chinma";
 }

```~~

→ **affiche : machin**

→ **affiche : ~~true~~ machin**

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);

PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());

point1.setX(5);
System.out.println(point2.toString());
```

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());

point1.setX(5);
System.out.println(point2.toString());
```

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)

**a**  
**5**

# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

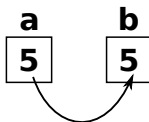
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)





# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

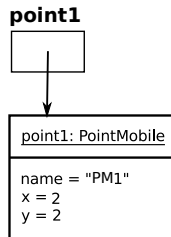
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

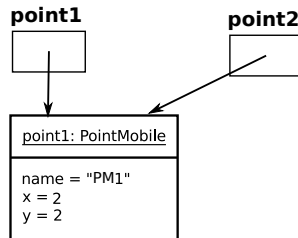
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

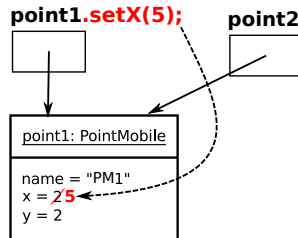
—————→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

—————→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

—————→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

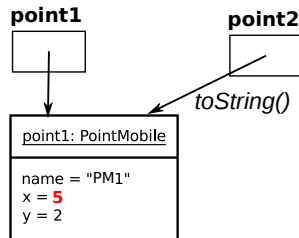
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

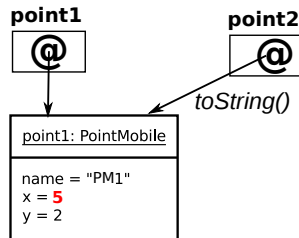
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



# Valeurs et références

```
int a = 5;
int b = a;
a = 3;
System.out.println(b);
```

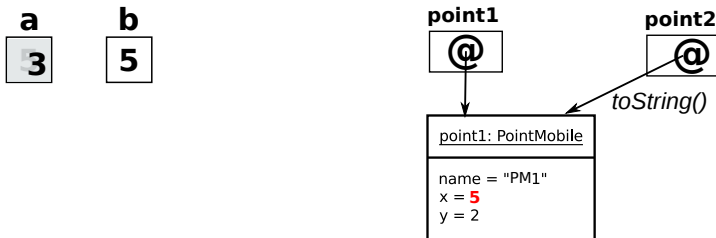
→ affiche : 5

```
PointMobile point1 = new PointMobile(2, 2);
point1.setNom("PM 1");
PointMobile point2 = point1;
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (2, 2)

```
point1.setX(5);
System.out.println(point2.toString());
```

→ affiche : [PointMobile-PM 1] (5, 2)



Variables types primitifs : valeur/Variables objets : références

# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```



# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```

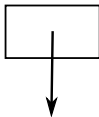
***NullPointerException !***

# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```

***NullPointerException !***

**point1**

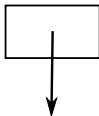


# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```

***NullPointerException !***

**point1**



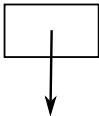
```
PointMobile point1; ⇔ PointMobile point1 = null;
```

# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```

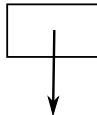
***NullPointerException !***

**point1**



```
PointMobile point1;
```

**point1**

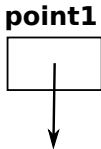


`PointMobile point1; ⇔ PointMobile point1 = null;`

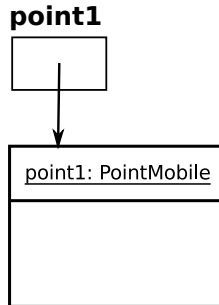
# C'est null

```
PointMobile point1;
point1.setNom("PM 1");
```

***NullPointerException !***



```
PointMobile point1;
point1 = new PointMobile();
```

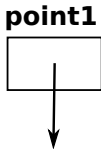


`PointMobile point1;`  $\Leftrightarrow$  `PointMobile point1 = null;`

# C'est null

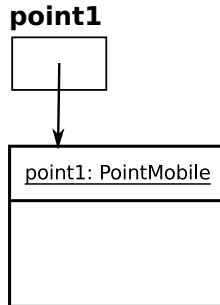
```
PointMobile point1;
point1.setNom("PM 1");
```

***NullPointerException !***



```
PointMobile point1;
point1 = new PointMobile();
point1.setNom("PM 1");
```

***OK***



`PointMobile point1; ⇔ PointMobile point1 = null;`