**Null, Exceptions, Valeurs par défaut**

**Null**

**Null =** valeur particulière qu’on attribue à une variable de type référence signifiant que la variable ne référence aucun objet

|  |
| --- |
| Cercle |
| - rayon : double |
| + Cercle (rayon :double)  + getRayon():double  + toString():String |

**Cercle cer = null;   
cer = new Cercle(1.5);**

|  |
| --- |
| **cer : Cercle** |
| **rayon = 1.5** |

**cer**

**valeur par défaut**

la valeur par défaut en java est :

* **Numérique** = 0
* **Booléen** = false
* **Caractère** = ‘/U0000
* **Type référence** = null

**Les variable locale (déclaré dans une méthode) ne reçoivent pas de valeur par défaut**

**Exception**

* Objet spécial permettant de signaler une erreur
* Construit pendant l’éxécution si l’érreur se produit
* Nombreux types d’exceptions existant en java (IllegalArgumentException, NullPointerException,…)

Throw permet de lancer une exception lorsqu’un problème est détecté



public class Cercle{

private

double rayon;

public Cercle(double rayon) {

if (rayon < =0)

throw

new

IllegalArgumentException

(

"Rayon

invalide");

this.rayon

=

rayon;

}

…

}

Création d’un objet de la classe

IllegalArgumentException

Lancer une

exception

**Il est possible de traiter plusieurs erreurs reçues comme ceci :**



public class

CerclePlace

{

private

double rayon;

private

Point centre;

public

CerclePlace

(

double

rayon,Point

centre){

if (rayon <=0)

throw

new

IllegalArgumentException

(

"Rayon invalide");

if (centre ==

null

)

throw

new

NullPointerException

(

"Le centre ne peut être

null

");

this.rayon

=

rayon;

this.centre

centre;

=

}

…

}

public class

TestCerclePlace

{

public

static

void

main(String[]

args

){

**try**

**{**

CerclePlace

cercle = new

CerclePlace

(3.0

,null);

System.out.println

(

cercle.getRayon

;

())

**catch**

**(**

**}**

**IllegalArgumentException**

**e){**

//traitement 1

**}**

**catch**

**(**

NullPointerException

e) {

//traitement 2

**}**

}

}