## Annexe 11– Exceptions contrôlées

## En résumé ( exceptions non – contrôlées )

* Les objets, **provenant de sous-classes de RuntimeException**, se lancent tout seuls
* Bloc try/catch/finally pour régler les problèmes

## Exceptions contrôlées / Exceptions créées par le programmeur

**Ne sont pas des sous-classes de RuntimeException**

* IOException : exception d’entrée / sortie pour des fichiers externes
* FileNotFoundException : exception si le fichier à accéder n’est pas trouvé / n’existe pas
* SQLException : problème avec l’accès aux tables SQL
* Nos propres classes d’Exception : pour gérer des situations inhabituelles propres à un projet

Exemple : Dans l’exercice Placement de l'annexe 8, gérer à l’aide d’une structure try /catch les montants inférieurs à 0 $

Où lancera-t-on cette exception ? dans le constructeur de placement (modèle) dans le constructeur des objets placement car ca faut une partie de la logique de l’app mentionner qu’on accepte pas les montant négatif

Où la captera-t-on ? dans l’activité car c’est là qu’on communique avec l’usager

Pour les gérer, vous devez absolument faire **une des deux actions** suivantes à ch. étape de la pile d’appel:

1. Faire une structure try/catch

OU

1. Lance explicitement l’exception dans la pile d’appel pour qu’elle soit captée plus tard c’est faire un throw dans l’entête de la méthode qui est susceptible de lancer l’exception

## Exercice : trace : 2-3-4-1-6-8-9

public class ExempleException {

public static void main ( String [] args ){

ExempleException ex = new ExempleException();

try{

ex.lancerException();

}

catch ( Exception exception ){

System.out.println ( " 1. Exception traitée par le main " );

}

ex.nePasLancerException();

}

public void lancerException () throws Exception {

try{

System.out.println ( "2. Méthode lancerException " );

throw new Exception();

}

catch ( Exception exception ){

System.out.println ( "3.Exception traitée dans lancerException");

throw exception;

}

finally{

System.out.println ("4. finally dans lancerException");

}

}

public void nePasLancerException (){

try{

System.out.println ("6.Méthode ne pas lancerException");

}

catch(Exception exception ){

System.out.println ("7. Exception dans nePasLancerException");

}

finally{

System.out.println ("8. Finally dans nePasLancerException");

}

System.out.println ("9. Fin de la méthode nePasLancerException" );

}

}