

# Fondamentaux du développeur

Séance 5 - Introduction au débogage - Types d'erreurs et concepts clés

Philippe ROUSSILLE



## Section 1

Qu'est-ce que le débogage ?

# Qu'est-ce que le débogage ?

- Le débogage est le processus systématique qui permet d'identifier la cause réelle d'un comportement incorrect dans un programme.
- Il est essentiel pour garantir que votre code fonctionne comme prévu dans toutes les situations.
- Un bon débogueur apprend non seulement à corriger un bug, mais aussi à comprendre pourquoi il est apparu et comment éviter qu'il se reproduise.

## Section 2

### Typologie des erreurs en programmation

# Typologie des erreurs en programmation

- **Erreurs de syntaxe** : fautes dans la grammaire ou la structure du code, détectées avant exécution (ex. parenthèse manquante).
- **Erreurs d'exécution** : erreurs déclenchées pendant l'exécution du programme (ex. division par zéro).
- **Erreurs de logique** : le programme s'exécute sans erreur mais produit un résultat incorrect.
- **Erreurs de contexte** : erreurs dues à des facteurs externes (ex. fichier inexistant, API indisponible).
- **Erreurs de programmation** : violation d'une précondition, souvent détectées via la programmation défensive.

# Les erreurs de syntaxe

- Elles empêchent le programme de s'exécuter : parenthèses non fermées, mots-clés erronés, identifiants incorrects.
- Elles sont détectées facilement car les IDE les signalent avant l'exécution.
- Elles sont généralement les plus simples à corriger.

# Les erreurs d'exécution

- Le code compile mais s'arrête avec une exception (ex. `NullPointerException`, dépassement d'index).
- Elles surviennent à cause de conditions imprévues.
- Un débogueur permet de localiser la ligne précise où elles apparaissent.

# Les erreurs de logique

- Aucune exception n'est levée mais le résultat est incorrect (algorithme mal conçu, mauvaise condition).
- Souvent les plus difficiles à repérer.
- Elles nécessitent une bonne compréhension du problème et une observation minutieuse.

# Les erreurs de contexte

- Fichier introuvable, serveur indisponible, données mal formatées.
- Elles nécessitent une gestion d'erreur robuste (try-catch, validations).
- Il est important de les anticiper lors de la conception.

## Section 3

### Pourquoi débogue-t-on mal ?

# Pourquoi débogue-t-on mal ?

- On modifie le code trop vite sans comprendre le problème.
- On tente plusieurs corrections à la fois, rendant l'origine du bug impossible à identifier.
- On ne reproduit pas le bug correctement avant de le corriger.
- On observe seulement une variable au lieu du flux global.

## Section 4

### Méthodologie CLEAR pour un débogage efficace

# Méthodologie CLEAR pour un débogage efficace

- **Comprendre** : reproduire le bug de manière fiable.
- **Localiser** : réduire la zone suspecte (binary search, breakpoints).
- **Examiner** : analyser les variables, conditions, pile d'appels.
- **Agir** : modifier une seule hypothèse à la fois.
- **Retester** : vérifier la correction et tester les cas limites.

## Section 5

### Concepts clés du débogage

# Concepts clés du débogage

- **Breakpoints** : permettent d'arrêter l'exécution à un moment clé.
- **Pas-à-pas** : exécution contrôlée ligne par ligne pour suivre le déroulement réel.
- **Inspection des variables** : observation de l'état du programme en temps réel.

# Breakpoints et breakpoints conditionnels

- Un breakpoint arrête le programme à une ligne donnée pour analyser l'état.
- Un breakpoint conditionnel n'interrompt l'exécution que si une condition est remplie.
- Ils permettent de cibler précisément les situations problématiques.

# Pas-à-pas

- **Step Into** : entrer dans la fonction appelée.
- **Step Over** : exécuter la fonction sans y entrer.
- **Step Out** : revenir à la fonction appelante.

# Inspection des variables

- Vérifier les variables locales, globales et objets complexes.
- Comparer valeurs attendues et valeurs réelles.
- Comprendre la cause d'un comportement inattendu.

## Section 6

### Conclusion : l'art du débogage

# Conclusion : l'art du débogage

- Le débogage s'apprend par la pratique, la méthode et la patience.
- Comprendre les types d'erreurs et maîtriser les outils du débogueur est essentiel.
- Un bon débogage améliore à la fois le code et la compréhension du problème.

# Merci pour votre attention !

- Des questions ?
- Des points pas clairs ?
- N'hésitez pas.