

Détecter et corriger les erreurs

Le but de cet exercice est d'apprendre à utiliser l'outil de débogage intégré d'eclipse (*debugger* in english) pour corriger des erreurs de programmation.

Fonctionnalités :


Un débogueur permet de :

- contrôler finement l'exécution d'un programme :
 - exécuter un programme une instruction à la fois (mode pas à pas),
 - interrompre temporairement l'exécution en certains points précis (point d'arrêt).
- examiner et modifier la valeur des variables ;
- afficher la pile d'appels de fonctions ;
- ...

Perspective de travail :

Le débogueur dispose d'une perspective de travail propre, constituée de différents onglets. Si cette perspective ne s'affiche pas automatiquement lors de l'exécution du débogueur, on peut la sélectionner dans la liste des perspectives disponibles (menu **Window**).

- dans l'onglet **Debug**, on voit la pile des appels de fonction ;
- dans l'onglet du code source, la dernière instruction exécutée est mise en évidence ;
- l'onglet **Variables** affiche la liste des variable locales avec leur valeur ;
- l'onglet **Breakpoints** sert à afficher la liste des points d'arrêt ;


- un jeu d'icônes spécifiques  donne un accès rapide à certaines fonctionnalités du débogueur. De gauche à droite :
 - exécuter le programme en ignorant les points d'arrêt,
 - poursuivre l'exécution (**Resume**) jusqu'au prochain point d'arrêt, la prochaine erreur ou la fin du programme,
 - mettre le programme en pause (utile si le programme boucle),
 - interrompre définitivement le programme,
 - déconnecter eclipse d'un programme externe en cours de débogage,
 - exécuter la prochaine instruction ; s'il s'agit d'un appel de fonction, exécuter l'appel puis arrêter avant la première instruction de la fonction,
 - exécuter la prochaine instruction ; s'il s'agit d'un appel de fonction, exécuter la fonction dans sa totalité puis arrêter à l'instruction qui suit l'appel,
 - poursuivre l'exécution d'une fonction interrompue puis arrêter à l'instruction qui suit l'appel de cette fonction.

Exercice

- Créez un projet et copiez le fichier **Prime.java**¹ dans le répertoire **src** ;
- Exécutez « normalement » le programme **Prime** et constatez l'erreur ; on peut déjà noter deux choses ;
 - l'erreur est due à un indice hors bornes ; voir le message d'erreur :
`java.lang.ArrayIndexOutOfBoundsException ;`

1. voir dans `/share/esir1/prog/tp00`

- eclipse affiche la pile des appels de fonctions en cours au moment de l'interruption du programme (onglet **Console**).
- Démarrez l'exécution du programme, cette fois-ci avec l'aide du débogueur (icône **Debug** ou Menu **Run/Debug**) : l'exécution du programme devrait s'interrompre sur l'instruction qui provoque l'erreur.
- Placez un point d'arrêt sur cette instruction : il suffit de faire un double-clic à gauche du numéro de ligne ; un point bleu signale le point d'arrêt qui apparaît par ailleurs dans l'onglet **Breakpoints**.
Remarque : il est possible d'associer une condition à un point d'arrêt ; dans ce cas, le programme ne s'arrête que si la condition est remplie. Très utile pour arrêter une itération lorsqu'une variable atteint ou dépasse une certaine valeur.

- Mettez fin au programme () , puis relancez-le, toujours en mode **Debug**. Examinez les valeurs des variables locales dans l'onglet **Variables**.
- Cliquez l'icône **Resume** ; à chaque arrêt, examinez les valeurs des variables. Répétez jusqu'à obtenir l'erreur.
- Si vous ne voyez pas la cause de l'erreur, exécutez votre programme en pas-à-pas depuis le début, en plaçant des points d'arrêt où vous le jugez nécessaire.
- Corrigez l'erreur qui provoque l'exception, supprimez ou désactivez les points d'arrêt puis relancez votre programme. Cette fois-ci, il devrait s'exécuter jusqu'à la fin et les résultats affichés devraient être corrects.
- Dans la fonction `main`, remplacez la constante `19` par `53` et relancez le programme ; cette fois-ci, il devrait y avoir des erreurs dans les résultats.
- Utilisez le débogueur pour corriger le programme. Pour valider votre correction, calculez tous les nombres premiers ≤ 104729 et comparez avec les résultats fournis (fichier *premiers.txt*).

Compléments

De nombreuses ressources sont disponibles sur internet ; en voici deux :

- mini tutoriel en français sur le site developpez.com
- [tutoriel](#) en anglais