

# Debugging

## Implémentation d'un debugger graphique

Steven Costiou  
Centre Inria de l'Université de Lille

Décembre 2025

Dans cet exercice **noté**, vous reprendrez vos commandes de debugging du précédent TP. Vous devez construire un debugger graphique basé sur ces commandes. La technologie employée n'a pas d'importance : du java, du web, du dessin dans un terminal, etc. Cela ne joue pas sur la note.

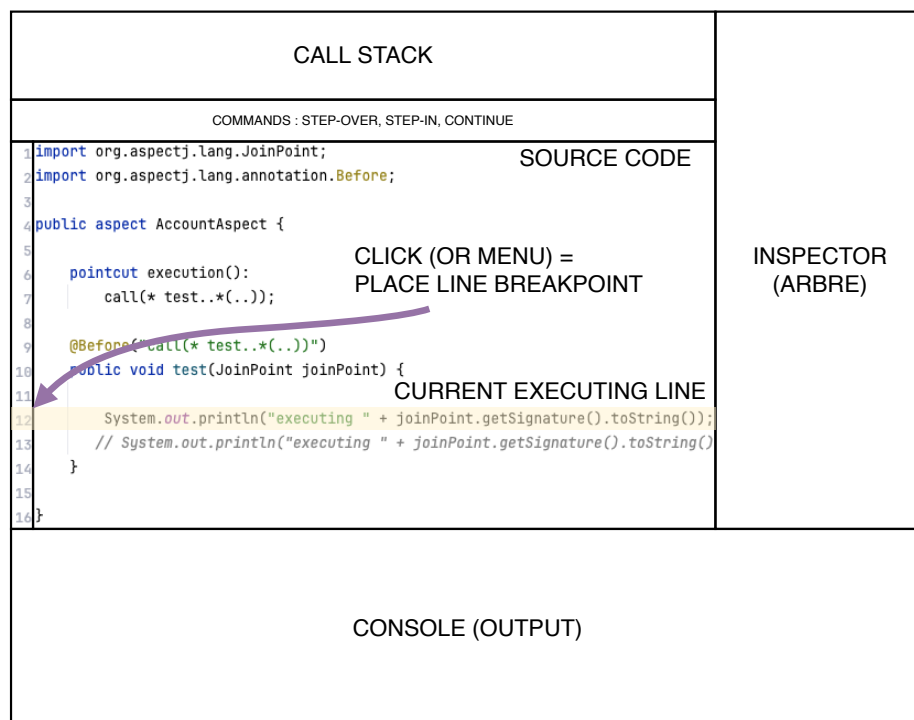


Figure 1: Debugger graphique : illustration.

La Figure 1 illustre une mise en page possible d'un debugger graphique. C'est un exemple que vous pouvez adapter à votre convenance, du moment que les fonctionnalités sont implémentées. Les points sont indiqués par fonctionnalité, pouvant se réaliser presque toutes indépendamment les unes des autres. Votre debugger doit contenir :

- Un panneau "SOURCE CODE" qui montre le code de la méthode en cours d'exécution, c'est-à-dire celle sur laquelle le debugger est arrêté, avec ses numéros de lignes. (*2 points*)
- Dans le panneau "SOURCE CODE", la ligne sur laquelle l'exécution s'est arrêtée est mise en valeur (surlignée, soulignée, etc.). (*1 point*)
- Dans le panneau "SOURCE CODE", un clic sur un numéro de ligne ou un clic droit sur une ligne via un menu déroulant permet de placer un breakpoint (simple) sur cette ligne. (*1 point*)
- Un panneau présente une liste de commandes de contrôle de l'exécution que vous avez déjà implémentées :
  - STEP-OVER
  - STEP-IN
  - CONTINUE
  - STOP

Les commandes se présentent sous la forme de boutons. Un clic sur un des boutons déclenche l'action de la commande associée. (*3 points*)

- Un panneau "CALL STACK" présente la pile d'appel de méthodes, avec en première position la dernière méthode appelée. Un clic sur un des éléments de la pile (frame) met à jour tous les autres composants avec le contenu cette frame. (*2 points*)
- Un panneau "INSPECTOR" qui présente l'ensemble des variables connues de la frame sélectionnée sous forme d'arbre, c'est-à-dire qu'un objet composé de variables d'instances se présente sous la forme d'un noeud racine affichant le type de l'objet, dont les feuilles sont ses variables d'instances, qui peuvent posséder également des feuilles. (*2 points*)
- Un panneau "OUTPUT" qui présente les éléments envoyés au flux de sortie du programme debuggé au moment de l'exécution présent. (*1 point*)
- Effectuer une action, comme un pas d'exécution (STEP) ou l'interruption dû à un point d'arrêt (BREAKPOINT) doit mettre à jour l'ensemble des composants du debugger. (*2 points*)