

# Debugger

Damit beantworte ich:

- Welchen Wert haben meine Variablen an einer bestimmten Stelle im Programmablauf?
- Wird mein Code an dieser Stelle ausgeführt (bzw. wie oft)?

Das hilft mir bei:

- Kompilierzeitfehlern     Laufzeitfehlern     logischen Fehlern

So gehe ich dabei vor:

The image shows a screenshot of the BlueJ Debugger interface with several annotations in blue boxes:

- Haltepunkt setzen durch Klick am Rand:** Points to a red square icon on the left margin of the code editor.
- Aktuelle Zeile ist hervorgehoben:** Points to the line of code where the debugger has stopped: `String eingabe = konsole.readLine();`.
- Gehe eine Zeile weiter:** Points to the 'Step' button in the debugger's control panel.
- Springe in die Methode, die in der aktuellen Zeile aufgerufen wird:** Points to the 'Step Into' button.
- Gehe bis zum nächsten Haltepunkt:** Points to the 'Continue' button.

The code editor shows the following code snippet:

```
/**  
 * Spielmethode mit Spielschleife. Abwechselnd dürfen der Spieler  
 * und der Computer einen Zug machen.  
 */  
void spiele() throws IOException, InterruptedException {  
    //Name einlesen  
    System.out.println("\nBitte gib deinen Namen ein: ");  
    String eingabe = konsole.readLine();  
    spielerName = eingabe;  
}
```

The debugger interface also shows the following information:

- Threads:** main (stopped)
- Call Sequence:** TicTacToe.spiel
- Static variables:** (empty)
- Instance variables:**
  - private Spielfeld spielfeld = <object reference>
  - private String spielerName = "Wolfgang Händler"
  - private boolean spielerAmZug = true
  - BufferedReader konsole = <object reference>
- Local variables:**
  - String eingabe = "Wolfgang Händler"

The control panel at the bottom includes buttons for Stop (Halt), Step, Step Into, Continue, and Terminate.

Hinweise:

- Haltepunkt stoppt **vor** Ausführung der Zeile, in der er gesetzt ist.
- In BlueJ werden die Haltepunkte beim erneuten Übersetzen entfernt.