Debugging – eine kurze Einführung

Einführung in die Programmierung 1

Wintersemester 21/22

Debugging

- Ausgangspunkt: Programm hat Fehler
  - Ziel: Programm ohne Fehler
- „Werkzeuge“
  - Systematische Überlegungen
  - print-Anweisungen
  - Debugger in IDE

Vorgehensweise bei Fehlerbehebung

Daten analysieren (Testresultate, Ausgaben, Fehlermeldungen, Programmtext)

Hypothese aufstellen

Wiederholbare Experimente festlegen

Einfachste Eingabe für Tests festlegen

println-Anweisungen

- Einfache Methode zum Überprüfen von Hypothesen
- Typische Punkte für Ausgaben
  - Betreten einer Methode
  - Parameterwerte
  - Rückgabewerte
  - Direkt vor und nach einer Schleife

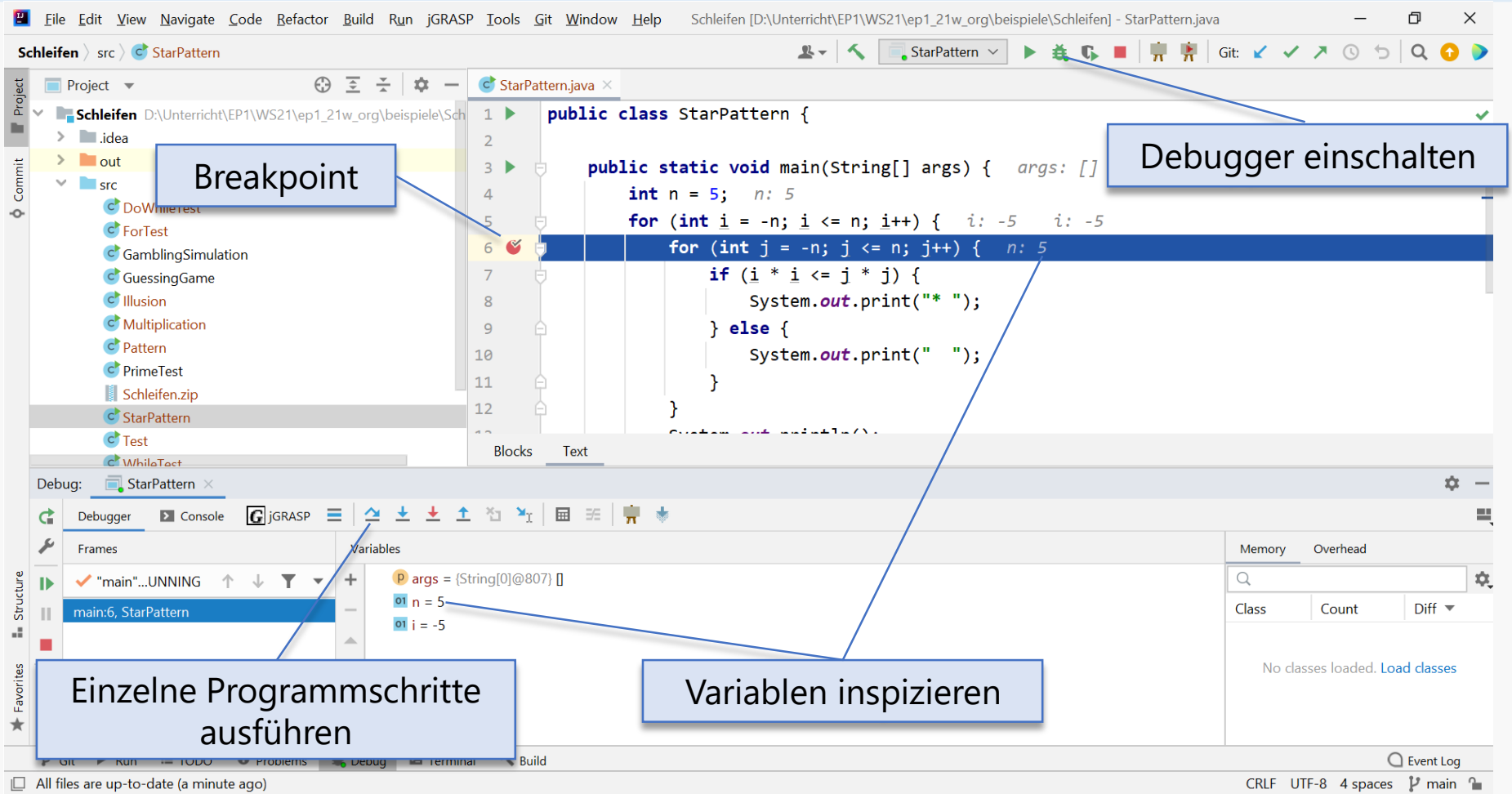
println in großen Pogrammen?

- Problemzone mit Hilfe der „binären Suche“ eingrenzen
  - println-Anweisung nach der Hälfte des Programms
  - println-Anweisung abhängig von der Ausgabe neu in eine Hälfte platzieren usw.
- Meist umständlich

Debugger

- Debugger
  - Werkzeug zum Diagnostizieren und Auffinden von Fehlern in Programmen
- Typische Funktionen
  - Steuerung des Programmablaufs (Haltepunkte, engl. Breakpoints)
  - Schrittweise Durchführung von Programmen
  - Inspizieren und modifizieren von Variablen

Debugger in IntelliJ



Beispiele

- Demonstration der Funktionalität mit unterschiedlichen Beispielen aus dem letzten Foliensatz (Arrays)

Fehler finden – einfache Fälle

- Ausnahmen beachten, z. B.
  - Indexfehler bei Arrays oder Strings
  - Nullpointer
  - Falsches Format bei Eingaben

Logische Fehler – schwierig

- Programm überdenken
  - Nicht einfach Zeilen verändern
  - Schlechte Beispiele
    - Programming by permutation
      - https://en.wikipedia.org/wiki/Programming_by_permutation
    - Shotgun debugging
      - https://en.wikipedia.org/wiki/Shotgun_debugging
- Zusätzliche Hilfe
  - Diagramme zeichnen
  - Programm anderen Personen erklären

Debuggen größerer Programme (1)

Schlecht

- Ganzes Programm schreiben
- Ganzes Programm testen
- Ganzes Programm debuggen

Besser

- Einzelne Methode implementieren
- Einzelne Methode testen und debuggen
- Einzelne Methode implementieren
- Einzelne Methode testen und debuggen
- …
- Alle Methoden integrieren

Debuggen größerer Programme (2)

Schlecht

- Programm ändern
- Änderung sich „merken“
- Testen
- …
- Keine Ahnung, wo der Fehler war!

Besser

- Backup des Programms anlegen (bzw. Versionsverwaltung)
- Programm ändern
- Potentiellen Fehler dokumentieren (z. B. Kommentare)
- Programm testen
- Neue Version mit alter vergleichen
- Bei Problemen – Backup!

Weitere Informationen zu Debugging in IntelliJ

- https://www.jetbrains.com/help/idea/debugging-code.html