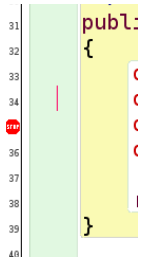
Arbeitsblatt Debugger (BlueJ)

Du willst mit deinen Klassenkameraden verschlüsselt kommunizieren, ohne dass eure Lehrerin mitlesen kann. Wie ihr bereits wisst, ist die sog. Caeser-Chiffre, bei der jeder Buchstabe um eine bestimmte Zahl im Alphabet verschoben wird, aber einfach zu knacken.

Daher habt ihr euch für folgenden Algorithmus entschieden, der das Knacken erschwert:

1. Tausche den ersten mit dem letzten und den zweiten mit dem vorletzten Buchstaben.
   z.B. wird aus „geheim“ → „miheeg“
2. Verschiebe jedes Zeichen im Alphabet um einen Schlüssel, der sich in Abhängigkeit von der Länge des Klartextes ergibt ((Länge des Klartextes modulo 4) +1).
   z.B. wird aus „miheeg“ → „plkhhj“
3. Wie du weißt, können chars durch ASCII-Zeichen und damit eine Zahl kodiert werden. Verschiebe jedes „gerade“ Zeichen um 6 (also b wird zu h, d wird zu j, …)
   z.B. wird aus „plkhhj“ → „vrknnp“

Aufgabe 1: Öffne die Klasse *„Nachrichtenaustausch“* des gleichnamigen Projekts, das du unter ddi.cs.fau.de/schule/debugging findest mit BlueJ.



Indem du auf eine Zeilennummer klickst, kannst du sogenannte *Haltepunkte (engl: Breakpoints)* an beliebigen Stellen im Code einfügen:

→ Setze einen Haltepunkt in Zeile 13 und in Zeile 16.

Erzeuge nun ein Objekt *Nachrichtenaustausch* und führe die Methode *verUndEntschlüsseln()* mit einem beliebigen String als Eingabe aus. Es öffnet sich ein neues Fenster, der *Debugger*. **Finde heraus, welche Bedeutung die folgenden Schaltflächen haben** (Hinweis: Setze dazu auch weitere Haltepunkte).

_____

_____

_____

Aufgabe 2: Beantworte nun mit Hilfe des Debuggers und passend gesetzten Haltepunkten folgende Fragen, jeweils für die Eingabe „geheim“:

1) Welchen Wert hat die Variable *schluessel* in der Klasse *Nachrichtenaustausch* nach Ausführung der Zeile 33 (`int schluessel = text.length() % 4`)? _____

2) Im Rahmen des Algorithmus zur Verschlüsselung werden Zahlen (integers) auf Buchstaben (chars) addiert. Beobachte mit Hilfe des Debuggers, was dabei passiert (z.B. Zeile 17 in der Klasse *Verschluesselung*): _____

3) Wie oft wird die Alternative in Zeile 22 der Klasse *Verschluesselung* betreten? _____

4) Beschreibe den zugrunde liegenden Fehler und behebe ihn.

   _____

Aufgabe 3: Offensichtlich funktioniert die Ver- und Entschlüsselung trotzdem noch nicht richtig. Finde mit Hilfe des Debuggers alle weiteren Fehler und behebe sie!

Fehler 1: _____

Fehler 2: _____

(Hinweis: Es bietet sich an, Haltepunkte nach jedem Schritt des Algorithmus zu setzen, um zu überprüfen, ob die einzelnen Schritte richtig durchgeführt wurden.)

Der Debugger

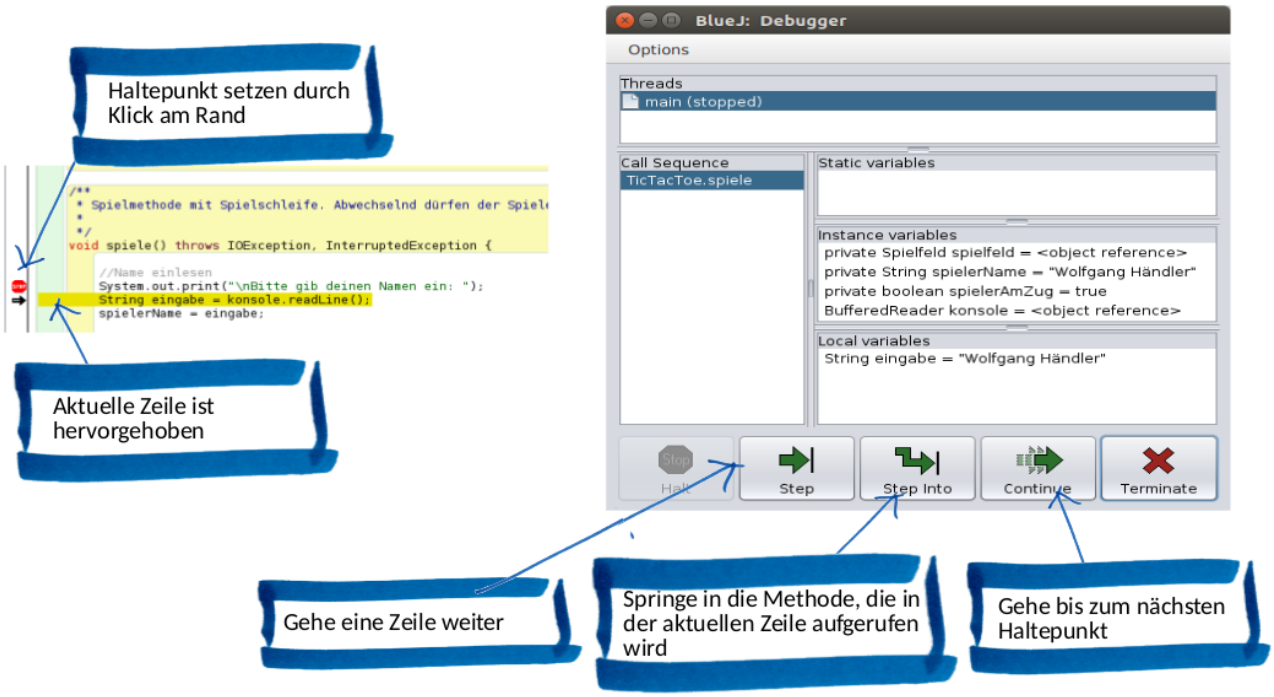
Mit Hilfe des Debuggers lassen sich folgende Fragen beantworten:

- _____
- _____

Das hilft mir bei:

☐ Kompilierzeitfehlern ☐ Laufzeitfehlern ☐ logischen Fehlern

So gehe ich dabei vor:



Hinweise:

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