LU01c - Eingaben in der Konsole

Voraussetzungen

Um die Eingaben des Benutzers zu lesen, benötigen wir eine Variable vom Typ Scanner. Der Typ Scanner und die benötigten Befehle sind nicht Teil der Java-Basis, sondern werden in einer externen Bibliothek definiert. Daher müssen wir die Bibliothek java.util.Scanner in unserem Sourcecode importieren.

- 1. Der Befehl import java.util.Scanner; importiert die externe Bibliothek.
 - Dieser Befehl muss **vor** der Zeile **public class ...** stehen.
- 2. Mittels static Scanner scanner; deklarieren wir die Variable vom Typ Scanner.
 - o Der Befehl wird direkt nach der Zeile **public class ...** geschrieben.
 - In den Beispielen nennen wir die Variable **scanner**. Natürlich können Sie auch einen anderen Bezeichner verwenden.
- 3. scanner = new Scanner(System.in); öffnet die Verbindung zur Konsole.
 - Dieser Befehl steht im Programmblock constructor: initialize attributes
- 4. Zuletzt schliessen wir den Scanner wieder: scanner.close();
 - ∘ Dieser Befehl steht am Ende des Programmblocks static void main() { ... }

Java-Gerüst mit java.util.Scanner

```
import java.util.Scanner;
                                    Importieren der externen Bibliothek
/**
                                    java.util.Scanner.
* short description of this
                                   Diese Bibliothek definiert den Typ
                                    ''Scanner''
program
                                    und enthält alle Befehle, die wir zum
* @author Ibid umm
                                   Lesen von der Konsole benötigen.
 * @since yyyy-mm-dd
 * @version 1.0
*/
public class Template {
    // declare attributes
    static Scanner scanner;
                                   Deklaration einer Variable vom Typ
     * constructor: initialize
                                    "Scanner".
                                   Wir geben der Variable den Bezeichner
attributes
                                    "scanner".
                                   Natürlich könnten Sie auch einen anderen
    public Template() {
        scanner = new
                                   Bezeichner verwenden.
Scanner(System.in);
    }
    /**
     * starts the execution
                                   Öffnen der Verbindung zur Konsole.
     *
      @param args
                  command line
arguments
    public static void
main(String[] args) {
        Template program = new
Template();
        program.run();
        scanner.close();
    }
                                   Schliessen der Verbindung zur Konsole.
    /**
     * description
     */
    private void run() {
        // TODO: realize the
program logic
    }
```

Allgemeines

Wenn das Programm die Benutzereingaben lesen soll, so wartet die weitere Verarbeitung so lange, bis der Benutzer die Enter-Taste drückt. Erst mit dem Drücken der Enter-Taste werden die eingegebenen Zeichen an unser Programm übertragen und können verarbeitet werden.

https://wiki.bzz.ch/ Printed on 2025/11/19 22:43

Alle Eingaben des Benutzers werden vom Programm als Text (Zeichenkette) gelesen. Möchte der Benutzer eine Zahl eingeben, so muss er in der Konsole die entsprechende Ziffernfolge (Zeichen) eingeben. Erst im Programm wird aus dieser Ziffernfolge eine Zahl.

Eingabe von Zahlen

Wie in der Einleitung geschrieben, werden alle Eingaben des Benutzers als Zeichen (character) gelesen. Um die eingegebenen Ziffern als Zahl zu speichern, muss der Datentyp angepasst werden. Dazu stellt Ihnen die Scanner-Bibliothek Befehle wie .nextInt() oder .nextDouble() zur Verfügung.

Diese Befehle lesen die Zeichen von der Konsole ein und wandeln die Zahl in den jeweiligen Datentyp um. Nun können Sie die umgewandelte Zahl in eine Variable vom Typ int, double, float oder einem anderen Zahlentyp speichern. Natürlich muss der Datentyp der verwendeten Variable zum eingesetzten Befehl passen.

Beispiel

```
/**
  * description
  */
private void run() {
    int    zahl1;
    double zahl2;

    zahl1 = scanner.nextInt();
    scanner.nextLine(); // Enter-Taste aus dem Scanner löschen

    zahl2 = scanner.nextDouble();
    scanner.nextLine(); // Enter-Taste aus dem Scanner löschen
}
```

scanner.nextLine()

Nach jedem nextInt(), nextDouble(), etc. ist es notwendig die verbleibende "Enter-Taste" noch aus dem Scanner zu löschen.



Ihre Eingabe in der Konsole war z.B. 15 und dann haben Sie die "Enter-Taste" gedrückt. Der Scanner speichert sowohl die Zahl 15 wie auch die "Enter-Taste", der Befehl nextInt(), nextDouble(), etc. holt jedoch nur die Zahl aus dem Scanner und die "Enter-Taste" bleibt im Scanner hängen. Diese muss anschliessen noch aus dem Scanner gelöscht werden!

Im Beispiel wird eine Ganzzahl mit dem Befehl nextInt() gelesen und in die Variable zahl1 vom Typ int gespeichert. Danach wird eine Dezimalzahl mit dem Befehl nextDouble() gelesen und in die Variable zahl2 vom Typ double gespeichert.

Für andere Datentypen existieren ebenfalls die entsprechenden next-Befehle. Zum Beispiel:

- nextShort
- nextLong
- nextFloat

Eingabe von Text

Mit dem Befehl nextLine() werden alle vom Benutzer eingegebenen Zeichen von der Konsole gelesen. Um eine solche Zeichenkette zu speichern, benötigen Sie den Datentyp String.

```
/**
 * description
 */
private void run() {
    String einText;

    einText = scanner.nextLine();
}
```

Zunächst werden wir aber vor allem einzelne Buchstaben (Character) und Zahlen verarbeiten. Der Datentyp String wird im Modul 404 genauer behandelt.

Eingabe von Buchstaben

Um einzelne Buchstaben zu lesen, müssen wir den Befehl charAt(n) verwenden. Das Argument in den runden Klammern gibt an, welches Zeichen wir lesen möchten:

```
charAt(0) = 1. ZeichencharAt(1) = 2. Zeichen
```

• ...

```
/**
  * description
  */
private void run() {
    char zeichen;

  zeichen = scanner.nextLine().charAt(0);
}
```

In diesem Beispiel lesen wir das erste Zeichen, das der Benutzer eingeben hat. Alle weiteren Zeichen werden einfach ignoriert.

https://wiki.bzz.ch/ Printed on 2025/11/19 22:43

Benutzerinteraktion

Der Benutzer kann nicht erraten, was das Programm von ihm möchte. Daher ist es wichtig, dass unser Programm klare Informationen liefert, was der Benutzer nun machen soll.

Vor jeder Eingabe schreibt das Programm, was nun vom Benutzer erwartet wird.

Geben Sie jeweils an

- Was der Benutzer eingeben soll (z.B. Anzahl).
- Welche Werte gültig sind (z.B. 1-99).
- Das Symbol > als Aufforderung, etwas einzugeben.

```
System.out.print ("Bitte Anzahl eingeben (1-99) >");
anzahl = scanner.nextInt();
scanner.nextLine(); // Enter-Taste aus dem Scanner löschen
```

Beim Ausgeben einer Variable schreibt das Programm, was dieser Wert bedeutet.

```
System.out.print( "Total = " + summe);
```

Umsetzung

```
import java.util.Scanner;
/**
* short description of this program
* @author Ibid umm
* @date yyyy-mm-dd
* @version 1.0
*/
public class Template {
   // declare attributes
    static Scanner scanner;
    /**
     * constructor: initialize attributes
    */
    public Template() {
        scanner = new Scanner(System.in);
    }
     * starts the execution
```

```
@param args
              command line arguments
 */
public static void main(String[] args) {
    Template program = new Template();
    program.run();
    scanner.close();
}
/**
 * description
private void run() {
    int
          anzahl;
    double preis;
    double summe;
    System.out.print ("Bitte Anzahl eingeben >");
    anzahl = scanner.nextInt();
    scanner.nextLine(); // Enter-Taste aus dem Scanner löschen
    System.out.print ("Preis>");
    preis = scanner.nextDouble();
    scanner.nextLine(); // Enter-Taste aus dem Scanner löschen
    summe = preis * anzahl;
    System.out.print( "Total = " + summe);
```



From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu01/lu01c-eingabe

Last update: 2024/03/28 14:07



https://wiki.bzz.ch/ Printed on 2025/11/19 22:43