Aufgabe: Briefanrede

Ausgangslage

Ein Programm soll je nach Geschlecht die korrekte Briefanrede in mehreren Sprachen ausgeben.

Sprache	Weiblich	Männlich
D	Sehr geehrte Frau	Sehr geehrter Herr
F	Chère madame	Cher monsieur
E	Dear Mrs.	Dear Mr.

Ablauf des Programms

```
1. Der Benutzer gibt das Geschlecht ein: W(eiblich), M(ännlich) oder X(Exit)
2. Solange das Geschlecht nicht 'X' ist.
  2.1. Falls das Geschlecht 'W' ist.
    2.1.1. Das Programm gibt die Anreden für Weiblich aus.
  2.2. Sonst
    2.2.1. Das Programm gibt die Anreden für Männlich aus.
  2.3. Der Benutzer gibt das Geschlecht ein: W(eiblich), M(ännlich) oder
X(Exit)
```

Ablauf im Terminal

```
Geschlecht (W oder M / X=Ende) > W
D: Sehr geehrte Frau
F: Chère madame
E: Dear Mrs.

Geschlecht (W oder M / X=Ende) > X

Uf Wiederluege
```

© Marcel Suter

2024/03/19 08:26

Auftrag: Projekt einrichten

1. Erstellen Sie ein neues Modul in Ihrem (Java-)Workspace.
2. Kopieren Sie die Vorlage in ihre IDE.
3. Passen Sie den Programmkopf (Autor, Datum) an.

Vorlage

Salutation.java

```java
import java.util.Scanner;

/**
 * show the salutations depending on gender
 *
 * @author ?????
 * @since 201?-10-??
 * @version 1.0
 */
public class Salutation {

    static Scanner scanner;
    /**
     * constructor
     */
    public Salutation() {
        scanner = new Scanner(System.in);
    }

    /**
     * starts the execution
     *
     * @param args
     *            command line arguments
     */
    public static void main(String[] args) {
        Salutation program = new Salutation();

        System.out.println("Willkommen");
        program.salute();
        System.out.println("\nUf Wiederluege!");
        scanner.close();
    }

    /**
     * enter the gender and show the salutations
     */
    private void salute() {
        // TODO-1 Variablen deklarieren

        // TODO-2.1 Ausgabe "Geschlecht (W oder M / X=Ende) >"
        // TODO-2.2 Eingabe Geschlecht

        // TODO-4 Iteration Solange Geschlecht nicht gleich X
```

```
            // TODO-3 Selektion: Falls Geschlecht gleich W
                // TODO-2.1 Ausgabe Sehr geehrte Frau
                // TODO-2.1 Ausgabe Chère madame
                // TODO-2.1 Ausgabe Dear Mrs.
            // TODO-3 Sonst
                // TODO-2.1 Ausgabe Sehr geehrter Herr
                // TODO-2.1 Ausgabe Cher monsieur
                // TODO-2.1 Ausgabe Dear Mr.
            // TODO-3 Ende der Selektion

            // TODO-2.1 Ausgabe "Geschlecht (W oder M / X=Ende) >"
            // TODO-2.2 Eingabe Geschlecht
        // TODO-4 Ende der Iteration
    }
}
```

© Marcel Suter

2024/03/19 08:26

Variablen deklarieren

Einleitung

Immer wenn ein Programm einen Wert speichern muss, benötigen wir eine Variable. Mit Variablen können wir:

- Eingaben des Benutzers speichern.
- Berechnungen durchführen und das Ergebnis festhalten.
- Bedingungen formulieren.

Zur Erinnerung

In Java werden Variablen durch Angabe eines Datentyps (`int, float, char, …`) und eines Bezeichners deklariert.

```
int zahl;
char buchstabe;
```

Auftrag

1. Bestimmen Sie alle benötigten Variablen für das Programm „Salutation“:
    - Bezeichner (Name) der Variable
    - Datentyp (char, int, float, …)

1. Deklarieren Sie die Variablen im Programm (`TODO-1 Variablen deklarieren`)

© Marcel Suter

2024/03/19 08:26

Ein- und Ausgaben realisieren (Sequenzen)

Einleitung

Eingaben und Ausgaben gehören zu den typischen Befehlen eines Programms.

Zur Erinnerung:

- System.out.print(„???“)
  Gibt eine Zeichenkette aus.
- System.out.println(„???“)
  Gibt eine Zeichenkette und anschliessend einen Zeilenumbruch aus.
- variable = scanner.nextLine()
  // Liest eine Zeichenkette und speichert diese in der Variable.//
- variable = scanner.nextLine().charAt(0)
  // Liest eine Zeichenkette und speichert das erste Zeichen in der Variable.//

Auftrag

Realisieren Sie die Eingaben und Ausgaben im Programm.

Das Scanner-Objekt (`static Scanner scanner`) wurde bereits deklariert. Auch das Öffnen (`scanner = new Scanner(System.in)`) und Schliessen (`scanner.close()`) ist bereits in der Vorlage umgesetzt.

Vorgehen

1. Ausgaben realisieren
    1. Ersetzen Sie im Programm die Kommentare für die Ausgaben (TODO-2.1) mit den korrekten Befehlen.
    2. Kompilieren Sie das Programm und korrigieren Sie allfällige Fehler.
    3. Führen Sie das Programm aus; Sie sehen die Ausgaben in der Konsole.
2. Eingaben realisieren
    1. Ersetzen Sie die Kommentare für die Eingaben (TODO-2.2) mit den korrekten Befehlen.
    2. Kompilieren Sie das Programm und korrigieren Sie allfällige Fehler.
    3. Führen Sie das Programm aus; Sie können nun 2x ein Zeichen eingeben.

© Marcel Suter

2024/03/19 08:26

Selektion realisieren

Einleitung

Mit Selektionen können wir den Ablauf unseres Programms steuern. Abhängig von einer Bedingung wird ein Block von Befehlen ausgeführt oder nicht.

Zur Erinnerung

- Das Schlüsselwort für eine Selektion ist `if`.
- Die Bedingung **muss** in runden Klammern geschrieben werden.
- Alle Befehle die zu einem Programmblock gehören, werden mit geschweiften Klammern eingefasst.
- Mit dem optionalen Schlüsselwort `else` wird ein alternativer Programmblock eingeleitet. Dieser Block wird ausgeführt, wenn die Bedingung **nicht** erfüllt ist.
    - `else` hat **keine** Bedingung.

```
if (variable == wert) {
  befehl
  befehl
} else {
  befehl
}
```

Auftrag

Realisieren Sie die Selektion im Programm.

Vorgehen

1. Ersetzen Sie die Kommentare (TODO-3) für `Falls` und `Sonst` durch die korrekten Befehle.
2. Fassen Sie die beiden Programmblöcke mit den Ausgaben in geschweiften Klammern ein.
3. Kompilieren Sie das Programm und korrigieren Sie allfällige Fehler.
4. Führen Sie das Programm aus; Sie erhalten nun die korrekten Anreden für Weiblich und Männlich.

© Marcel Suter

2024/03/19 08:26

Iteration realisieren

Einleitung

Durch eine Iteration kann ein Programmblock mehrmals ausgeführt werden. Solange die Bedingung erfüllt ist, wird ein Block von Befehlen ausgeführt.

Zur Erinnerung

- Das Schlüsselwort für eine Selektion ist `while`.
- Die Bedingung **muss** in runden Klammern geschrieben werden.
- Alle Befehle die zu einem Programmblock gehören, werden mit geschweiften Klammern eingefasst.
- Eine Iteration kennt **kein** `else`.

```
while (variable == wert) {
  befehl
  befehl
}
```

Auftrag

Realisieren Sie die Iteration im Programm.

Vorgehen

1. Ersetzen Sie den Kommentare für `Solange` (TODO-4) durch den korrekten Befehl.
2. Fassen Sie den Programmblock in geschweiften Klammern ein.
3. Kompilieren Sie das Programm und korrigieren Sie allfällige Fehler.
4. Führen Sie das Programm aus; Das Programm funktioniert nun vollständig.

© Marcel Suter

2024/03/19 08:26

Testen

Einleitung

Beim Testen versuchen wir Fehler im Programm zu finden. Finden wir einen Fehler, so halten wir diesen fest und testen weiter. Das Korrigieren der Fehler erfolgt erst, wenn wir alle Tests durchgeführt haben.

Auftrag

Testen Sie Ihr Programm und prüfen Sie, ob das Programm korrekt reagiert. Halten Sie allfällige Fehler fest.

Testfälle

1. Starten Sie das Programm.
    1. Geben Sie 'W' ein.
    2. Geben Sie 'M' ein.
    3. Geben Sie 'X' ein.
2. Starten Sie das Programm.
    1. Geben Sie 'X' ein.
3. Starten Sie das Programm.
    1. Geben Sie 'B' ein.
    2. Geben Sie 'X' ein.

© Marcel Suter

2024/03/19 08:26

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

Permanent link:
https://wiki.bzz.ch/modul/archiv/m319/learningunits/lu04/aufgaben/anrede

Last update: **2024/03/28 14:07**