

Programmieren und Software-Engineering I

Übung 15

Name: _____ Klasse: _____ Datum: _____

Lernziele:

- Benutzereingabe
- Testdokumentation
- switch

Aufgabe 1: Benutzereingabe (Ex_15_01_UserInput)

Erstelle ein Java-Programm, bei welchem der Benutzer genau 3 Zahlen eingeben muss. Die Eingabe ist nur gültig, wenn eine Zahl einstellig, eine Zahl zweistellig und eine Zahl dreistellig ist. Die Reihenfolge spielt dabei **keine** Rolle.

Beispiel:

Eingabe von 3 Zahlen (einstellig, zweistellig, dreistellig):

Zahl 1: 24
Zahl 2: 375
Zahl 3: 16

Nicht korrekt.

Eingabe von 3 Zahlen (einstellig, zweistellig, dreistellig):

Zahl 1: 24
Zahl 2: 375
Zahl 3: 6

Korrekt. Beim 2. Versuch

Erweiterung: Eingabe von 4, 5... Zahlen (vierstellig, fünfstellig,...)

Aufgabe 2: Russische Multiplikation (Ex_15_02_RussMult)

Die Russische Bauernmultiplikation (auch Ägyptisches Multiplizieren, Abessinische Bauernregel oder Verdopplungs-Halbierungs-Methode genannt) ist ein einfaches Verfahren zur Multiplikation zweier natürlicher Zahlen. Am Beispiel $13 \times 17 = 221$ wird im Folgenden das Schema der russischen Bauernmultiplikation dargestellt:

13 : 2 = 6 Rest 1	17
6 : 2 = 3 Rest 0	17 * 2 = 34
3 : 2 = 1 Rest 1	34 * 2 = 68
1 : 2 = 0 Rest 1	68 * 2 = 136
1 * 17 = 17	
0 * 34 = 0	
1 * 68 = 68	
1 * 136 = 136	

221	

Das Verfahren besteht aus folgenden Schritten:

1. Man schreibt die beiden zu multiplizierenden Zahlen nebeneinander.
2. Auf der linken Seite werden die Zahlen jeweils halbiert (Reste abgerundet) und die Ergebnisse untereinander geschrieben, bis man zur 1 gelangt.
3. Auf der rechten Seite werden die Zahlen verdoppelt und unterinandergeschrieben.
4. Die rechtsstehenden (verdoppelten) Zahlen werden gestrichen, wenn die linksstehende Zahl gerade ist.
5. Die Summe der nicht gestrichenen rechtsstehenden Zahlen ergibt das gesuchte Produkt.

Implementiere ein Java-Programm, das zwei Faktoren (positive int-Werte!) mit Hilfe der Klasse Scanner einliest und deren Produkt nach der russischen Bauernmultiplikation berechnet und ausgibt.

Testdokumentation

Beschreibung wie das Programm getestet wurde.

Testfall	Faktoren	Ausgabe erwartet	Ausgabe tatsächlich

Programmieren und Software-Engineering I

Übung 15

Aufgabe 3: Zahlensysteme (Ex_15_03_NumberSystem)

Erstelle ein Java-Programm, das beliebige Umrechnungen einer Dezimalzahl in andere Zahlensysteme ermöglicht. Dazu wählt der Benutzer eine Zahl zwischen 1 und 1.000.000, sowie eine Basis zwischen 2 und 9 bzw. 16 (Hexadezimal). Gibt er eine der beiden Größen falsch ein, erhält er eine Fehlermeldung und muss die Eingabe so lange wiederholen, bis er eine richtige Größe eingegeben hat.

Sind beide Werte korrekt eingegeben, erfolgt die Umrechnung und die Ausgabe wie in Beispiel 1. Anschließend erfolgt die nächste Benutzereingabe. Das Programm stoppt ohne weitere Aktion, sobald der Benutzer eine 0 eingibt, egal ob er sie als Wert oder Basis eingibt.

Beispiel:

Umwandlung in Zahlensysteme (Abbruch mit 0)

Zahl (1-1000000): 24
Basis (2-9 oder 16): 8
Die Zahl 24 lautet zur Basis 8: 30

Zahl (1-1000000): -4
Falsche Zahleneingabe!

Zahl (1-1000000): 4
Basis (2-9 oder 16): 4
Die Zahl 4 lautet zur Basis 4: 10

Zahl (1-1000000): 45
Basis (2-9 oder 16): 16
Die Zahl 45 lautet zur Basis 16: 2D

Zahl (1-1000000): 0
Ende der Verarbeitung!

Testdokumentation

Beschreibung wie das Programm getestet wurde.

Testfall	Zahl	Basis	Ausgabe erwartet	Ausgabe tatsächlich

Aufgabe 4: Zahl als Text (Ex_15_04_NumberAsText)

Erstelle ein Java-Programm, das eine Zahl zwischen 0 und 99 in Worten ausdrückt. Versuche im Rahmen dieses Beispiels wiederum, Verzweigungen effizient einzusetzen und mit einer möglichst geringen Zahl von Fallunterscheidungen auszukommen. Der Auszugebende Text (=die Zahl in Worten geschrieben) soll dabei in einer Variable vom Datentyp "String" gespeichert werden, und erst am Ende des Programms durch ein einziges "System.out.println" auf den Bildschirm ausgegeben werden.

Beispiel:

0:	null
1:	eins
10:	zehn
68:	achtundsechzig
41:	einundvierzig
11:	elf

Programmieren und Software-Engineering I

Übung 15

Aufgabe 5: Figur (Ex_15_05_Figur)

Erstelle ein Java-Programm das nach von Höhe, Breite und Randbreite folgende Figur ausgibt:

Beachte dabei **folgende Bestimmungen:**

Höhe: 9 – 90

Breite: 9 - 90

Randbreite: 1 – 10 und kleiner als Höhe/3 und kleiner als Breite

Beispiel: für Höhe = 15, Breite = 25 und Rand = 4

```
*****
*****
*****
*****
****          ****
****          ****
****          ****
****          ****
****          ****
****          ****
****          ****
****          ****
****          ****
*****
*****
*****
*****
```

WICHTIG

Alle Programme müssen einen Programmkopf (=Beschreibung) enthalten.

z.B.:

```
/******
*      Name:      Max Mustermann
*      Hü:        4
*      Bsp:       2
*      Datum:     20.10.2015
*      Dateiname:  HUE_04_02_Dreieck.java
*      Beschreibung: Es wird für gegebene Seitenlängen a, b und c geprüft, ob es ein gleichseitiges, ein
                    gleichschenkeliges, ein rechtwinkeliges, ein sonstiges gültiges oder ein ungültiges
                    Dreieck ist.
*****/

public class HUE_04_02_Dreieck {
```