



Technische Universität Darmstadt

Fachbereich Informatik

Prof. Dr. Andreas Koch

Allgemeine Informatik 2 im SS 2008

Übungsblatt 2

Bearbeitungszeit: 21.04. bis 27.04.2008

Aufgabe 1: Buchstaben zählen

Starten Sie BlueJ und erstellen Sie ein Projekt **uebung02**.

Erstellen Sie dann eine Klasse **LetterCounter** und löschen Sie im Quelltext alles zwischen **public class LetterCounter {** und der schließenden geschweiften Klammer. Nach diesen Vorarbeiten können wir beginnen.

Ihre Klasse soll folgendes enthalten:

- ein Attribut **text** vom Typ **String**,
- einen Konstruktor, der einen Parameter vom Typ **String** erwartet und das Attribut **text** damit initialisiert,
- eine Methode **int count(char letter)** (Details siehe unten).

Die Methode **count** soll alle Vorkommen des übergebenen Buchstaben **letter** im String **text** zählen und als Ergebnis zurückgeben. Dazu müssen Sie den String **text** Zeichen für Zeichen durchgehen, Sie benötigen also eine **for**-Schleife. Ähnlich wie bei Arrays ist der Index des ersten Zeichens **0**, die Länge des Strings **text** liefert **text.length()**. Da Sie etwas zählen wollen, brauchen Sie außerdem eine Zählvariable, die mit **0** initialisiert wird.

In der Schleife vergleichen Sie jeweils das **i**-te Zeichen (**text.charAt(i)**) mit **letter** und erhöhen entsprechend die Zählvariable um **1**. Nach der Schleife geben Sie den Wert dieses Zählers mittels **return** zurück.

Achten Sie der Klasse und allen enthaltenen Methoden, auf angemessene JavaDoc-Kommentare mit den entsprechenden Variablen (**@author**, **@version**, **@param**, **@return**...)!

Um die Funktion Ihrer Klasse zu testen, müssen Sie sie zuerst wie in der ersten Übung kompilieren (evtl. heißt es bei Ihnen **Übersetzen** anstatt **Compile**). Erzeugen Sie dann ein Objekt der Klasse, indem Sie mit der rechten Maustaste darauf klicken und **new LetterCounter(String text)** auswählen. Da der Konstruktor einen String erwartet, müssen Sie ihn jetzt eingeben, z.B. **"sechzehn tester testen, welche lettern hier sehr gern erscheinen."** Vergessen Sie die Anführungszeichen des Strings nicht!

Im unteren Bereich des BlueJ-Fensters erscheint nun in rot ein Objekt der Klasse **LetterCounter**. Rufen Sie mittels Rechtsklick die Methode **int count(char letter)** auf. Auch den Parameter dieser Methode müssen Sie eingeben, z.B. **'e'**. Achten Sie auf die einfachen Anführungszeichen beim Datentyp **char**!

Es öffnet sich ein Informationsfenster, in dem der Rückgabewert der Methode steht (in unserem Beispiel also 16).

Aufgabe 2: Euklid

Der euklidische Algorithmus ist ein Verfahren zur Bestimmung des größten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen a und b . Er ist einer der ältesten bekannten Algorithmen der Welt, benannt nach dem griechischen Mathematiker Euklid, der ihn um 300 v. Chr. in seinem Werk „Die Elemente“ angegeben hat. Der Algorithmus kommt ohne die Kenntnis der Primfaktorzerlegung der Zahlen a und b aus.

Wir werden den Algorithmus in einer etwas moderneren Fassung implementieren. Informationen zum Algorithmus finden Sie auf http://de.wikipedia.org/wiki/Euklidischer_Algorithmus.

- Fügen Sie Ihrem Projekt **uebung02** eine neue Klasse **Euklid** hinzu. Auch hier löschen Sie wieder alles außer der Klassendefinition und dem Klassenkommentar.
- Implementieren Sie nun eine Methode **static int ggt(int a, int b)**, die den größten gemeinsamen Teiler nach folgendem Algorithmus berechnet:

Solange $b \neq 0$:

$h \leftarrow a \bmod b$

(modulo entspricht in Java dem Prozentzeichen: %)

$a \leftarrow b$

$b \leftarrow h$

Ergebnis: a

Das **static** sorgt dafür, dass wir die Methode aufrufen können, ohne ein Objekt der Klasse zu erzeugen (Details folgen später in der Vorlesung).

- Kompilieren Sie nun Ihre Klasse und rufen die Methode **ggt** mittels Rechtsklick auf die Klasse auf. Probieren Sie ruhig verschiedene Parameter aus. Beispiele: $\text{ggT}(100, 75) = 25$, $\text{ggT}(10, 17) = 1$.
- Nun nutzen wir die gerade geschriebene Methode, um auch das kleinste gemeinsame Vielfache (kgV) zu berechnen. Implementieren Sie in der Klasse **Euklid** eine weitere Methode **static int kgv(int a, int b)**.

Das kgV berechnet sich nach folgender Formel:

$$\text{kgV}(a, b) = \frac{|a \cdot b|}{\text{ggT}(a, b)}$$

Für den mathematischen Betrag nutzen Sie die Java-Methode **Math.abs(...)**.

- Kompilieren Sie Ihre Klasse neu und rufen die Methode **kgv** auf. Probieren Sie ruhig auch hier verschiedene Parameter aus. Beispiele: $\text{kgV}(100, 75) = 300$, $\text{kgV}(10, 17) = 170$.
- Denken Sie natürlich auch an die JavaDoc-Kommentare!