



Übungsblatt 06

Zu bearbeiten bis 14.12.2007, 24h

Legen Sie auf Ihrem Checkout-Verzeichnis ein Unterverzeichnis `blatt06` an und legen Sie dort Ihre Lösungsdateien für dieses Übungsblatt ab.

Aufgabe 1

Die Binomialkoeffizienten lassen sich nach folgender Formel rekursiv berechnen:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}, \binom{n}{n} = 1, \binom{n}{0} = 1$$

Implementieren Sie eine (doppelt) rekursive Funktion

```
static int rBK(int n, int k)
```

die den Binomialkoeffizienten $\binom{n}{k}$ nach diesem Rekursionsschema berechnet.

Die Methode `rBK` können Sie direkt in eine Testklasse `TestRBK` integrieren, die in Ihrer `main`-Methode das Pascalsche Dreieck bis zur Höhe 20 ausgibt.

Geben Sie den Sourcecode ab.

Aufgabe 2

Bei der Berechnung von $n!$ tritt sehr schnell ein arithmetischer Überlauf ein. Leider löst Java bei einem solchen Überlauf keine Ausnahme aus. Die Folge: falsche Ergebnisse!

- Leiten Sie aus der Klasse `RuntimeException` eine Klasse `LongOverflowException` ab und implementieren Sie eine Methode `long fak(int n)` throws `LongOverflowException`, die diese Ausnahme auslöst, wenn bei der Berechnung von $n!$ die Werte überlaufen. `LongOverflowException` soll eine geeignete Fehlermeldung übergeben. Testen Sie Ihre Klasse, in dem Sie eine `main`-Methode schreiben, die durch "Probieren" das größte n herausfindet, das (noch) keine Ausnahme beim Berechnen von $n!$ auslöst.
- Was ändert sich, wenn `LongOverflowException` aus `Exception` (statt `RuntimeException`) abgeleitet wird?

Hinweis: die Multiplikation $a*b$ läuft über, falls $\text{Long.MAX_VALUE}/a < b$ ist.

Abzugeben bis Freitag, 14.12.2007, 24h in Ihr Gruppenverzeichnis auf <https://svn.uni-koblenz.de/oopm0708/students> .