

Übungsblatt 4

Abgabe bis Dienstag, 21.11.06, 11 Uhr

Hinweis: Programmieraufgaben immer per Email (eine Email pro Blatt und Gruppe) an den zuständigen Tutor schicken (Java Quellcode und eventuell benötigte Datendateien). Sie erhalten die Email-Adresse Ihres Tutoren in der ersten Übungsstunde. Bitte werfen Sie Ihre schriftlichen Lösungen in die Briefkästen in Geb. 051, Erdgeschoss ein. Für den Erhalt von Bonuspunkten müssen Sie in wenigstens 10 Übungen anwesend sein.

Aufgabe 1

Ein Professor möchte die Präsenz seiner Studenten in den Übungen elektronisch verwalten, so dass er erkennen kann, ob ein Hauptfach-Student die Anwesenheitspflicht von 50 Prozent erfüllt hat. Außerdem will er den Überblick über die erlangten Bonuspunkte eines Studenten haben. Schreiben Sie deshalb eine Klasse `Student`, die folgende Methoden enthält:

```
public String getName()  
public int getMatrNr()  
public boolean isComputerScienceMajor()  
public void hasVisited()  
public void hasVisited(int bonusPoints)  
public void hasNotVisited()  
public boolean allowedForExam()  
public int getBonusPoints()
```

Testen Sie Ihre Klasse, indem Sie in einer `main`-Methode verschiedene `Student`-Objekte erzeugen, die Anwesenheit der Studenten aus Haupt- und Nebenfach über 13 Übungstermine hin protokollieren, sowie deren Bonuspunkte verwalten und die Klausurzulassung ermitteln.

Aufgabe 2

Schreiben Sie die Klasse `LogDouble`, die das Rechnen im Log-Space erlaubt. Dies bedeutet, dass intern alle Zahlen in log Form abgespeichert werden (z.B., die Zahl 3 wird als $\log(3)$ gespeichert).

```
public double toDouble()  
public double logValue()  
public void toLogDouble(double x)
```

```

public void mul(LogDouble x)
public void mul(double x)
public void div(LogDouble x)
public void div(double x)

```

Schreiben Sie eine main-Methode die die verschiedenen Methoden der Klasse `LogDouble` testet und äußern Sie eine Vermutung unter welchen Voraussetzungen diese Klasse Vorteile gegenüber der Klasse `Double` hat.

Hinweis: Die Klasse `Math` stellt die Funktionen `log` und `exp` zur Verfügung, die Sie benutzen dürfen.

Aufgabe 3

Schreiben Sie eine Klasse `MathUtil`, die folgende Methoden enthält:

```

static int abs (int x)

```

Diese Methode liefert den Absolutbetrag von `i`. Verwenden Sie dabei keine Methode der Klasse `Math`.

```

static int longToInt (long l)

```

Diese Methode soll den übergebenen `long`-Wert in einen `int` umwandeln. Dabei soll bei Bereichsüber- bzw. Unterschreitungen der maximale bzw. minimale Integer-Wert zurückgegeben werden.

```

static int max3 (int x, int y, int z)
static int min3 (int x, int y, int z)

```

Diese Methode liefert das Maximum bzw. Minimum der drei Werte zurück.

Aufgabe 4

Erweitern Sie die Klasse `MathUtil` um eine Methode

```

static int testDivisibility (int x, int d)

```

Die Methode soll als Rückgabewerte liefern:

$$\left\{ \begin{array}{l} 0 \quad , \text{ wenn } x \text{ durch } d \text{ ohne Rest teilbar ist} \\ -1 \quad , \text{ wenn } x \text{ nicht ohne Rest durch } d \text{ teilbar ist} \\ +1 \quad , \text{ wenn einer der Parameter negativ ist} \\ \quad \quad \text{oder die Berechnung nicht vorgenommen werden kann} \end{array} \right.$$

Schreiben Sie eine main-Methode in `MathUtil`, die vom Benutzer die Eingabe von `x` und `d` verlangt, die Berechnung aufruft und das Ergebnis ausgibt.