



# Universität Karlsruhe (TH)

## Institut für Innovatives Rechnen und Programmstrukturen (IPD)

Informatik I WS 2002/03

Dozent: Prof. Dr.rer.nat. G. Goos

Übungsleiter: Markus L. Noga

<http://eins.info.uni-karlsruhe.de>

[goos@ipd.info.uni-karlsruhe.de](mailto:goos@ipd.info.uni-karlsruhe.de)

[noga@ipd.info.uni-karlsruhe.de](mailto:noga@ipd.info.uni-karlsruhe.de)

Übungsblatt 12

Ausgabe: 17.1.2003

Abgabe: 24.1.2003 14:00 Uhr

Einwurf im Keller des Informatik-Hauptbaus (Geb. 50.34)

### Aufgabe 1: Häufigkeitszählung (15 P- Punkte)

Implementieren Sie absolute und relative Häufigkeitszählung für Bigramme von Buchstaben aus unserem Chiffrialphabet sowie eine Abstandsfunktion auf relativen Häufigkeitsverteilungen. Erzeugen Sie alle möglichen Transpositionen der Breite 4. Erstellen Sie eine Vergleichsverteilung für einen beliebigen deutschen Text und entschlüsseln Sie durch Vergleich mit dieser Verteilung folgende Kryptogramme:

1. EIRHNIIBMHECCSHNEIRHRAFDHCSIENS

2. AERFNHRUTGISEEINTNUECZLIAHESLCHEEEEIDCRMAAHTMENSIEIMMSRERRTKUCZNAMHDESMANRIEBHAUCXTEX

### Aufgabe 2: Zufallszahlen (20 P- Punkte)

Folgendes Verfahren approximiert die Ludolfsche Zahl  $\pi$ :

1. Zeichnen Sie ein Gitter mit  $n \times n$  Zellen.
2. Markieren Sie alle Zellen, deren Abstand  $d$  vom Mittelpunkt des Gitters  $d \leq \frac{n}{2}$  genügt.
3. Ziehen Sie fortwährend zufällige Koordinaten und notieren Sie den Anteil  $m$  der Treffer in markierten Zellen. Es gilt:

$$m \rightarrow \pi \left( \frac{n}{2} \right)^2 / n^2 = \frac{\pi}{4}$$

Setzen Sie dieses Verfahren in Haskell um. Repräsentieren Sie das Gitter dabei als [String] und unmarkierte Zellen als Leerzeichen. Erstellen Sie Kreisdarstellungen für  $n \in \{1, 5, 11, 55, 111, 555\}$  und approximieren Sie  $\pi$ . Welcher Zusammenhang besteht zwischen der Gittergröße, der Anzahl gezogener Koordinaten und der Genauigkeit der Approximation?

### Aufgabe 3: Typklassen (15 T- Punkte)

#### 3.1 Dynamische Bindung

Gegeben folgende Typklassen:

```
class My1 a where
  g :: (a -> a) -> [a] -> a
  f :: a -> a
```

mit Instanzen:

```
instance My1 Bool where
  g f [] = True
  g f (x:xs) = f x &&& ( g f xs)
  f True = False
  f False = True
```

```
instance My1 Int where
  g f [] = 1
  g f (x:xs) = f x + (g f xs)
  f x = - x
```

```
class Size a where size :: a -> Integer

instance Size Integer where size = const 1
instance Size Char    where size = const 1
instance Size Bool    where size = const 1

instance (Size a, Size b) => Size (a, b) where
    size (x, y) = 1 + size x + size y

instance Size a => Size [a] where
    size = (+1) . sum . map size

sum = foldr (+) 0
```

welches Ergebnis liefern diese Aufrufe?

1.g f [True, True, True, False, True, False]

2.g f [-3::Int, 4::Int,-30::Int,8::Int]

3.size [True, 1, 'a', False]

4.size "allemeineentchen"

5.size "alleshateinendenurdiewursthat2"

### 3.2 Klassenvererbung in Prelude.hs

Aus der Vorlesung haben Sie nun schon einige Vererbungsstrukturen kennengelernt, z.B.: `Real` erbt von `Num` und `Ord`.

Gibt es noch weitere solchen Vererbungshierarchien in Haskell? Untersuchen Sie die `Prelude.hs` und schreiben sie die vorhandenen Klassenvererbungen auf.

### 3.3 Supermarkt-Theorie

Der Supermarkt Wull-Mert in Karlsdorf verkauft über 3000 unterschiedliche Produkte, u.a. Äpfel, Tiefkühlpizzen, Haushaltstücher und Tragetaschen.

Führen Sie geeignete Abstraktionen für Produktkategorien ein und modellieren Sie die Produktpalette als Typklassen-Diagramm.

### Aufgabe 4: Füllen von Behältern (10 T- Punkte)

Da Karlsdorf deterministisch ist, reißen Tragetaschen genau ab einer Grenzbelastung von genau 10 kg. Erweitern Sie Typklasse und konkreten Typ um eine entsprechende Abfragefunktion.

Wie kann man seine Einkäufe mit möglichst wenigen Tüten nach Hause transportieren? Geben sie einen Algorithmus an und beschreiben Sie seine Eigenschaften.