

Teil 3 - Skriptsprachen

1. Beantworten Sie die folgenden Fragen: (3 Punkte)
 - Wie unterscheiden sich *Instanzen-* und *Klassenvariablen*?
 - Welche *Vererbungsmechanismen* gibt es in Ruby?
 - Welche Funktion hat die Methode *super*?
2. Definieren Sie die Klasse **Zweirad** mit den Subklassen **Motorrad** und **Fahrrad**. Überlegen Sie genau, welche Attribute am besten in der Super- und welche in den Subklassen zu spezifizieren sind. Definieren Sie (eine/mehrere) angemessene *to_s*-Methode(n) und je eine Instanz pro Klasse. (5 Punkte)
3. Definieren Sie die Methode *zähle_wörter*, die als Argument den Namen (Pfadangabe) einer Textdatei nimmt. Diese Methode soll eine Frequenzliste aller in der Datei vorkommenden Wörter (Wortformen) berechnen und ausgeben. (6 Punkte)

Beispiel

```
zähle_wörter("c:/mein_text.txt")
Die Datei c:/mein_text.txt enthält 211 Wörter:
die - 9
das - 3
...
```

4. Definieren Sie die Methode *split*, die eine beliebige Zahl von Argumenten akzeptiert und als Wert ein Array liefert, das selbst wiederum 2 Arrays enthält: Das erste enthält die numerischen, das zweite die nicht-numerischen Argumente der Methode. (3 Punkte)

Beispiel

```
p split 1, "test", 2.3
[[1, 2.3], ["test"]]
```

5. Die Instanzenvariablen *@form*, *@bigramme* enthalten die Zeichen bzw. Bigramme, die in dem String enthalten sind, der bei der Generierung einer Instanz der Klasse Bigramm als Argument mit übergeben wird. Die Methode *generiere_bigramme* hat die Aufgabe, diese Bigramme zu berechnen. Das Programm enthält sechs Fehler. Markieren Sie die Fehler. (3 Punkte)

```
1   class bigramm
2     attr_reader :bigramme
3     def initialize(string)
4       @form = string
5       @vokabular = string.scan(/./).uniq
6       @bigramme = generiere_bigramme
7
8     def generiere_bigramme
9       a = @form.each_char
10      b = @form[1...@form.length].each_char
11      resultat = [ ]
12      loop do
13        resultat + "#{a.next}->#{b.next}"
14      end
15      return @bigramme
16    end
17  end
```