LV 4752 / 7363
Web-Engineering/WBA
Übung 01

Ruby-Übungen

Organisatorisches

- Arbeitsverzeichnis:

  `~/lv/wba/01/`

- Dateinamen:

```
01-eksteuer.rb # neu erstellen & abgeben
01-htwc.rb     # neu erstellen & abgeben
```

- Werkzeuge:

```
ruby           # Der Interpreter
emacs          # mit Ruby-Mode. NICHT X-Emacs
scite          # Ein portabler Editor, auch
               # mit Ruby-Mode
komodo         # IDE für Ruby u.a. Skriptsprachen
```

- Vorlagen:

  `(keine)`

Aufgabe A

- Allgemeine Beschreibung
  - Prozedurales, zahlenorientiertes Arbeiten mit Ruby und der Kommandozeile
  - Hier: Ermittlung von Einkommensteuer-Werten

- Material, Hinweise:
  - Angaben auf der Kommandozeile erhalten Sie im Programm über die Hash-artige Konstante **`ARGV`**
  - Berechnungsgrundlagen zum Einkommensteuertarif finden Sie z.B. hier: **`http://de.wikipedia.org/wiki/Einkommensteuertarif`**
  - Formal erhalten Sie die Einkommensteuer-Funktion durch Integration des Grenzsteuersatzes. In der Praxis läuft das auf Addition von ein paar Trapez- und Rechteckflächen hinaus.

Aufgabe A

A: Implementieren Sie folgende Methoden/Funktionen:

```
def grenzsteuersatz(zve)   --> aFloat
  # Berechnet den Grenzsteuersatz aus dem gegebenen
  # zu versteuernden Einkommen
  # Beispiel:
  # grenzsteuersatz(7664.0)          # --> 0.15
  # grenzsteuersatz(10000.0)         # --> 0.191…
End

def ek_steuer(zve)  --> aFloat
  # Berechnet die zu zahlende Einkommensteuer aus dem
  # gegebenen zu versteuernden Einkommen
  # Beispiel:
  # ek_steuer(7664.0)                # --> 0.0
  # ek_steuer(10000.0)               # --> 398. …
end
```

Aufgabe A

A: Testen Sie nun diese Funktionen

- Ergänzen Sie Code, der ein, zwei oder drei Zahlen von der Kommandozeile entgegen nimmt und deutet als
  - zvE bzw.
  - zvE1, zvE2 bzw.
  - zvE1, zvE2, zvE_increment

```
$ ./01-eksteuer.rb 10000
(Ausgabe von EK-Steuer und Grenzsteuersatz)

$ ./01-eksteuer.rb 10000 20000
(Ausgabe von zvE, EK-Steuer und Grenzsteuersatz in
  Tabellenform, hier: zwei Zeilen)

$ ./01-eksteuer.rb 10000 30000 2000
(Ausgabe von zvE, EK-Steuer und Grenzsteuersatz in
  Tabellenform, hier: 11 Zeilen)
```

Aufgabe A

A: Abschlussteil

- Jemand weist 2000 € Werbungskosten (über Pauschale) nach, z.B. für diverse Ausgaben für Fachbücher und Rechner-Hardware.

- Berechnen Sie, wie viel Einkommensteuer er spart bzw. wie viel er „netto“ (also unter Berücksichtigung der eingesparten EK-Steuer) bezahlt
  - bei 5000, 10000, 20000, und 50000 EUR zu versteuerndem Einkommen

- Hinweis:
  - Ihr Programm soll diese Angaben in einer each-Schleife ausgeben, die eine Liste der o.g. zvE-Werte iteriert.

Aufgabe B

- Allgemeine Beschreibung
  - Einlesen einer (X)HTML-Seite im WWW via HTTP und eine kleine statistische Analyse ihrer Inhalte
- Material
  - Ruby-Standardbibliothek, Pakete Net::HTTP und URI
  - String#scan sowie Reguläre Ausdrücke in Ruby für die fortgeschrittenen Teile
- Hinweise:
  - Beschränkung auf „http://…“ !
  - Beschränkung auf Seiten mit XHTML 1.0-Inhalt, wahlweise auch HTML 4.01 (Programm muss nicht mit anderen Inhalten funktionieren)
  - Exakte Zählerwerte nicht erwartet – es geht um‘s Prinzip

Aufgabe B

- Aufruf

```
$ ./01-htwc.rb <url>
(Report mit Ergebnissen, vgl. Demo)
```

- Beispiele

```
$ ./01-htwc.rb http://www.fh-wiesbaden.de

$ ./01-htwc.rb http://www.heise.de
```

Aufgabe B

- Gewünschte Ergebnisse
  - Das Programm soll die angegebene Seite per HTTP samt Header lesen und wie folgt analysieren
- Protokoll:
  - Mit welcher HTTP-Version antwortete der Server?
- Header:
  - Wieviele Header sendete er? Welche?
- Body:
  - Anzahl Zeilen gesamt?
  - Anzahl Leerzeilen?
  - Anzahl (HTML)-Kommentare?
  - Anzahl Start-Tags?
  - Anzahl End-Tags?
  - Anzahl Empty Element-Tags (XHTML)?
  - Anzahl Wörter im Nutztext (ohne Markup)?

Aufgabe B

- „Spielregeln“
  - Zweier-Teams sind erlaubt, solange jedes Team-Mitglied beide Aufgabenteile beherrscht
  - **Abnahme zu Beginn der nächsten Übung**