

# El assignment 1

## Exercise 1.

### 1.) Wie kann man eine formale Programmiersprache beschreiben?

- Eine formale Programmiersprache dient dazu im Gegensatz zu einer normalen Sprache nicht unbedingt miteinander kommunizieren zu können. Sie bezieht sich auf eine mathematische Anwendung, wobei man mit bestimmten Regeln und Strukturen gewisse Aufgaben von einer Maschine ausführen lassen kann. Diese Regeln, Strukturen und auch die Wörter sind präzise festgelegt, so dass es von einer Maschine eindeutig gelesen werden kann. Ein Beispiel für eine formale Programmiersprache ist die Sprache C, dort sind die Wörter direkt die ausführbaren Programme, was es uns ermöglicht direkt mit der Maschine zu interagieren.

### 2.) Was ist der Unterschied zwischen einem Schlüsselwort (keyword) und einer Variablen (variable) in einer Programmierungssprache?

- Schlüsselwörter einer Programmiersprache sind von den Entwicklern der Sprache vorgegeben, diese Wörter sind reserviert für bestimmte Aktionen und können vom Programmierer nicht für andere Zwecke definiert werden. Eine Variable kann der Programmierer selber definieren und ihr unterschiedliche Werte zuweisen. Eine Programmiersprache ist in diesem Falle wie das übergeordnete Element sie bietet uns verschiedene Werkzeuge an um ausführbare Dateien ausdrücken zu können. Zu solchen Werkzeugen gehören z.B Schlüsselwörter oder Variablen, aber unter anderem auch Funktionen, Nummern, Statements

### 3.) Warum sollte eine rekursive Funktion normalerweise eine if-then-else-Anweisung enthalten?

- Wenn keine if-then-else Struktur vorgegeben ist, um einen «basecase» erreichen zu können, wird es zu Laufzeitfehlern kommen da. Die Funktion welche unter anderem durch sich selbst definiert ist, würde immer wieder sich selber aufrufen. Dies nennt man dann unendliche Rekursion, welche in der Informatik, wenn möglichst zu vermeiden ist

## Exercise 2 - Binär- und Dezimalsystem Formen Sie folgende Zahlen vom Dezimal- ins Binärsystem um:

Wir rechnen die Zahl / 2 und notieren uns den Rest. Danach rechnen wir das Ergebnis durch 2 und notieren wieder den Rest. Dies machen wir so lange bis wir auf ein Ergebnis = 0 kommen.

• 12

$$12 / 2 = 6 \quad \text{Rest } 0$$

$$6 / 2 = 3 \quad \text{Rest } 0$$

$$3 / 2 = 1 \quad \text{Rest } 1$$

$$1 / 2 = 0 \quad \text{Rest } 1$$

Folglich ist die  $(12)_{10}$  der Binärzahl  $(1100)_2$  zuzuordnen.

Die folgenden Zahlen werden analog berechnet

- 33

$$33 / 2 = 16 \quad \text{Rest } 1$$

$$16 / 2 = 8 \quad \text{Rest } 0$$

$$8 / 2 = 4 \quad \text{Rest } 0$$

$$4 / 2 = 2 \quad \text{Rest } 0$$

$$2 / 2 = 1 \quad \text{Rest } 0$$

$$1 / 2 = 0 \quad \text{Rest } 1$$

Folglich ist  $(33)_{10}$  die Binärzahl; 100001

- 5

$$5 / 2 = 2 \quad \text{Rest } 1$$

$$2 / 2 = 1 \quad \text{Rest } 0$$

$$1 / 2 = 0 \quad \text{Rest } 1$$

Somit ist  $(5)_{10}$  die Binärzahl; 101

- 120

$$120 / 2 = 60 \quad \text{Rest } 0$$

$$60 / 2 = 30 \quad \text{Rest } 0$$

$$30 / 2 = 15 \quad \text{Rest } 0$$

$$15 / 2 = 7 \quad \text{Rest } 1$$

$$7 / 2 = 3 \quad \text{Rest } 1$$

$$3 / 2 = 1 \quad \text{Rest } 1$$

$$1 / 2 = 0 \quad \text{Rest } 1$$

Somit ist  $(120)_{10}$  die Binärzahl: 1111000

### Exercise 3 - Paradigmen der Programmiersprachen

#### 1. Ist Smalltalk eine Objekt-orientierte Sprache? Wie sieht es mit C aus?

- Wie in der Vorlesung erläutert, war Smalltalk die erste Sprache welche Objekte als Basis der Sprache verwendete. In Smalltalk ist alles ein Objekt. Somit ist Smalltalk objektorientiert.

- C hingegen ist nicht Objektorientiert und wird viel für low-level System- und Anwendungsprogrammierung (also sehr nahe an der Hardware) verwendet.

## 2. Ist Java eine domänenspezifische Sprache? Wie sieht es mit HTML aus?

- HTML ist eine domänenspezifische Sprache, da sie für einen speziellen Zweck entwickelt wurde. Sie dient dazu Websites zu erstellen und ihre Funktionalität zu gewährleisten.

- Java hingegen ist eine universell einsetzbare Sprache und kann auf viele verschiedene Probleme und verschiedenen Anwendungsfeldern eingesetzt werden. Somit ist Java keine domänenspezifische Sprache.

## 3. Was ist der Unterschied zwischen einem Compiler und einem Interpreter?

- Beide wandeln die zugehörige Programmiersprache um und führen sie aus. Der Unterschied ist jedoch, dass der Interpreter jeden Befehl im Code einzeln interpretiert und direkt ausführt. Der Compiler wandelt den Code in Maschinensprache um, dieser kann gespeichert werden und immer wieder ausgeführt werden, ohne umgewandelt zu werden. Ausserdem ist aufgrund dessen, dass der Interpreter den Code jedesmal den Code Schritt für Schritt ausführt langsamer. Da der Compiler eine Datei mit dem Maschinencode anlegt, benötigt er zusätzlich verfügbaren Speicherplatz, was der Interpreter wiederum nicht braucht.

## Exercise 4 - Typische Anwendungsumgebungen Verbinden Sie die Sprachen mit ihren heutzutage typischen Anwendungsumgebungen:

UML	Allgemeiner Zweck
C#	Browser
SQL	Styling
CSS	Modellierung
Typescript	Datenbanken