6.1 Bytecode–Beispiel I

Installieren Sie das Module `Jvm` von der Web–Seite der Vorlesung.

1. Compilieren Sie das `Java`–File `bct.java` aus der Vorlesung in `bct.class`.
2. Dekodieren Sie `bct.class` mit Hilfe der Haskell–Funktion `decodeClassFile`
3. Erweiteren Sie die Klasse **bct** um ein zweites Attribut

   ```
   Integer j;
   ```

4. Erweitern Sie die Haskell-Datenstruktur des Bytecode so, dass das neue Attribut enthalten ist.
5. Erzeugen Sie mit Hilfe der Funktion `encodeClassFile` ein neues Classfile.
6. Testen Sie mit Hilfe einer Java–Klasse, die eine `bct`–Objekt erzeugt, ob Ihr Classfile korrekt ist.

6.2 Bytecode–Beispiel II

Codieren Sie das folgende `Java`–Programm `ggt` in Bytecode.

```
static int ggT(int a, int b)
{
   if (a==b) return(a);
   else
  {
      if (a>b) return(ggT(a-b,b));
      else return(ggT(b-a,a));
   }
}
```

Benutzen Sie folgendes Vorgehen:

1. Übersetzen Sie zunächst *von Hand* die Methode in Bytecode.
2. Rechnen Sie das Bytecode–File für `ggt(6,3)` durch.
3. Compilieren Sie das `Java`–File `ggt.java` in `ggt.class`

   Benutzen Sie hierzu die Version 1.5 des Java–Compilers.

   Windows: `http://www2.ba-horb.de/~pl/CB/jdk-1_5_0_22-windows-i586-p.exe`

   Linux: `http://www2.ba-horb.de/~pl/CB/jdk-1_5_0_22-linux-i586.bin`

4. Dekodieren Sie `ggt.class` mit Hilfe der Haskell–Funktion `decodeClassFile`

   ```
   import Jvm.Data.ClassFormat
   import Jvm.BinaryClass
   import Data.ByteString.Lazy.Internal


   main = do
        cf <- (decodeClassFile "ggt.class")
        print cf
   ```

5. Übersetzten Sie das `Code`–Attribute in lesbare `Bytecode`–Befehle.
6. Erweitern Sie die Haskell-Datenstruktur des Bytecode so, dass Sie eine Abfrage einbauen, die Variable `a` < 0 ist. Wenn ja soll `-1` zurückgegeben werden.
7. Testen Sie Ihr Classfile mit dem Aufruf **ggt.ggT(-1,3)** in einer anderen Java–KLasse.

6.3 ASM

1. Schreiben Sie die Deskriptoren von folgenden Java-Methoden:
   (a) `int m1(long a, double b, String s)`
   (b) `void m2(boolean[] p)`
   (c) `long[][] m3(int a, Integer b, Long c)`
2. Erzeugen Sie mithilfe von ASM den Bytecode der folgenden Java-Programme:
   (a)
   ```
   public interface Name {
     int NUM1 = 1;
     int NUM2 = 2;
     String method(long a, boolean b);
   }
   ```
   (b)
   ```
   public class GGT {
     static int ggT(int a, int b) {
       if (a == b) return a;
       else {
         if (a > b) return ggT(a - b, b);
         else return ggT(b - a, a);
       }
     }
   }
   ```