



Abgabe: KW 16 vor dem jeweiligen Praktikum

Praktikum Grundlagen der Programmierung

Aufgabe 1 (Ü) Hello World

Gegeben sei folgendes Java-Programm:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello_world");
    }
}
```

- Was macht das Programm?
- Geben Sie das Programm mit Hilfe eines Editors ein.
- Speichern Sie das Programm unter dem Namen `HelloWorld.java` ab.
- Übersetzen Sie das Programm mit dem Befehl `javac HelloWorld.java` in Bytecode.
- Führen Sie Ihr Programm mit `java HelloWorld` aus.

Aufgabe 2 (Ü) Schleifen

Schreiben sie jeweils ein Programm mit einer

- `while`-Schleife,
- `for`-Schleife,

welches alle Zahlen zwischen 0 und 100 auf der Konsole ausgibt. Testen Sie Ihr Programm. Was passiert, wenn Sie die Bedingungen der Schleifen verändern?

Hinweis: Verwenden Sie für die Ausgabe die Java-Methode `System.out.println()`.

Aufgabe 3 (Ü) MiniJava

Gegeben sei folgendes MiniJava-Programm:

```
int x , y ;
x = read();
y = read();
while(x != y){
    if(x > y){
        x = x - y;
    }else{
        y = y - x;
    }
}
write(x);
```

- Was macht das Programm?
- Zeichnen Sie den zugehörigen Kontrollflussgraphen
- Zeichnen Sie den zugehörigen Syntaxbaum.
- Übersetzen Sie das angegebene Programm in MiniJavaVM Code.

Hinweis: Zum Zeichnen können Sie das Programm OpenOffice Draw verwenden, welches für viele verschiedene Plattformen frei verfügbar ist: <http://de.openoffice.org>

Aufgabe 4 (H) Wiederholungen und Bedingungen

(3+2 Punkte)

Schreiben sie jeweils ein Programm mit einer

- for-Schleife,
- while-Schleife,

welches von 100 bis 0 *herunter* zählt. Dabei soll jede durch 10 teilbare Zahl auf der Konsole ausgegeben werden

Hinweis: Verwenden sie für die Ausgabe die Java-Methode `System.out.println()`.

Aufgabe 5 (H) MiniJava

(2+4+4 Punkte)

Gegeben sei folgendes MiniJava-Programm:

```
int x , y;
x = 1;
y = read();
while( x <= 100){
    if( x % y == 0)
        write(x);
    else
        write(0);
    x = x + 1;
}
```

- Zeichnen Sie den zugehörigen Kontrollflussgraphen
- Zeichnen Sie den zugehörigen Syntaxbaum.
- Übersetzen Sie das angegebene Programm in MiniJavaVM Code.

Hinweis: Zum Zeichnen können Sie das Programm OpenOffice Draw verwenden, welches für viele verschiedene Plattformen frei verfügbar ist: <http://de.openoffice.org>

Anhang

Regeln aus der Vorlesung zur Übersetzung von MiniJava nach MiniJVM:

prog	=	ALLOC n Übersetzung von ss HALT — sofern prog aus einer Deklaration von n Variablen, gefolgt von der Statement-Folge ss besteht.
x	=	LOAD i — sofern x die Variable mit Adresse i ist.
c	=	CONST c — sofern c eine Konstante ist.
expr ₁ + expr ₂	=	Übersetzung von expr ₁ Übersetzung von expr ₂ ADD — Analog für die anderen Operatoren (SUB, MUL, DIV, MOD, AND, OR, LESS, LEQ, EQ, NEQ, GT, GEQ)
-expr	=	Übersetzung von expr NEG — Analog für NOT
x = expr;	=	Übersetzung von expr STORE i — sofern x die Variable mit Adresse i ist.
x = read();	=	READ STORE i — sofern x die Variable mit Adresse i ist.
write(expr);	=	Übersetzung von expr WRITE
if (cond) stmt	=	Übersetzung von cond FJUMP A Übersetzung von stmt A: ...
if (cond) stmt ₁ else stmt ₂	=	Übersetzung von cond FJUMP A Übersetzung von stmt ₁ JUMP B A: Übersetzung von stmt ₂ B: ...
while (cond) stmt	=	A: Übersetzung von cond FJUMP B Übersetzung von stmt JUMP A B: ...
stmt ₁ ... stmt _k	=	Übersetzung von stmt ₁ ... Übersetzung von stmt _k

Die Grammatik von MiniJava

```
<program> ::= <decl>* <stmt>*
<decl> ::= <type> <name> (, <name>)* ;
<type> ::= int
<stmt> ::= ; | { <stmt>* } |
          <name> = <expr>; | <name> = read(); | write( <expr> ); |
          if ( <cond> ) <stmt> |
          if ( <cond> ) <stmt> else <stmt> |
          while ( <cond> ) <stmt>
<expr> ::= <number> | <name> | ( <expr> ) |
          <unop> <expr> | <expr> <binop> <expr>
<unop> ::= -
<binop> ::= - | + | * | / | %
<cond> ::= true | false | ( <cond> ) |
          <expr> <comp> <expr> |
          <bunop> <cond> | <cond> <bbinop> <cond>
<comp> ::= == | != | <= | < | >= | >
<bunop> ::= !
<bbinop> ::= && | ||
```

<name> und <number> für Bezeichner und Zahlen werden nicht weiter verfeinert.