TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Sprachen und Beschreibungsstrukturen
Praktikum Grundlagen der Programmierung
F. Forster, T. Gawlitza, A. Flexeder

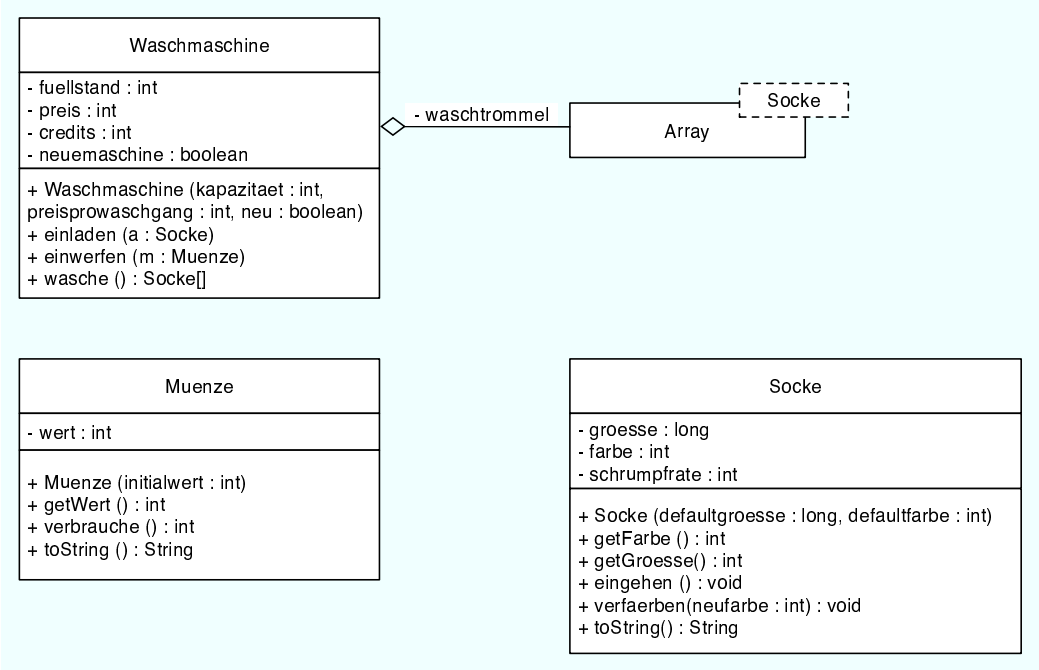
WS 2007/2008
Lösungsvorschläge zu Blatt 5
16. November 2007

Übungen zu Praktikum Grundlagen der Programmierung

Aufgabe 27 Übersetzung von MiniJava nach MiniJVM (Lösungsvorschlag)

```
   ALLOC 4 // int x, y, z, res;

   CONST 1 // x=1;
   STORE 0

   CONST 2 // y=2;
   STORE 1

   CONST 5 // z = 5*x - y/2
   LOAD 0
   MUL
   LOAD 1
   CONST 2
   DIV
   SUB
   STORE 2

   LOAD 2 // (z<y)?
   LOAD 1
   LESS
   FJUMP A
   LOAD 2 // true => res = z
   STORE 3
   JUMP B
A: LOAD  1// false => res = y
   STORE 3
B: LOAD 3
   WRITE
   HALT
```

Aufgabe 28 Waschsalon (Lösungsvorschlag)




```
public class Waschsalon {
  public static void main(String[] args){
    Waschmaschine bosch = new Waschmaschine(10,2,true);
    Waschmaschine miele = new Waschmaschine(20,1,false);
    Socke lieblingssocke = new Socke(38,4);
    Socke [] businesssocke = new Socke[5];
    for (int i=0; i< businesssocke.length;i++)
      businesssocke[i]=new Socke(38,0);
    Muenze zwickel = new Muenze(2);
    Muenze euro = new Muenze(1);
    Muenze nocheineuro = new Muenze(1);


    // Wir betreten den Waschsalon:
    bosch.einwerfen(zwickel);
    bosch.einladen(lieblingssocke);
    for (int i = 0; i< businesssocke.length; i++)
        bosch.einladen(businesssocke[i]);
    bosch.wasche();
    // alles sabuer, hurra!
    System.out.println("Lieblingssocke␣:"+lieblingssocke);
    // aber das Geld is verbraucht!

    bosch.einwerfen(euro);
    bosch.einladen(lieblingssocke);
    // nicht genug Geld
    if (bosch.wasche()==null) System.out.println("mist,␣nicht␣genug␣Geld");

    // dann halt andere Maschine
    miele.einwerfen(nocheineuro);
    miele.einladen(lieblingssocke);
    miele.einladen(businesssocke[0]);
    miele.wasche();
    // Schade um die Lieblingssocke:
    System.out.println("Lieblingssocke:␣"+lieblingssocke);
  }
}

public class Waschmaschine {
  private int fuellstand;
  private int preis;
  private int credits;
  private boolean neuemaschine;
  private Socke [] waschtrommel;
  public Waschmaschine(int kapazitaet,int preisprowaschgang, boolean neu){
```

```
    neuemaschine = neu;
    credits = 0;
    fuellstand = 0;
    preis = preisprowaschgang;
    waschtrommel = new Socke[kapazitaet];
  }
  public boolean einladen(Socke s) {
    if (fuellstand >= waschtrommel.length) return false;
    waschtrommel[fuellstand++] = s;
    return true;
  }
  public void einwerfen(Muenze m) {
    credits+=m.verbrauche();
  }
  public Socke [] wasche() {
    if (credits < preis) return null;
    credits -= preis;
    Socke[] rueckgabe = new Socke[fuellstand];
    int i,durchschnittsfarbe=0;
    for (Socke s = waschtrommel[i=0];i<fuellstand;s=waschtrommel[++i])
        durchschnittsfarbe+=s.getFarbe();
    durchschnittsfarbe /= fuellstand;
    for (Socke s = waschtrommel[i=0];i<fuellstand;s=waschtrommel[++i]){
      if(!neuemaschine) s.eingehen();
      if(!neuemaschine) s.verfaerben(durchschnittsfarbe);
      rueckgabe[i]=s;
    }
    waschtrommel = new Socke[waschtrommel.length];
    fuellstand = 0;
    return rueckgabe;
  }
}

public class Muenze {
  private int wert;
  public Muenze(int initialwert) {
    wert = initialwert;
  }
  public int getWert() {
    return wert;
  }
  public int verbrauche(){
    int verbraucht = wert;
    wert =0;
    return verbraucht;
  }
  public String toString(){
    return "Muenze mit Wert "+wert;
  }
}

public class Socke {
  private long groesse;
  private int farbe;
  private int schrumpfrate;
  public Socke(long defaultgroesse, int defaultfarbe) {
    groesse=defaultgroesse;
    farbe=defaultfarbe;
    schrumpfrate = 1;
  }
  public int getFarbe(){
    return farbe;
  }
  public long getGroesse(){
    return groesse;
  }
  public void eingehen(){
    groesse = groesse - schrumpfrate;
  }
  public void verfaerben(int neufarbe) {
    if (neufarbe<farbe) farbe--;
    if (neufarbe>farbe) farbe++;
    return;
  }
  public String toString(){
    return "Socke der Groesse "+groesse+" und Farbe "+farbe;
  }
```

```
}
```

Aufgabe 29 Matrix (Lösungsvorschlag)

```
public class Matrix {
    private int rows;
    private int cols;
    private int[][] matrix;

    public Matrix(int rows,int cols, int init){
        this.rows=rows;
        this.cols = cols;
        matrix = new int[rows][cols];
        for(int i=0; i < rows; i++)
            for(int j=0; j < cols; j++)
                setAt(i,j,init);
    }

    public Matrix(int[][] m){
  this.rows=m.length;
        this.cols = m[0].length;
        matrix = new int[rows][cols];
  for(int i=0; i<rows; i++)
      for(int j=0; j<cols; j++)
    matrix[i][j] = m[i][j];
    }

    public Matrix(int rows,int cols){
        this.rows=rows;
        this.cols = cols;
        matrix = new int[rows][cols];
    }

    public int getAt(int row,int col){
        int retvalue=-1;
        if(row>=0 && row<rows && col>=0 && col<cols)
            retvalue = matrix[row][col];
        return retvalue;
    }

    public void setAt(int row,int col, int value){
        if(row>=0 && row<rows && col>=0 && col<cols)
            matrix[row][col] = value;
    }

    public Matrix add(Matrix m){
        if(m.getRows() != getRows() && m.getCols() != getCols())
            return null;
        Matrix result = new Matrix(m.getRows(), m.getCols());
        for(int i=0; i<result.getRows(); i++)
            for(int j=0; j<result.getCols(); j++)
                result.setAt(i,j, this.getAt(i,j)+m.getAt(i,j));
  return result;
    }

    public int getRows(){
        return rows;
    }

    public int getCols(){
        return cols;
    }

    public String toString() {
        String s = "";
        for(int i=0; i<rows; i++){
      for(int j=0; j<cols; j++)
    s += getAt(i,j) + "\t";
      s += "\n";
  }
        return s;
    }


}
```