**import** java.util.ArrayList;

/\*\*

**\*** Eine Zusammenfassung basierend auf "Java lernen mit BlueJ".

**\***

**\*** **@author** **Marcel Gehrig**

**\*** **@version** 2012.12.05

\*/

**public** **class** Zusammenfassung //Klassennahme

{

**private** int preis; //Datenfeld

//-private: default

//-Datentyp: int, String, ArrayList<String>, HasMap<String, String>

/\*\*

**\*** Konstruktor. Wird nach derm Erzeugen des Objektes ausgeführt.

**\*** Der Konstruktor hat die selbe Bezeichnung wie die Klasse.

\*/

**public** Zusammenfassung()

{

preis = 10;

}

/\*\*

**\*** Mehrere Konstruktoren mit verschiedenen Argumenten sind möglich.

\*/

**public** Zusammenfassung(int setPreis)

{

preis = setPreis;

}

/\*\*

**\*** Sondierende Methode.

\*/

**public** int gibPreis()

{

int umrechnungsfaktor = 10; //Lokale Variable p.76

//Lokale Variable wurde gleich initialisiert

**return** preis;

}

/\*\*

**\*** Verändernde Methode.

\*/

**public** void setPreis(int setPreis)

{

preis = setPreis;

}

/\*\*

**\*** Ausgaben in Methoden.

\*/

**public** void druckePreis()

{

**System**.out.println("# " + preis + "SFr.");

**System**.out.println(); //neue leere Zeile

}

/\*\*

**\*** Ersetzt die toString Methode vom System

**\*** print(z1)**:** z1**,** wird automatisch in einen String gewandel

**\*** defaultmäsig zur Speicheradresse.

\*/

**public** **String** toString()

{

**return** "Re: " + re + " Im: " + im + "i";

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***/**

**/\*\*** Die bedingte Anweisung (p. 73) \*/

if (((preis > 0) && (preis <= 5)) ¦¦ (preis > 0)){

preis = 5;

}

else {

preis = 10;

}

/\*\* String (p. 83**,** 180) \*/

**private** String name;

name = "Peter läuft nach Hause. ";

name.substring(0,2); //->"Pe"

string1 == string2; //immer false

string1.equals(string2); //möglich

string1.contains(string2;

/\*\* Array**,** definierte Länge**,** mit Index (p. 153) \*/

**private** int[] zugriffeInStunde;

zugriffeInStunde = new int[24];

zugriffeInStunde = new int[]{ 4, 6, 2, 7, 5, 8, 9, 3, 1 }; //nicht überprüft

zugriffeInStunde.legth; //Länge des Arrays

zugriffeInStunde[0]; //gibt erstes Element zurück

// 2D-Array

**private** int[][] pixelMatrixTemp = new int[pixelMatrix.length][pixelMatrix[0].length];

for (int x = 0; x < pixelMatrix.length; x++){

pixelMatrixTemp[x] = pixelMatrix[pixelMatrix.length-x-1];

/\*\* Array Liste**,** undefinierte Länge**,** mit Index (p. 127) \*/

import java.util.ArrayList;

**private** ArrayList<**String**> notizen;

notizen = **new** ArrayList<**String**>();

notizen.add("Notiz 1");

notizen.get(notiznummer);

notizen.size; //Länge einer ArrayListe

notizen.contains("Notiz 1");

notizen.remove(notiznummer); //Höhere Indezies rücken nach

notizen.remove("Notiz 1");

/\*\* HashSet**,** unsortierte Menge (p. 197)\*/ //Hash-Set: unsortierte ArrayList

HashSet<**String**> woerter = **new** HashSet<**String**>();

**String**[] wortArray = eingabezeile.split(" "); //Array

**for**(**String** wort : wortArray) {

woerter.add(wort);

}

/\*\* HashMap(p. 191)\*/ //Schlüssel-Wert-Paare wie Dictionaries bei Python

HashMap<**String**, int> telefonbuch = **new** HashMap<**String**, int>();

telefonbuch.put("Günter Schmidt", 079123654848);

nummer = telefonbuch.get("Günter Schmidt");

Set<**String**> keys = telefonbuch.keySet();

**for** (**String** singleKey : keys) {

**System**.out.println(telefonbuch.get(singleKey));

}

/\*\* for**-**each**-**Schleife (p. 135) \*/

**for** (**String** notiz : notizen){ // rechts vom : eine Sammlung (z.B Array oder Set (Hash-Map))

**System**.out.println(notiz + " ");

}

/\*\* while**-**Schleife (p. 137) \*/

int index = 0;

**while**(index < notizen.size()){

**System**.out.println(notizen.get(index) + "");

index++;

}

/\*\* for**-**Schleife (p. 161) \*/

**for**(int x = 0; x <= 100; x++){

**System**.out.println(x + " ");

}

//Klassendefinition / Klassenvereinbaruung: Ersten Par Zeilen, inklusive Datenfelder=Instanzvariablen

//Klassen

//Konstruktor: Wird ausgeführt wenn ein Objekt ausgeführt wird

//Signatur / Kopf: 1. Zeile des Konstruktors oder Methode

//Rückgabewert: z.B. void, int, String...

//Bezeichner: Verweist in der Regel auf ein Objekt, Ausnahme: Primitive Datentypen: Integer, Double

//Lokale Variable

//Instanzvariable = Datenfeld

//Laufvariable: Nur innerhalb einer "for-" Schleife gültig

//Eine Variable wird mit einem bestimmten Typ deklariert

//Der Formale Parameter "absender" oder die Lokale Variable "absender

// wird dem Datenfeld "absender" zugewiesen (p. 116)

**this**.absender = absedner;

//Generische Klassen z.B.: ArrayList<Personen>;

// Ein zweiter Typ muss in den eckigen Klammern übergeben werden.