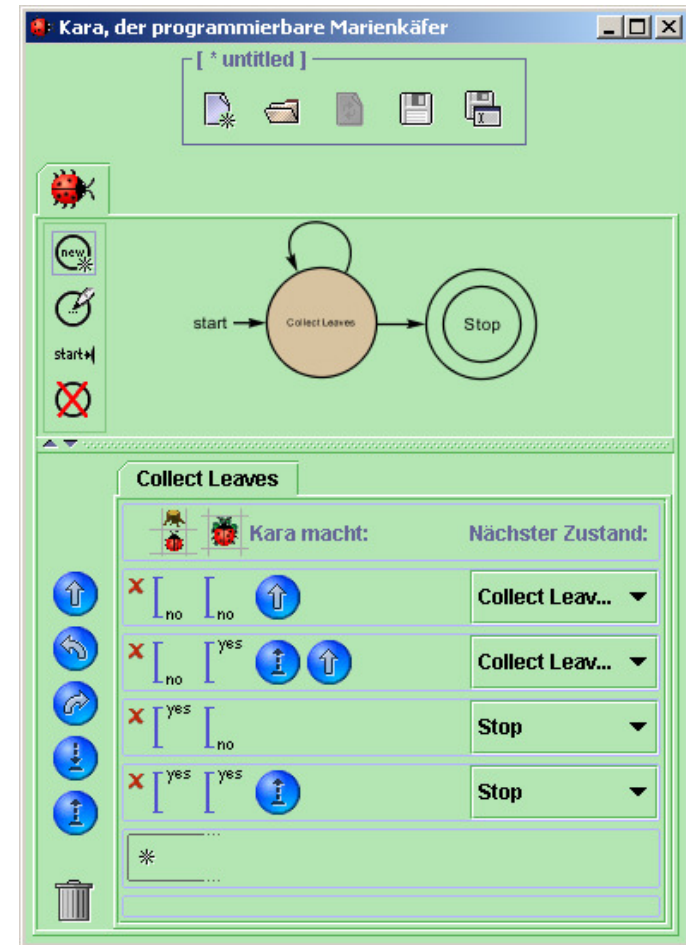


Gestern: Kara



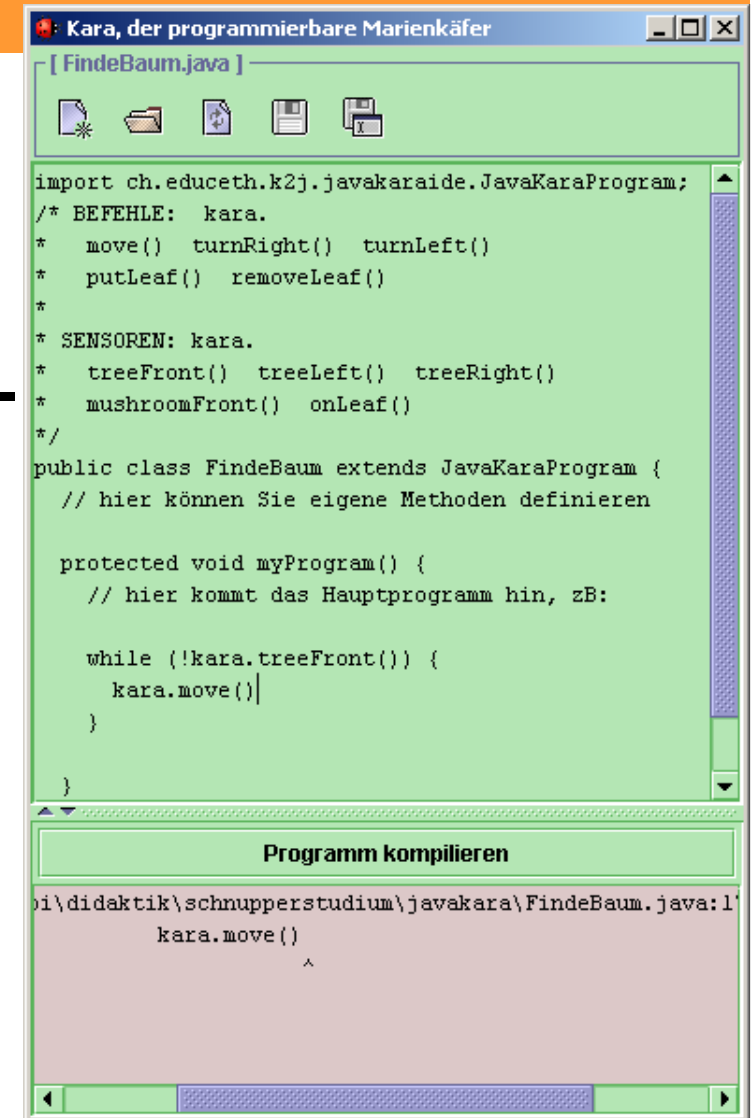
Steuerung:
Endliche
Automaten



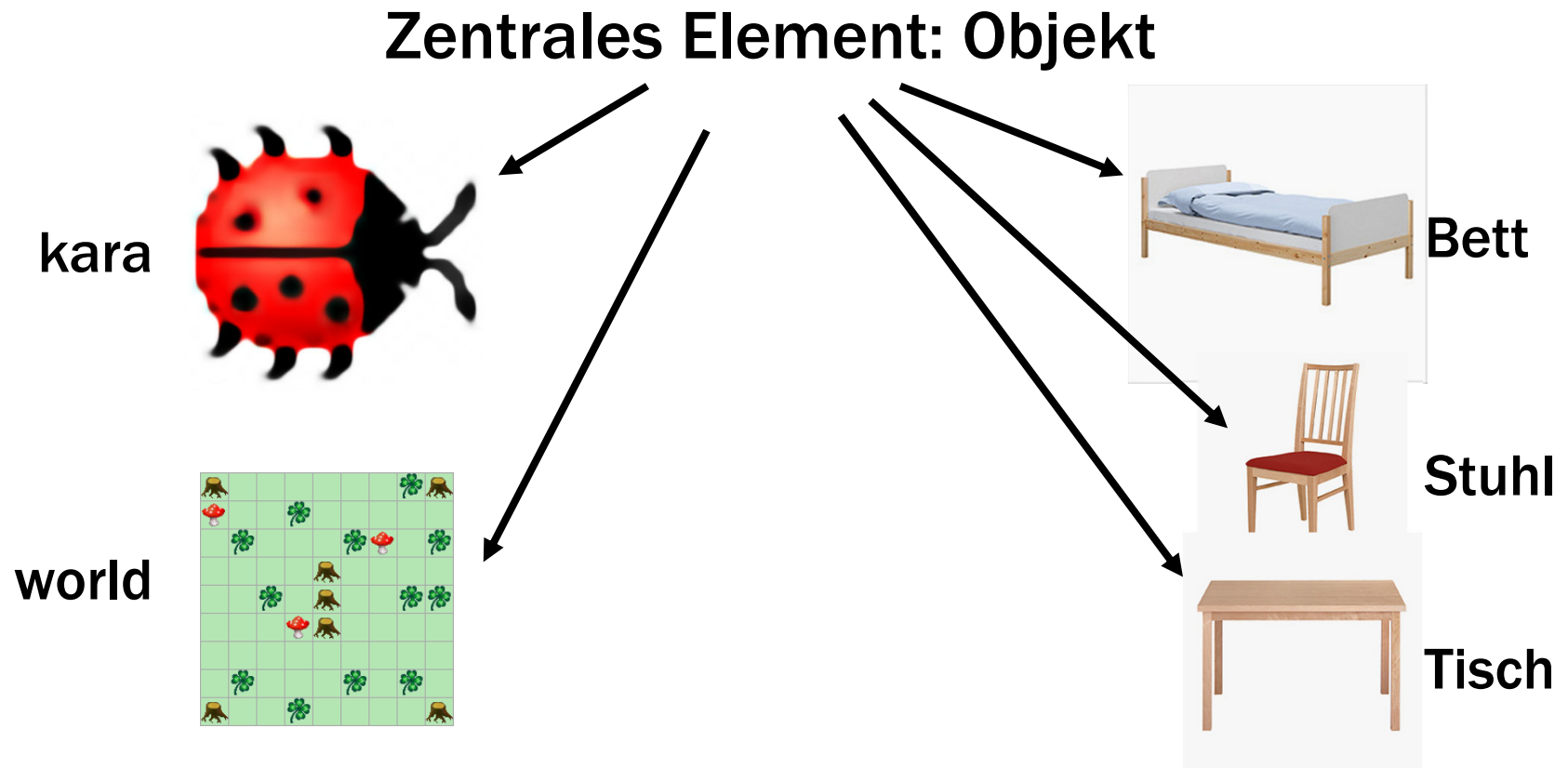
Heute: JavaKara



Steuerung:
Programmiersprache Java








Java ist „objektorientiert“



Objekte: Befehle und Sensoren

Jedes Objekt stellt Methoden zur Verfügung

Objekt	Methodenaufruf	Resultat
	<code>meinTisch.setColor(rot);</code>	
	<code>kara.move();</code>	
	<code>(...) kara.treeFront() (...)</code>	<code>true</code> (ja) / <code>false</code> (nein)

Was das Objekt „kara“ alles kann

Anweisungen



`kara.move()`

Kara läuft einen Schritt vorwärts



`kara.turnRight()`

Kara dreht sich an Ort und Stelle nach rechts



`kara.turnLeft()`

Kara dreht sich nach links



`kara.putLeaf()`

Kara legt ein Kleeblatt ab



`kara.removeLeaf()`

Kara nimmt ein Kleeblatt auf

Sensoren, um die Umgebung zu überprüfen



`kara.treeFront()`

Gibt es einen Baum auf dem Feld vor Kara?



`kara.treeLeft()`

Gibt es einen Baum auf dem Feld links von Kara?



`kara.treeRight()`

Gibt es einen Baum auf dem Feld rechts von Kara?



`kara.mushroomFront()`

Gibt es einen Pilz auf dem Feld vor Kara?



`kara.onLeaf()`

Steht Kara auf einem Kleeblatt?



Grundlagen des Programmierens

Ein kleiner Werkzeugkasten mit flexiblen und daher mächtigen Werkzeugen.

Die Werkzeuge fallen grob in zwei Kategorien:

- 1 – Steuerung des Programmablaufs: Kontrollstrukturen
- 2 – Verarbeiten von Daten: Variablen

Programme in JavaKara: „Schablone“ ist vorgegeben

```
import javakara.JavaKaraProgram;  
public class SchrittVorwaerts extends  
JavaKaraProgram {  
  
    public void myProgram() {  
        // hier kommt das Hauptprogramm hin  
  
        kara.move() ;  
  
    }  
}
```




Zweites Programm

```
import javakara.JavaKaraProgram;  
public class GeheUmBaumHerum extends JavaKaraProgram {  
    public void myProgram() {  
        // hier kommt das Hauptprogramm hin  
  
        kara.turnLeft();  
        kara.move();  
        kara.turnRight();  
        kara.move();  
        kara.move();  
        kara.turnRight();  
        kara.move();  
        kara.turnLeft();  
  
    }  
}
```

Identisch!



Geschickter: Neue Methode

```
import javakara.JavaKaraProgram;
public class GeheUmBaumHerum extends JavaKaraProgram {

    void viertelDrehung() {
        kara.move();
        kara.turnRight();
        kara.move();
    }

    public void myProgram() {
        // hier kommt das Hauptprogramm hin

        kara.turnLeft();
        viertelDrehung();
        viertelDrehung();
        kara.turnLeft();

    }
}
```



Programmieren im Kleinen

Programmieren bedingt gewisse Handfertigkeiten. Zuerst der Umgang mit einzelnen Befehlen und Kontrollstrukturen – die Tonleiter sozusagen.

Mit der Zeit kommen Design Patterns hinzu, man erkennt wiederkehrende Muster zur Strukturierung – Akkorde etc.

Und vor allem gilt: Übung macht den Meister! Auch Klavierspielen lernt niemand in einigen wenigen Lektionen.



MOZART
2006 250. GEBURTSTAG

Original-Notenblatt von W. A. Mozart, Österreichische Nationalbibliothek © ANTON WIESENHOFER

Programmieren im Grossen

Die einzelnen Bestandteile riesiger Software-Systeme sind überschaubar. Schlussendlich gibt es ja nur ein paar wenige Befehle, die ein Prozessor ausführen kann.

Die Komplexität entsteht wie bei einer Symphonie durch das raffinierte Zusammenspiel einer grossen Anzahl von Bestandteilen.

Was haben wir gemacht ?

- 1. Programm schreiben und speichern**
- 2. Programm laufen lassen**
- 3. Programm ausführen**

Unser Koch versteht nur Schwedisch

„Ägg i blå sås

Detta milt exotiska recept ingick i "Smekmånadens Kokbok", vilken Kooperativa Förbundet distribuerade till nygifta under 1930-talet. En verklig funkis-rätt! Var inte oroliga för metylenblått, den akuta giftigheten är låg.“



Wir brauchen einen Übersetzer!

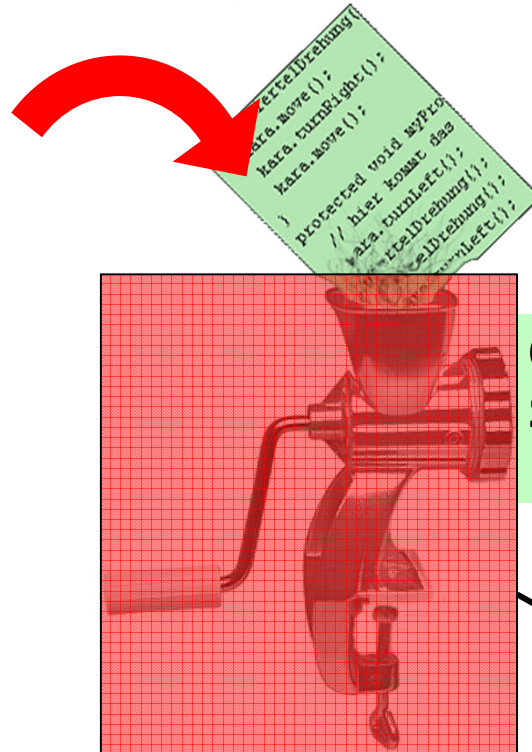
**Bei Programmiersprachen heisst ein solcher Übersetzer
Kompiler**

Kompilieren

Um Kara mit Java zu füttern, muss der Text mit dem Java-Programm kompiliert werden

GeheUmBaumHerum.java

```
void viertelDrehung(  
    kara.move();  
    kara.turnRight();  
    kara.move();  
}  
protected void myPro  
    // hier kommt das  
    kara.turnLeft();  
    viertelDrehung();  
    viertelDrehung();  
    kara.turnLeft();
```



```
01001001101110101111010  
10010100111000010110111  
10100110101011100
```

GeheUmBaumHerum.class

Compiler



Nochmals die Schablone für JavaKara-Programme

```
import JavaKaraProgram;  
public class SchrittVorwaerts extends JavaKaraProgram {  
    public void myProgram() {  
        // hier kommt das Hauptprogramm hin  
  
        kara.move();  
  
    }  
}
```

**McDonalds
machen
BigMacs**



Klassen definieren Datentypen:

McDonalds machen Burger. JavaKaraProgramme machen myProgram.

```
class McDonalds {  
  
    McDonalds(Ort) { ... }  
  
    BigMac gibMirBigMac() {  
        return machNenBigMac();  
    }  
  
}
```

Was macht McDonalds?

```
McDonalds macZurich = new  
McDonalds(zurich);
```

```
macZurich.gibMirBigMac();
```

```
class SchrittVorwaerts extends  
JavaKaraProgram {  
  
    public void myProgram() {  
        kara.move();  
    }  
  
}
```

Was macht JavaKara?

```
JavaKaraProgram program = new  
SchrittVorwaerts();
```

```
program.myProgram();
```