

2.– 5. September 2013  
in Nürnberg



# Herbstcampus

Wissenstransfer  
par excellence

## I, Robot

Programmiersession mit LEGO Mindstorms

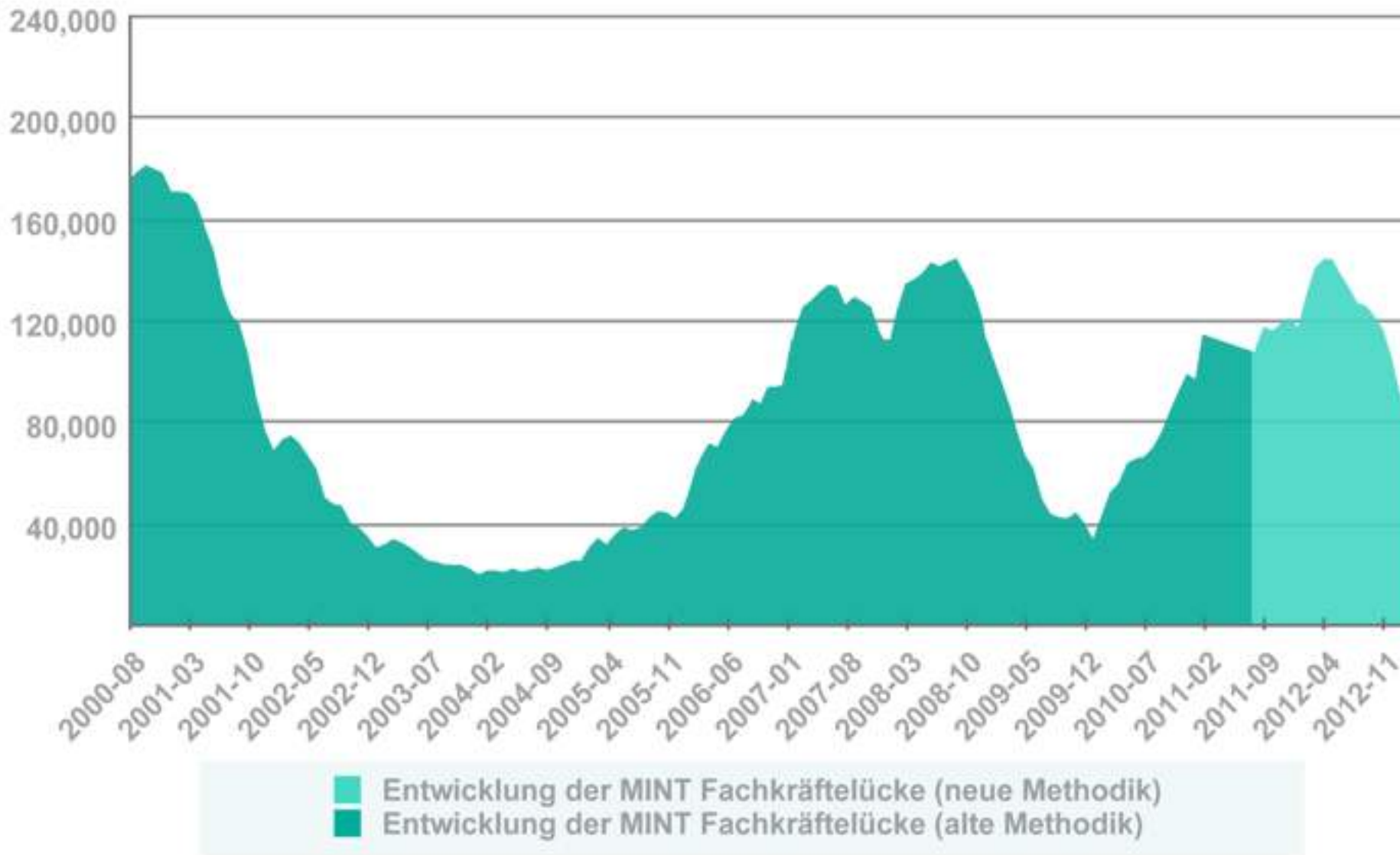
Jan Leßner

arvato IT Services



Vollständiger Klassensatz LEGO Mindstorm-Roboter

# Warum sollte man eine Computer-AG machen?







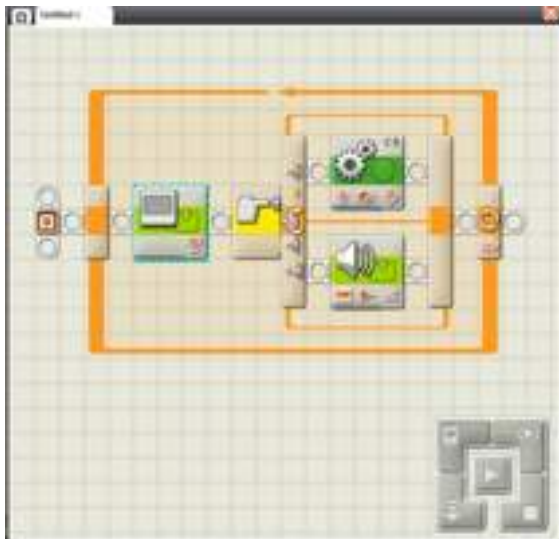
- Kids sind das Konsumieren mit Computern gewohnt, nicht aber das Produzieren.
- Erwartung an eigene Arbeitsergebnisse ist durch Computerkonsumgüter übersteigert
- Weg bis zum sichtbaren Erfolgserlebnis mit Standard-Hochsprachen ist sehr lang
- Konzentrationsfähigkeit der Kinder ist am Nachmittag reduziert
- Kinder haben kaum Erfahrung mit Teamarbeit und gegenseitigem Coaching\*
- Gelegentlich Zaungäste aus anderen AGs

5. bis 7. Klasse  
Einmal wöchentlich



Lieferung umfasst  
eine grafische  
Programmierung

Nach kurzer Zeit  
aufwendig und  
eingeschränkt



eclipse



Alternativ:  
Programmierung mit Java





[www.cebobot.com/cebobot](http://www.cebobot.com/cebobot)

```

Bewegung.java
TouchSensor touchSensor = new TouchSensor(SensorPort.S1);
UltrasonicSensor ultrasonicSensor = new UltrasonicSensor(SensorPort.S2);

pilot.setRotateSpeed(90);
pilot.setTravelSpeed(10);
// pilot.travel(25);
// pilot.arc(20.0, 90.0);
int anfangsEntfernung = ultrasonicSensor.getDistance();
pilot.travel(FAHRTSTRECKE, true);
while (pilot.isMoving()) {

    if (touchSensor.isPressed()) {
        pilot.stop();
        pilot.travel(-7);
        pilot.rotate(90);
        pilot.travel(FAHRTSTRECKE, true);
    }

    int aktuelleEntfernung = ultrasonicSensor.getDistance();
    System.out.println(anfangsEntfernung + " " + aktuelleEntfernung);
    if (Math.abs(aktuelleEntfernung - anfangsEntfernung) > 10) { // Plötzlicher
        aktuelleEntfernung = ultrasonicSensor.getDistance();
        System.out.println("Nachmessung: " + anfangsEntfernung + " " + aktuelle
    }
    if (aktuelleEntfernung > anfangsEntfernung + ABSTANDSDIFFERENZ ||
        aktuelleEntfernung < anfangsEntfernung - ABSTANDSDIFFERENZ ||
        aktuelleEntfernung < MINDEST_WANDABSTAND) {
        pilot.stop();
        if (aktuelleEntfernung < anfangsEntfernung || aktuelleEntfernung < MINDEST_WANDABSTAND) {
            pilot.rotate(KORREKTUR_WINKEL);
        } else {
            pilot.arc(-20, -KORREKTUR_WINKEL);
        }
    }
    // Entfernung nach Kurskorrektur messen. Die übernehmen wir nur dann
    // wenn der Wert nicht unerhöht groß ist. Die bedingte Übernahme stellt
    // Knicke in der Wand "herumstottert".
}
    
```





- Programmierung von Roboter-Autos auf dem Bildschirm
- In ansprechend animierter 3D-Welt
- **Java/C#-artige Sprache**
- Schnelle Erfolge
- Erstes Programm nur eine Zeile
- 50 vordefinierte Missionen
- **Didaktisch gut erklärt und aufgebaut**



Gleich mal  
ausprobieren!



- Hoher Spaßfaktor
  - Aufteilung in Kapitel und Missionen ist Einreiz zum „Leveln“
  - Coole Bots in höheren Leveln, z.B. Shooter, Grabber
- Quereinsteiger können ohne Anlaufzeit beschäftigt werden
  - Einstieg innerhalb von wenigen Minuten
  - Gute Missionsbeschreibungen → geringer Betreuungsaufwand
- Behutsame Einführung von Programmierkonstrukten
  - Schleifen, If / Else, Variablen, ...
  
- Nicht-Mitdenker bekommen nach ca. 25 Missionen trotzdem Schwierigkeiten

- Vorbereitung bzgl. Syntax und Konstrukte sehr weitgehend
- CeeBOT führt keine Funktionen ein, unterstützt das aber
- Selbstgebaute Erweiterungslevel schaffen Abhilfe
- CeeBOT ist ansatzweise objektorientiert  
→ Punkt-Notation ist bekannt
- Rest von OO ist schwach.  
→ prozeduraler Start in Java

The screenshot shows a window titled "Programmeditor" with a toolbar containing icons for file operations, undo, redo, cut, copy, paste, and a "get Com" button. The main text area contains the following code:

```
extern void object::Go() {
}

// Füge dieser Funktion einen Parameter für den Radius hinzu
void kreisbogen(int grad, int linksrum) {
    int drehwinkel = (linksrum > 0) ? 1 : -1;
    // Die Schrittweite musst Du in Abhängigkeit vom Radius berechnen
    int schrittweite = 0.2;
    repeat(grad) {
        move(schrittweite);
        turn(drehwinkel);
    }
}

extern
```

At the bottom of the window, there are buttons for "OK", "Abbrechen", and a set of navigation icons including a home icon, a right arrow, a person icon, and a "1" icon.



Erstes Ceebot-Programm

The screenshot shows a window titled "Programmeditor" with a toolbar and a code editor. The code in the editor is:

```
extern void object::Go()
{
    move(40);
}
```

The `move(40);` line is circled in red. Below the editor are buttons for "OK", "Abbrechen", and a "1" button.

Erstes Java/leJOS-Programm

The screenshot shows a window titled "Mission1.java" with the following code:

```
import lejos.nxt.Motor;
import lejos.robotics.navigation.DifferentialPilot;

public class Mission1 {

    public static void main(String[] args) {
        DifferentialPilot fahrwerk =
            new DifferentialPilot(5.7f, 11.2f, Motor.A, Motor.C);

        fahrwerk.travel(40);
    }
}
```

The `new DifferentialPilot` line and the `travel(40);` line are circled in red.

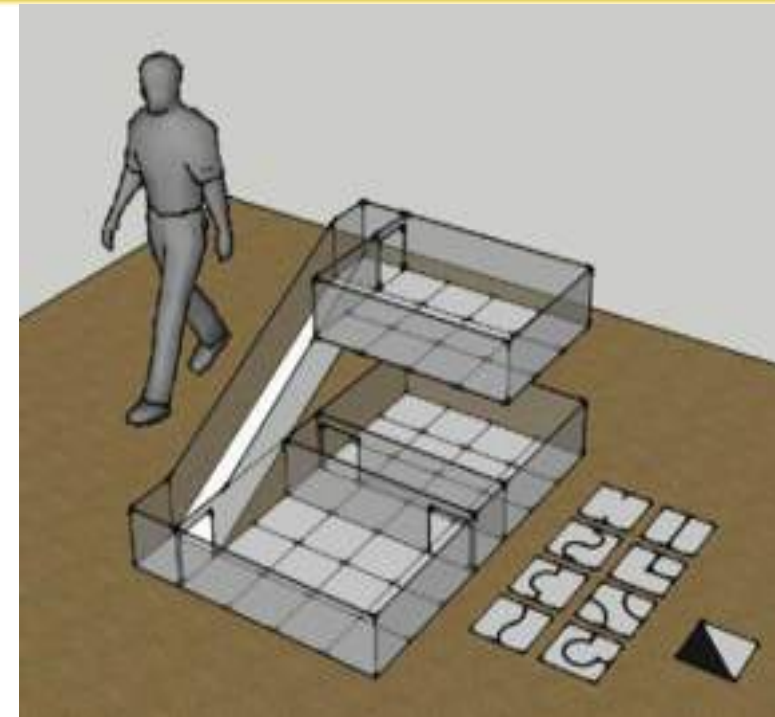
- Aussicht auf echten Bot hilft über neue Hürden hinweg
- DifferentialPilot erklärt sich gut aus Echtheit des Bots
- Installation Mindstorm-Treiber, leJOS, Eclipse und leJOS-Eclipse-Plugin vorbereitet
- Aber vorher natürlich: **Bot bauen! :-)**

Gleich mal  
ausprobieren!



- Komfort von Eclipse schafft stabilen Einstieg
- CeeBOT-Vorerfahrung erlaubt erste Berührung mit leJOS-API ohne große Fragezeichen
- „Verrücktes“ Fehlverhalten der Bots sorgt für Belustigung
- LEGO-Prinzip macht OO leicht verständlich
  
- Erste nennenswerte Hürden durch unsichere Messwerte
- Größere Missionen durch beschränkte Sensorik und schwierige Fehleranalyse rasch anspruchsvoll
- Hilfe für Konzeptionsarbeit: Programmieren in Pseudocode und auf Deutsch, viele, gut benannte Statusvariablen

- Wettkämpfe in unterschiedlichen Kategorien
- Spezielle Liga für Jugendliche
- Interessant für Einsteiger: Rescue
  - Roboter durch Parcours steuern und Gegenstand in sichere Zone bringen
  - Reglement liefert Stoff für kleine, separate Herausforderungen
- 2013 fast 400 Teams in 4 Quali-Tournieren





- Es stehen gut gemachte Lehrmittel zur Verfügung
- Nicht mehr nur für Nerds spannend
- Die besten Lernprogramme sind kostspielig
  - Mindestumsatz ca. 350,- €
  - CeeBOT Teen ca. 60,- €
- Kostenlose Angebote wie Scratch sind weniger attraktiv
  - Geringere Nähe zu „echten“ Sprachen
  - Deutlich mehr Betreuungsaufwand
- Einstieg schafft jeder, aber vielen wird es später zu kompliziert
- Die dauerhaft Interessierten haben tolle Möglichkeiten

**Vielen Dank!**

jan.lessner@bertelsmann.de