Prof. Dr. Nassrin Hajinejad

HTW Berlin 2024

Informatik I

Agenda

Über das Programmieren

In Anweisungen denken (Kara)

Objektorientiertes Programmieren

Class vs Objects

Kursinhalte und Prüfungsformat

APP

Digitalisierung

NOT ALL PEOPLE ARE SUPPOSED TO BE LOYAL TO YOU. SOME ARE MEANT TO COME ALONG AS A REMINDER TO WATCH THE COMPANY YOU KEEP.

YEARS WE THINK WE HAVE WITH PEOPLE

NUMBER OF TIMES WE'LL ACTUALLY BE WITH THEM

Problem today is don't cherish good people, to use them.

Private people will post all day, and still have a life you know nothing about.

Stay away from people who act like a victim in a problem they created.

Once i've addressed the situation multiple times and there's no changed behavior, I stop caring. It's above me now. People treat you exactly how they feel about you. You heard me & seen how it affected me the first time. At some point in life, you have to know your worth.

I really hope we'll never become strangers.

Pain changes people.

to live eople leave

Doctolib

Terminart

Gesetzlich

Verfügbarkeiten

Karte anzeigen

1183 Ergebnisse

Buchen Sie einen Termin online bei einem Zahnarzt oder ein/e Ärzt:in bzw. Therapeut:in mit diesem Schwerpunkt in Berlin

Herr Dirk-Armin Rohloff
Zahnarzt

Hubertusallee 13 14193 Berlin

Gesetzlich und privat Versicherte sowie Selbstzahlende

Die nächsten freien Termine

Vormittag	Di., 10.	Mi., 11.	Do., 12.
Nachmittag	Di., 10.	Mi., 11.	Do., 12.

Herr Dr. med. dent. Norman Kreller
Zahnarzt

Westfälische Str. 34 10709 Berlin

HTW Berlin - Campus Wilhel

Ähnliche Orte ansehen

Smartphone activities in selected countries 2023

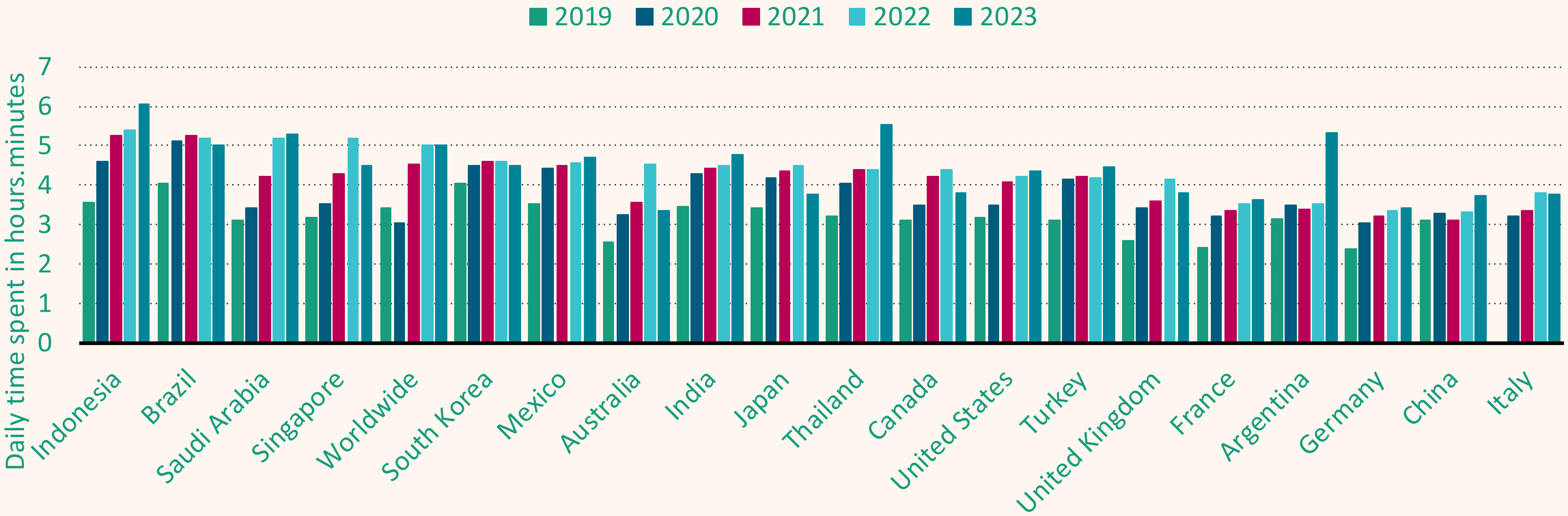
Smartphone user activities in selected countries worldwide from July 2022 to June 2023

	Australia	Austria	Brazil	Canada	China	France	Germany
Chatting or sending messages	54%	70%	84%	75%	67%	75%	70
E-mailing	73%	78%	72%	73%	35%	72%	75
Online banking	36%	65%	66%	63%	45%	60%	56
Listening to music	66%	53%	71%	60%	60%	56%	56
Watching videos	28%	52%	78%	60%	63%	56%	52
Searching for products	24%	56%	68%	54%	49%	53%	55
Buying products	59%	53%	65%	49%	60%	53%	55
Uploading videos or photos	26%	49%	58%	51%	57%	44%	48
Reading news pages	40%	56%	56%	39%	43%	44%	54

Note(s): Worldwide; July 2022 to June 2023; 18-64 years; respondents who use a smartphone
https://www.statista.com/statistics/1337846/smartphone-activities-by-country/
Source(s): Statista Consumer Insights; ID 1337846

Hours spent on mobile apps 2019-2023, by country

Number of hours spent per day using apps worldwide from 2019 to 2023, by country (in hours.minutes)



Note(s): Worldwide; 2019 to 2023; Android phones only

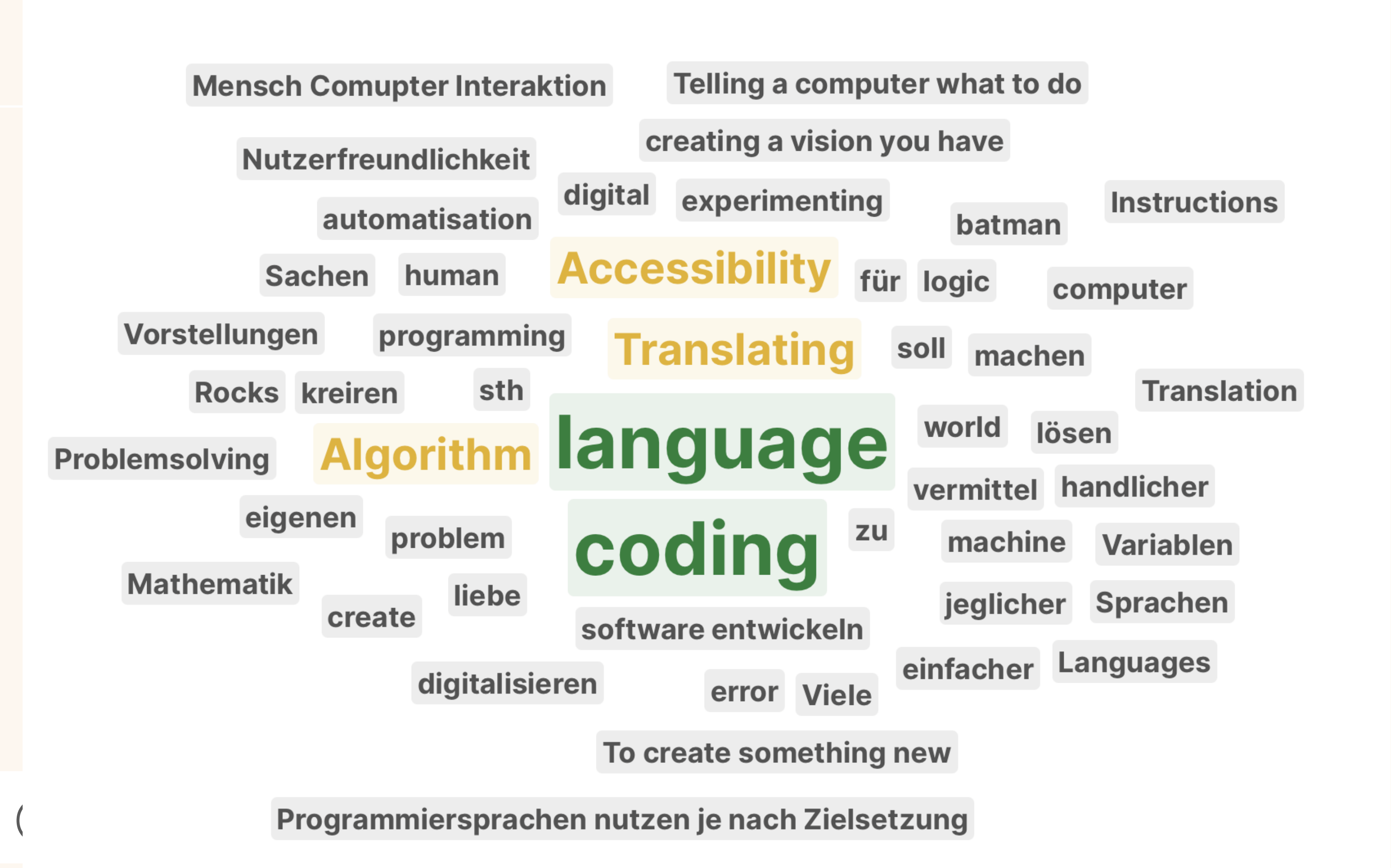
https://www.statista.com/statistics/1269704/time-spent-mobile-apps-worldwide/

Source(s): Data.ai; ID 1269704

slido

What is programming about?

Please download and install the
Slido app on all computers you use



Mensch Comupter Interaktion

Telling a computer what to do

creating a vision you have

Nutzerfreundlichkeit

digital

experimenting

batman

Instructions

automatisation

Sachen

human

Accessibility

für

logic

computer

Vorstellungen

programming

Translating

soll

machen

Rocks

kreiren

sth

Translation

language

world

lösen

Problemsolving

Algorithm

vermittel

handlicher

eigenen

problem

coding

zu

machine

Variablen

Mathematik

liebe

create

jeglicher

Sprachen

software entwickeln

einfacher

Languages

digitalisieren

error

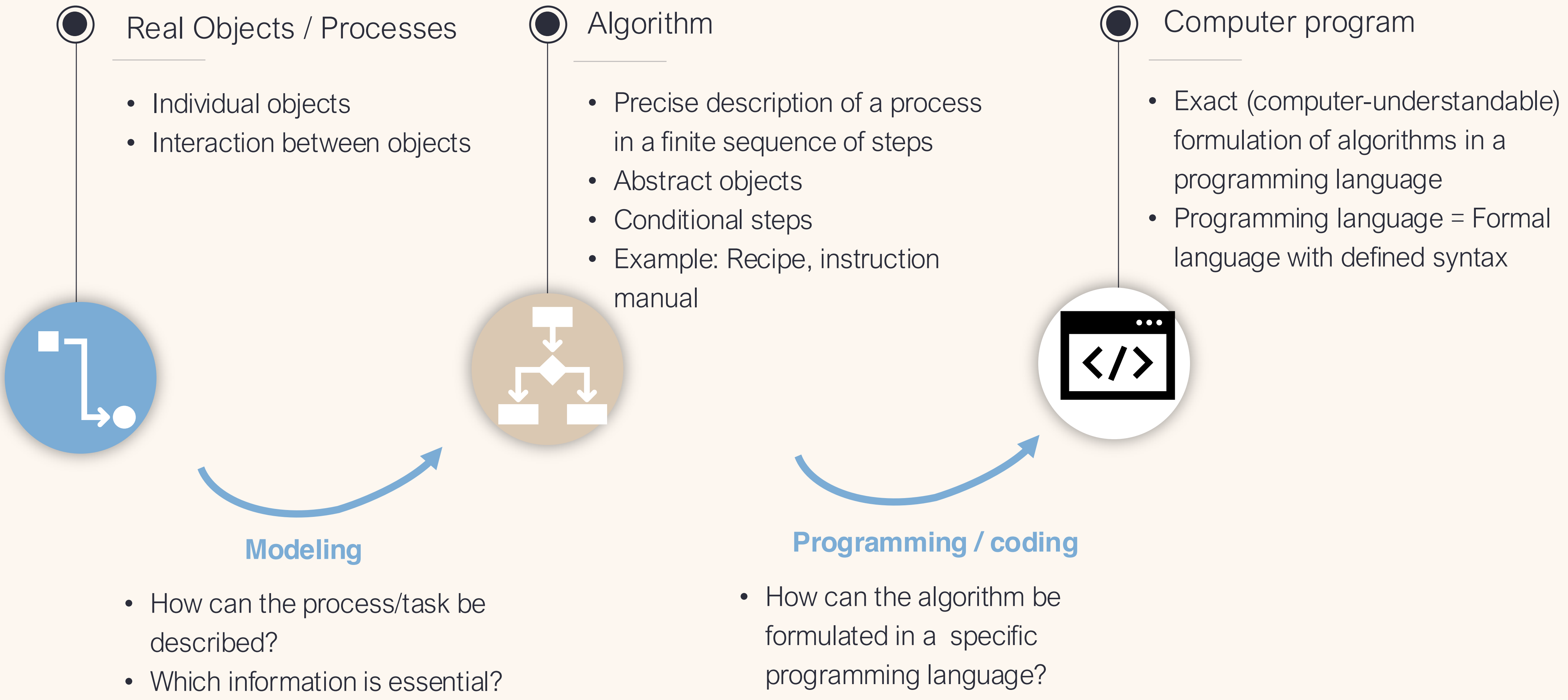
Viele

To create something new

(

Programmiersprachen nutzen je nach Zielsetzung

Translating real processes into computer-readable form



Real Objects / Processes

- Individual objects
- Interaction between objects

Modeling

- How can the process/task be described?
- Which information is essential?

Algorithm

- Precise description of a process in a finite sequence of steps
- Abstract objects
- Conditional steps
- Example: Recipe, instruction manual

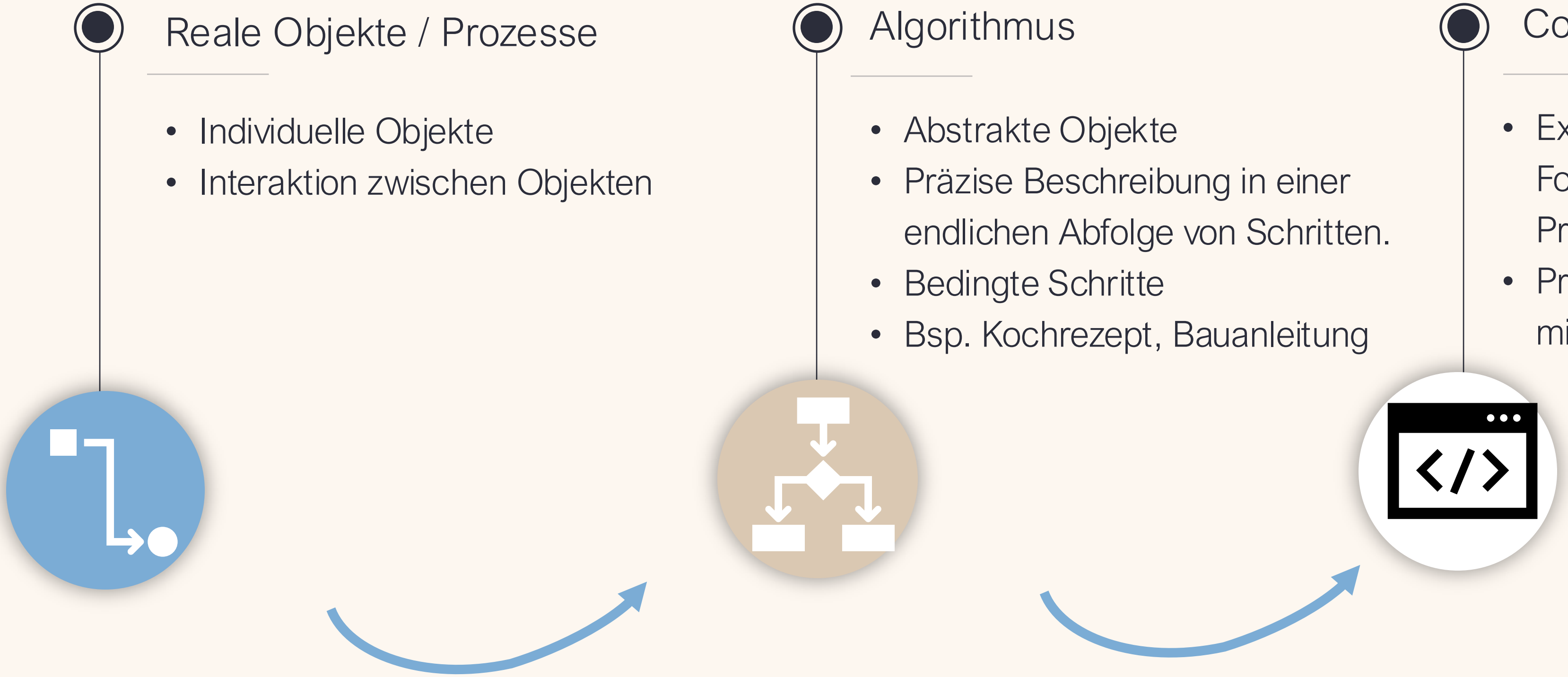
Programming / coding

- How can the algorithm be formulated in a specific programming language?

Computer program

- Exact (computer-understandable) formulation of algorithms in a programming language
- Programming language = Formal language with defined syntax

Reale Prozesse computerlesbar übersetzen



Reale Objekte / Prozesse

- Individuelle Objekte
- Interaktion zwischen Objekten

Modellieren

- Wie lässt sich der Prozess / die Aufgabe beschreiben?
- Welche Informationen sind wesentlich?

Algorithmus

- Abstrakte Objekte
- Präzise Beschreibung in einer endlichen Abfolge von Schritten.
- Bedingte Schritte
- Bsp. Kochrezept, Bauanleitung

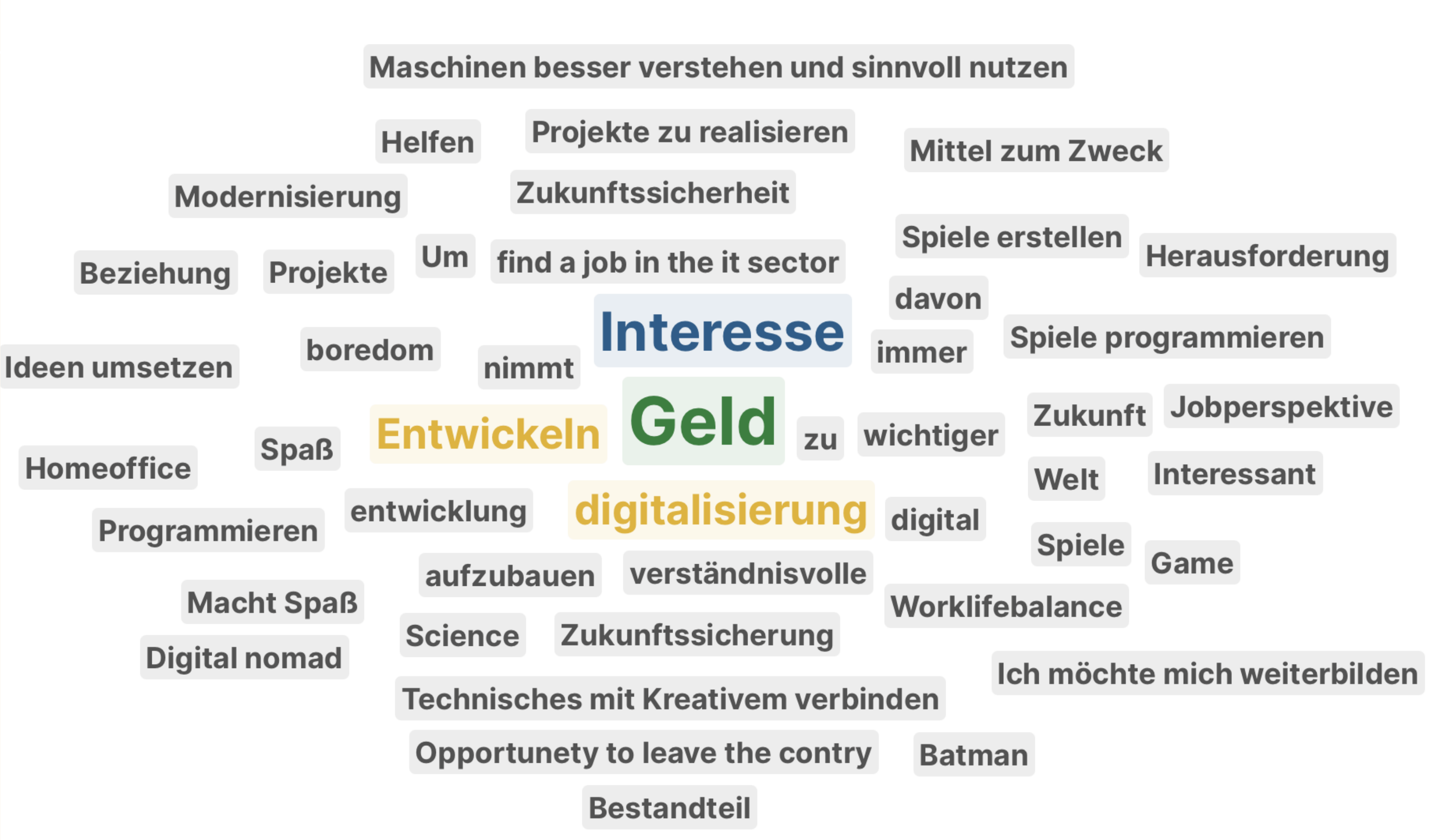
Programmieren

- Wie lässt sich der Algorithmus in der ausgewählten Programmiersprache formulieren?

Computer-Programm

- Exakte (computerverständliche) Formulierung von Algorithmen in einer Programmiersprache
- Programmierspreche = Formale Sprache mit definierter Syntax

Warum Lernen Sie Programmieren?



Maschinen besser verstehen und sinnvoll nutzen

Helfen

Projekte zu realisieren

Mittel zum Zweck

Modernisierung

Zukunftssicherheit

Spiele erstellen

Herausforderung

Beziehung

Projekte

Um

find a job in the it sector

davon

Interesse

immer

Spiele programmieren

Ideen umsetzen

boredom

nimmt

Geld

Zukunft

Jobperspektive

Entwickeln

zu

wichtiger

Homeoffice

Spaß

Welt

Interessant

Programmieren

entwicklung

digitalisierung

digital

Spiele

Game

aufzubauen

verständnisvolle

Worklifebalance

Macht Spaß

Science

Zukunftssicherung

Digital nomad

Ich möchte mich weiterbilden

Technisches mit Kreativem verbinden

Opportunety to leave the contry

Batman

Bestandteil

Warum Sie programmieren lernen…

- Grundlage des Studiengangs
- Verstehen reicht nicht, Sie müssen es anwenden können!

Computer

A device designed for automated data processing

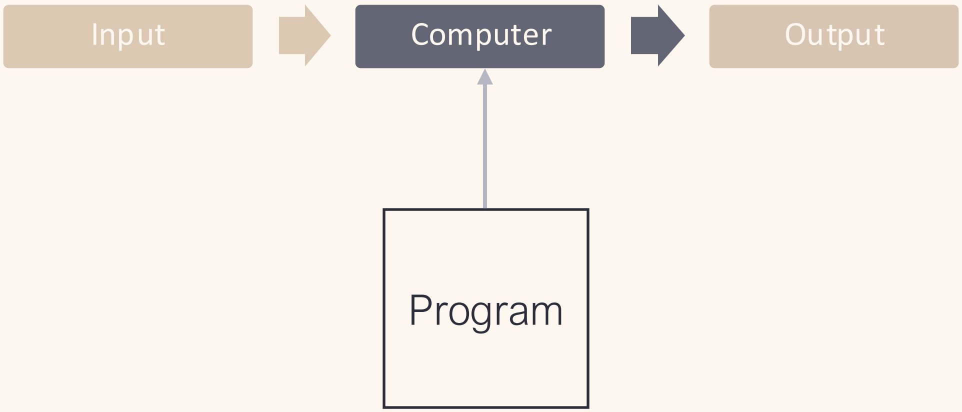
Hardware:

- The physical elements, including the central processing unit (CPU) and peripheral devices.

Software:

- The programs that control and manage the operations of the hardware.

Operation process:



Exploring algorithms

Kara teambuilding

Amateurs (with programming experience) come to the front

- Take materials for manual walkthrough

Beginners look for a free seat next to you

Amateurs choose a place next to beginners

Kara’s world

Actions

- move() //move one step in the current direction
- turnLeft() //change direction 90° to the left
- turnRight() // change direction 90° to the right
- putLeaf() // put a leaf on the current position
- removeLeaf() // remove leaf from the current position

Sensors:

- onLeaf() //returns true if the current tile contains a leaf
- treeFront() // returns true if the next tile in the current direction
- treeLeft() //returns true if tile to the left – again relative to current direction contains a tree.
- treeRight() //Same for treeRight, only to the right.

Kara’s world

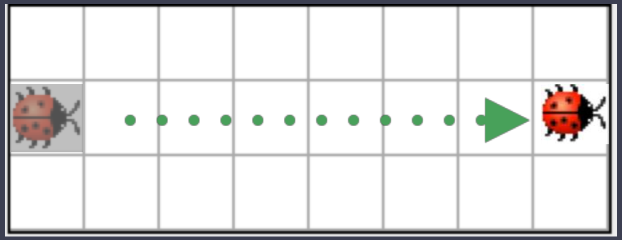
Actions

- move() = ein Kästchen vor
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise I



- Kara moves from left to right

Kara’s world

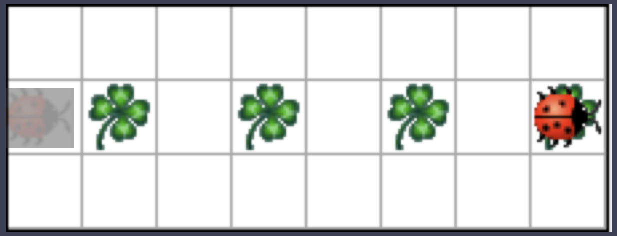
Actions

- move()
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise 2



- Kara moves from left to right
- And drops a leaf on every second tile

Kara's world

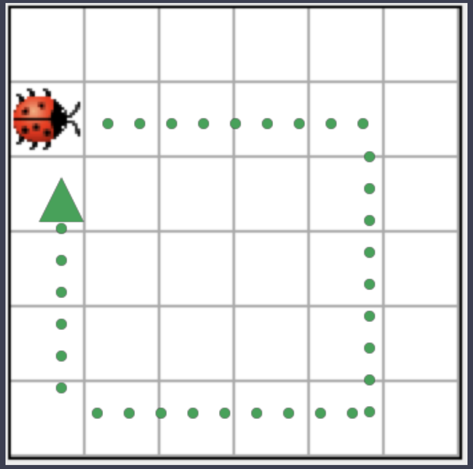
Actions

- move()
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise 3



- Kara walks in a square

Kara’s world

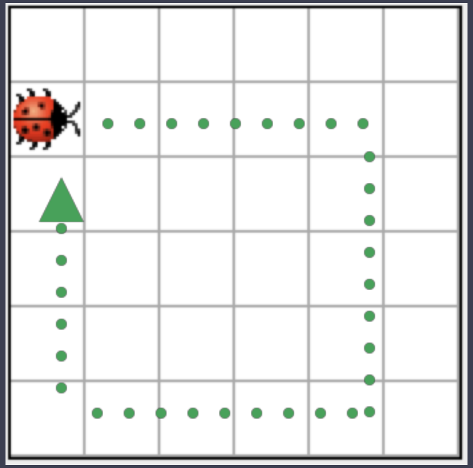
Actions

- move()
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise 3 optimized



- Kara walks in a square
- = 4 x walk one edge

Optimize:

- Look for repeating steps
- walkOneEdge()

Kara’s world

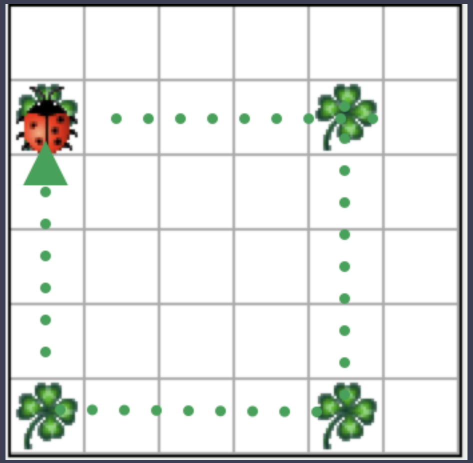
Actions

- move()
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise 4



- Kara walks in a square and drops a leaf on every corner

Optimize:

walkOneEdge()

Kara’s world

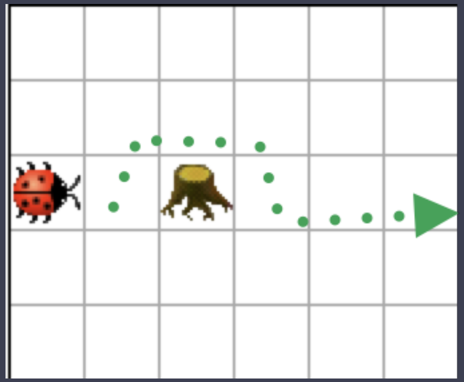
Actions

- move()
- turnLeft()
- turnRight()
- putLeaf()
- removeLeaf()

Sensors:

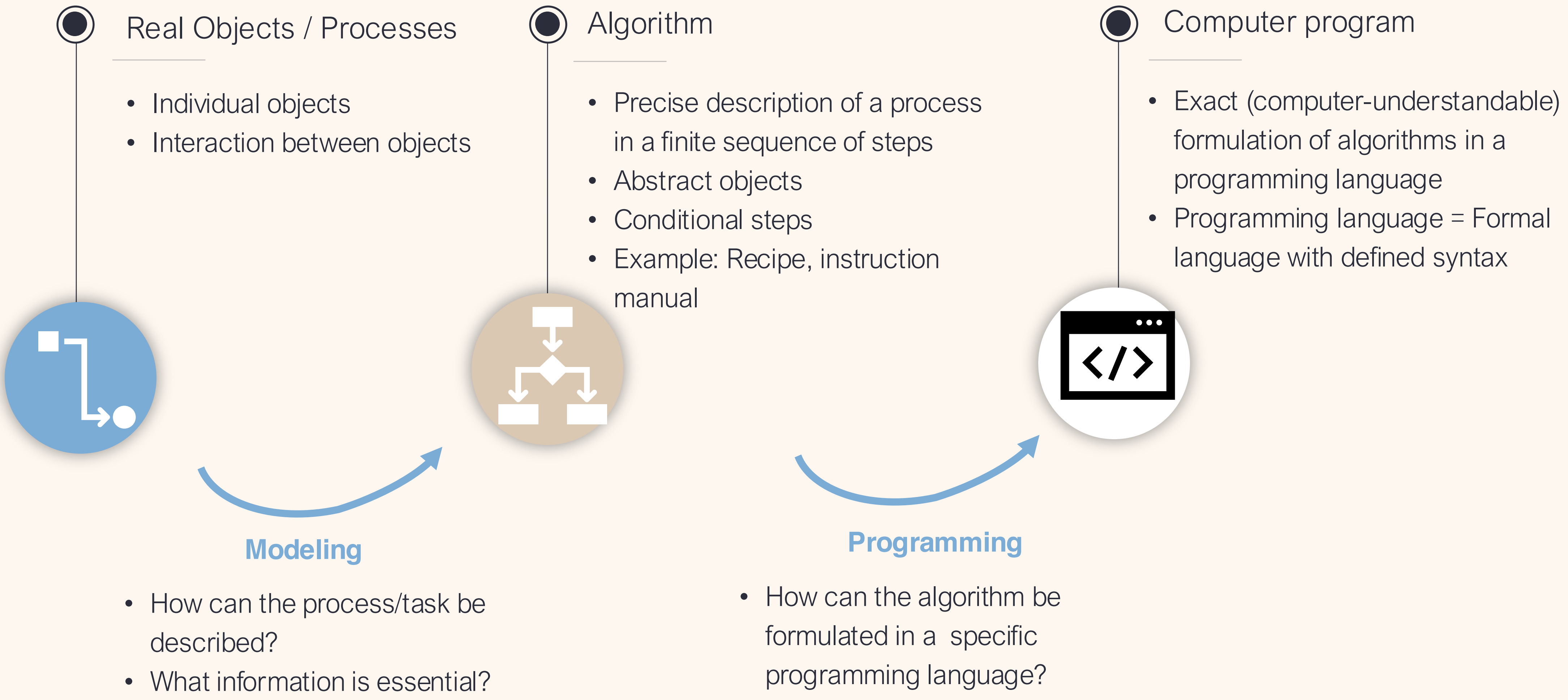
- onLeaf()
- treeFront()
- treeLeft()
- treeRight()
- mushroomFront()

Exercise 5



- Kara walks around a tree

Translating Real Processes into Machine-Readable Form



Real Objects / Processes

- Individual objects
- Interaction between objects

Algorithm

- Precise description of a process in a finite sequence of steps
- Abstract objects
- Conditional steps
- Example: Recipe, instruction manual

Computer program

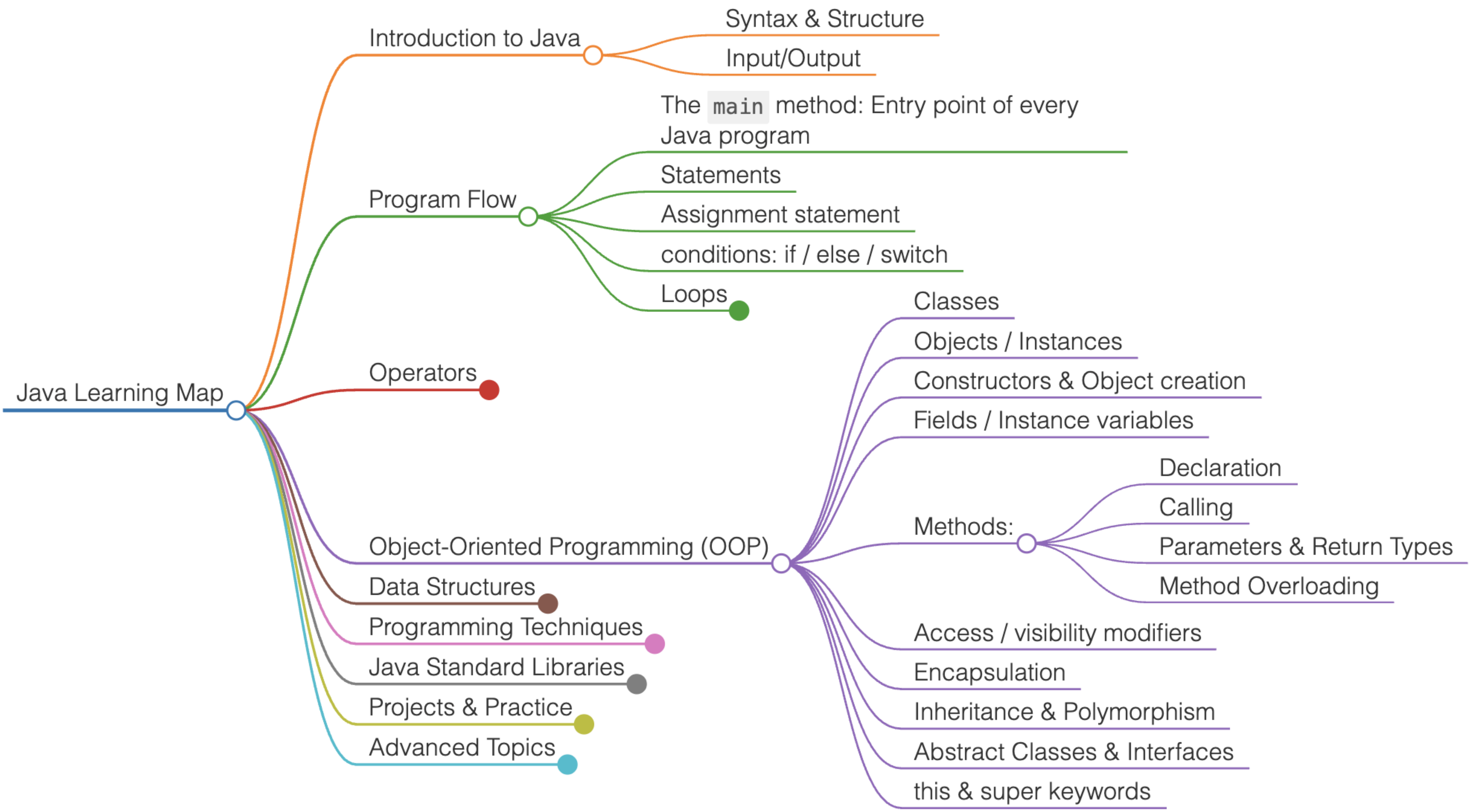
- Exact (computer-understandable) formulation of algorithms in a programming language
- Programming language = Formal language with defined syntax

Modeling

- How can the process/task be described?
- What information is essential?

Programming

- How can the algorithm be formulated in a specific programming language?



- Java Learning Map
  - Introduction to Java
    - Syntax & Structure
    - Input/Output
  - Program Flow
    - The `main` method: Entry point of every Java program
    - Statements
    - Assignment statement
    - conditions: if / else / switch
    - Loops
  - Operators
  - Object-Oriented Programming (OOP)
    - Classes
    - Objects / Instances
    - Constructors & Object creation
    - Fields / Instance variables
    - Methods:
      - Declaration
      - Calling
      - Parameters & Return Types
      - Method Overloading
    - Access / visibility modifiers
    - Encapsulation
    - Inheritance & Polymorphism
    - Abstract Classes & Interfaces
    - this & super keywords
  - Data Structures
  - Programming Techniques
  - Java Standard Libraries
  - Projects & Practice
  - Advanced Topics

Assignments

Lab assignments

- May be done in groups of max. 3, each of you should be able to explain
- Submit and explain during lab time **a week later**

Weekly Log

- your personal notes
- Upload in moodle individually, a **week later**

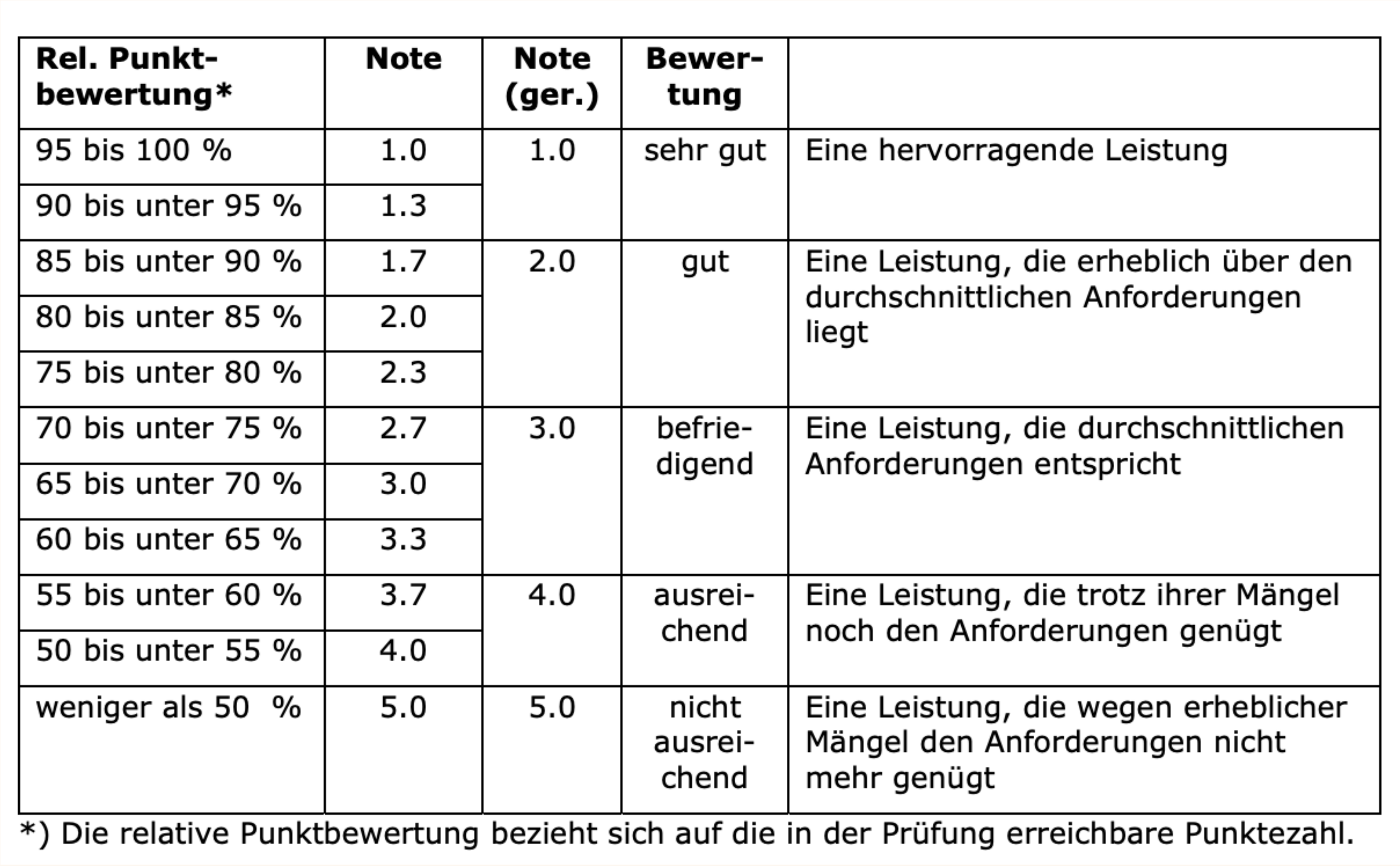
Info I

Grading

- Lab assignments (-1)
  Prerequisite for the exam
- Exam graded according to grading scale
- Weekly Logs (-3):
  5 Punkte
- Prüfungsform: Klausur 100%

https://home.htw-berlin.de/~kleinen/studies/grading/grading-scale/

https://www.htw-berlin.de/fileadmin/HTW/Zentral/Rechtsstelle/Amtliche_Mitteilungsblaetter/2013/04_13.pdf



Rel. Punkt-bewertung*	Note	Note (ger.)	Bewer-tung	
95 bis 100 %	1.0	1.0	sehr gut	Eine hervorragende Leistung
90 bis unter 95 %	1.3			
85 bis unter 90 %	1.7	2.0	gut	Eine Leistung, die erheblich über den durchschnittlichen Anforderungen liegt
80 bis unter 85 %	2.0			
75 bis unter 80 %	2.3			
70 bis unter 75 %	2.7	3.0	befrie-digend	Eine Leistung, die durchschnittlichen Anforderungen entspricht
65 bis unter 70 %	3.0			
60 bis unter 65 %	3.3			
55 bis unter 60 %	3.7	4.0	ausrei-chend	Eine Leistung, die trotz ihrer Mängel noch den Anforderungen genügt
50 bis unter 55 %	4.0			
weniger als 50 %	5.0	5.0	nicht ausrei-chend	Eine Leistung, die wegen erheblicher Mängel den Anforderungen nicht mehr genügt

*) Die relative Punktbewertung bezieht sich auf die in der Prüfung erreichbare Punktezahl.

Weekly Log

Week 1: xyz

Date: ______________________________

Topics Covered:

- …

Key Concepts Learned:

- …

Challenges Faced:

- …

Solutions/Resolutions:

- …

Additional Notes/Resources:

How to contact me

Contact

Please use your HTW email account for all correspondence.

Nassrin.Hajinejad@htw-berlin.de

Room: WH C 645

Moodle collaboration room to reach other students.

Expectations

- Participate and be professional
- Take responsibility for the quality of your time
- Contribute to our learning community:
  - Ideas, resources, challenges
- The goal is to learn (not yet to perform)

Agenda

Recap

IDE & project files

Creating a project

Class, Object, instance variables, methods, constructor

Was haben Sie bereits über die Java Syntax (Regeln) kennengelernt?

Methoden enden mit parametern "()"

Methoden werden Anfangs klein geschrieben

Ganz am Ende () und ;

Vereinfachung von langen Methoden

Actions

Real objects

methode(){}

double

Semicolon

();

Algorithmus

algoritmen

public void act

{}

public boolean

Sensors

Klammern

Public void

float

Semikolon

int

bool

Public

Datentypen

Keywords

String

input

vorgangsweise

klare Anweisungen

Methoden

()

Verschiedene

public und private

Strukturen und abläufe

Ein taben

Vorgegebene und Referenz Datentypen

; an das Ende

Methoden enden immer mit "()"

Loops

Output

Batman

Methoden fangen mit Kleinbuchstaben an

Methoden beginnen mit kleinem Buchstaben

Keywords

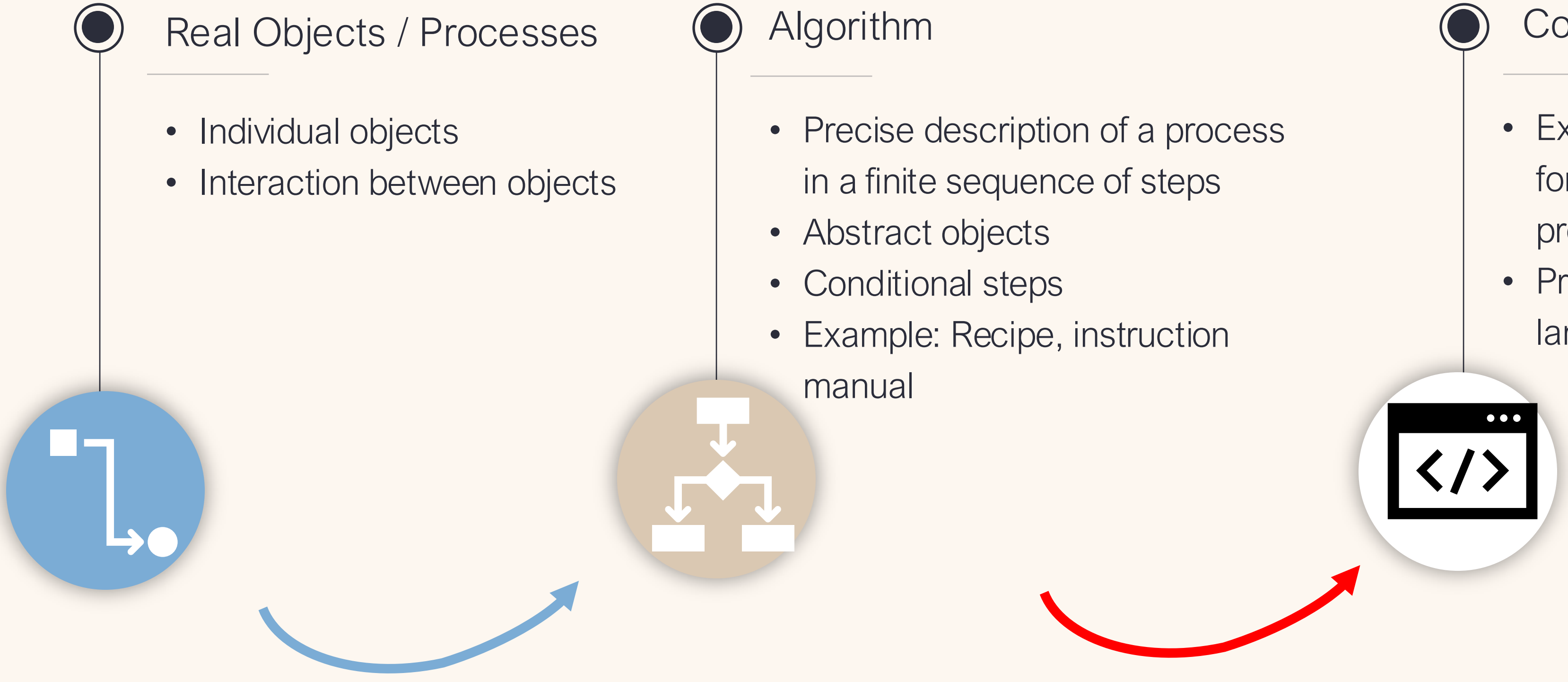
Words with a special meaning in the language:

- **`public`**
- **`class`**
- **`private`**
- **`int`**

Also known as *reserved words*.

Always entirely lower-case.

Translating real processes into computer-readable form



Real Objects / Processes

- Individual objects
- Interaction between objects

Algorithm

- Precise description of a process in a finite sequence of steps
- Abstract objects
- Conditional steps
- Example: Recipe, instruction manual

Computer program

- Exact (computer-understandable) formulation of algorithms in a programming language
- Programming language = Formal language with defined syntax

Modeling

- How can the process/task be described?
- Which information is essential?

Programming / coding

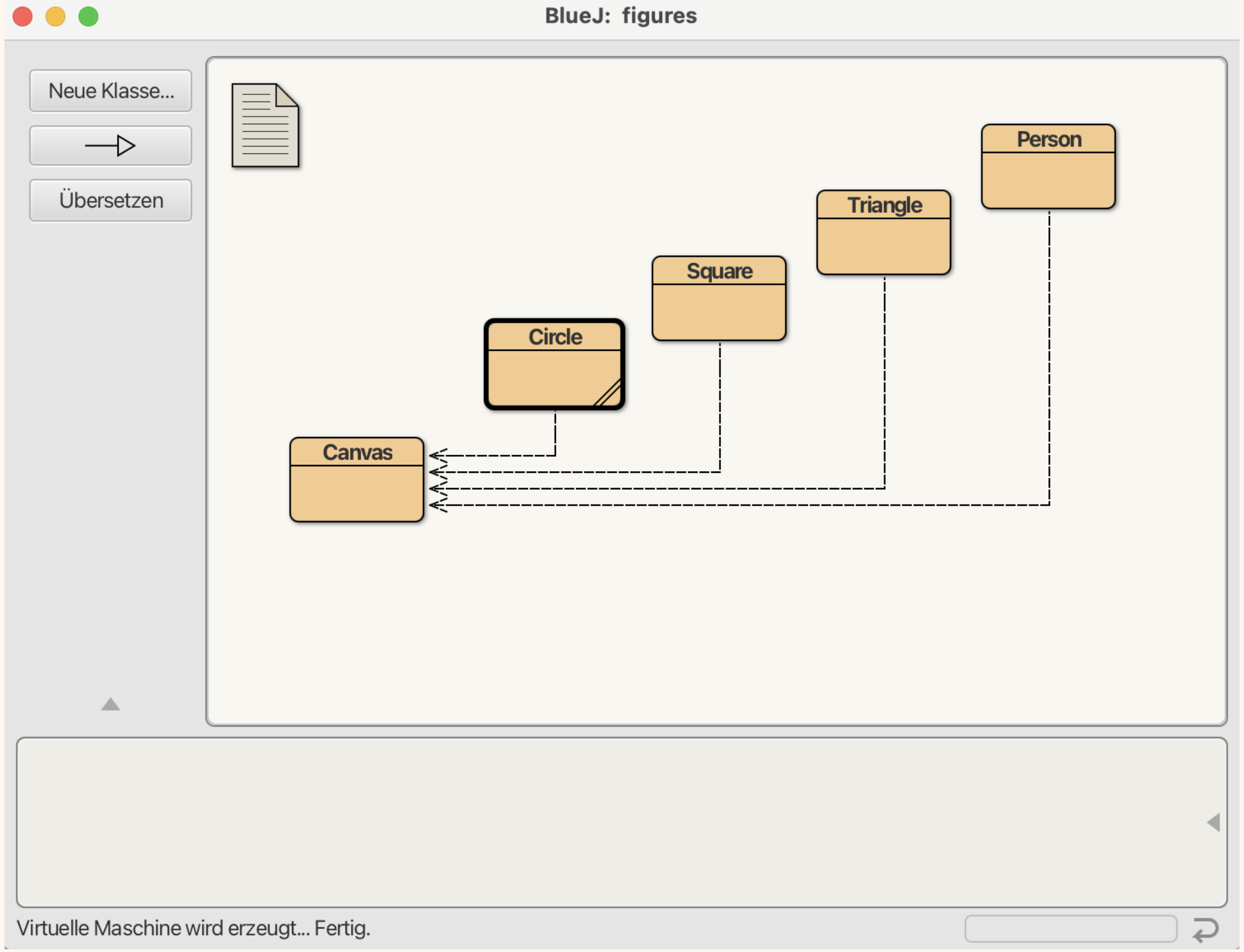
- How can the algorithm be formulated in a specific programming language?

BlueJ IDE

Implementing a program

An integrated development environment (IDE) is a software that helps writing a computer program developing a software, includes:

- Source-code editor, syntax highlighting
- Code completion (autocompletion)
- Debugger, to test and fix code
- Compiler translates source code from a high-level programming language (e.g. Java) to a low-level programming language (machine code) to create an executable program.

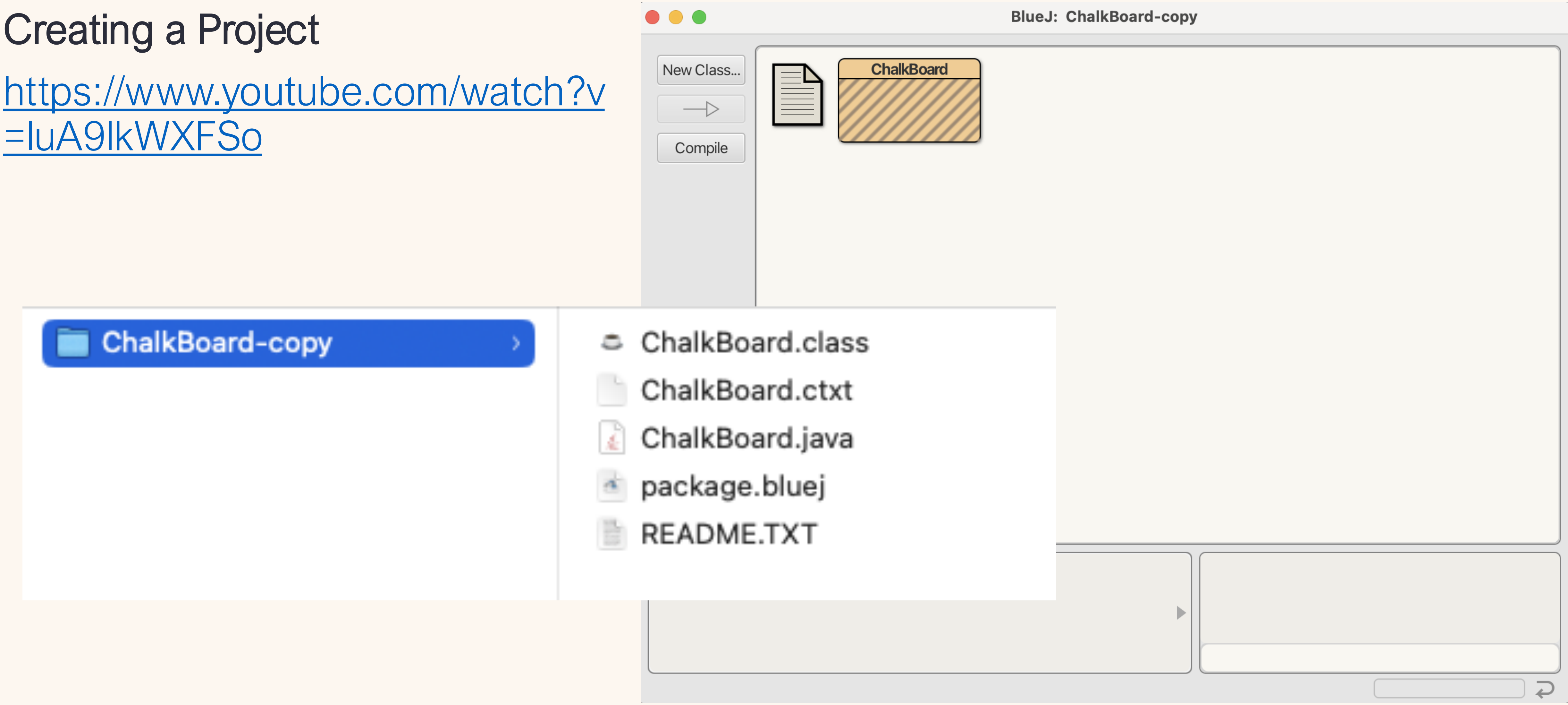


Virtuelle Maschine wird erzeugt... Fertig.

BlueJ

Creating a Project

https://www.youtube.com/watch?v=IuA9IkWXFSo



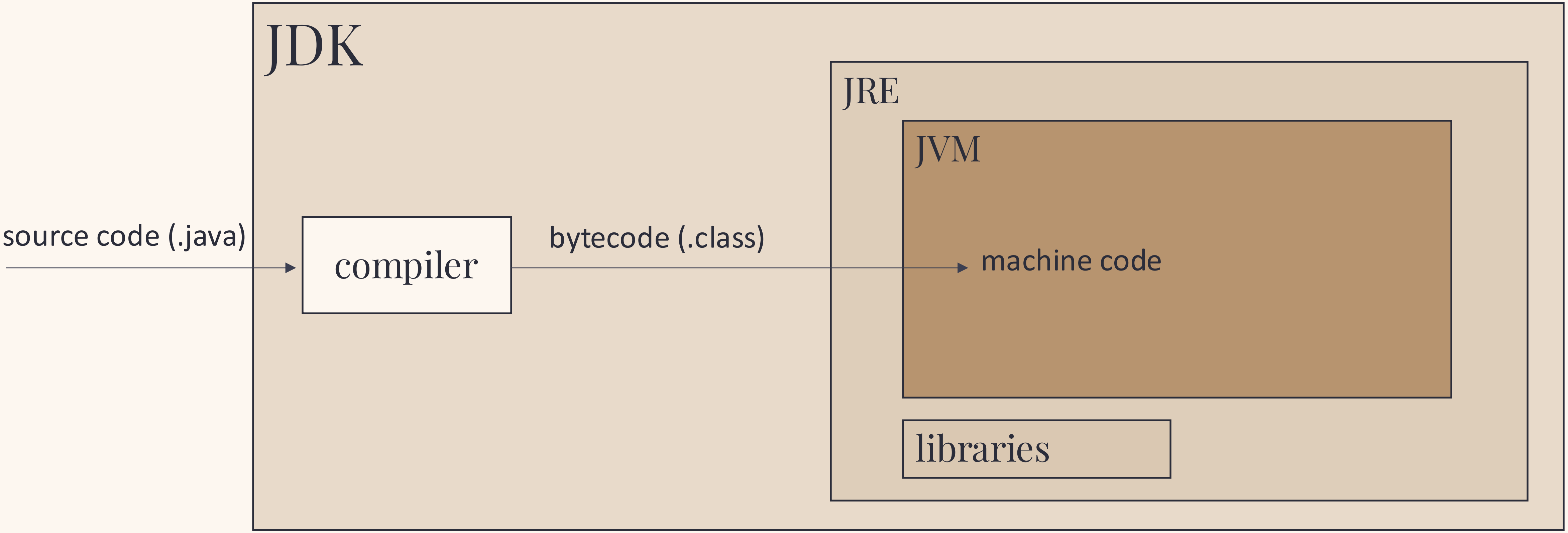
BlueJ: ChalkBoard-copy

ChalkBoard-copy

- ChalkBoard.class
- ChalkBoard.ctxt
- ChalkBoard.java
- package.bluej
- README.TXT

Java: Development Kit / Runtime Environment / Virtual Machine

- Portability : Write once, run anywhere (WORA)



A book (and IDE) to work with

David J. Barnes & Michael Kölling
Objects First with Java
A Practical Introduction using BlueJ
/Java Lernen mit BlueJ

6th (or 5th/4th) edition,

Pearson Education

Klassifikation

Programmiersprachen

Abstrahierungsgrad

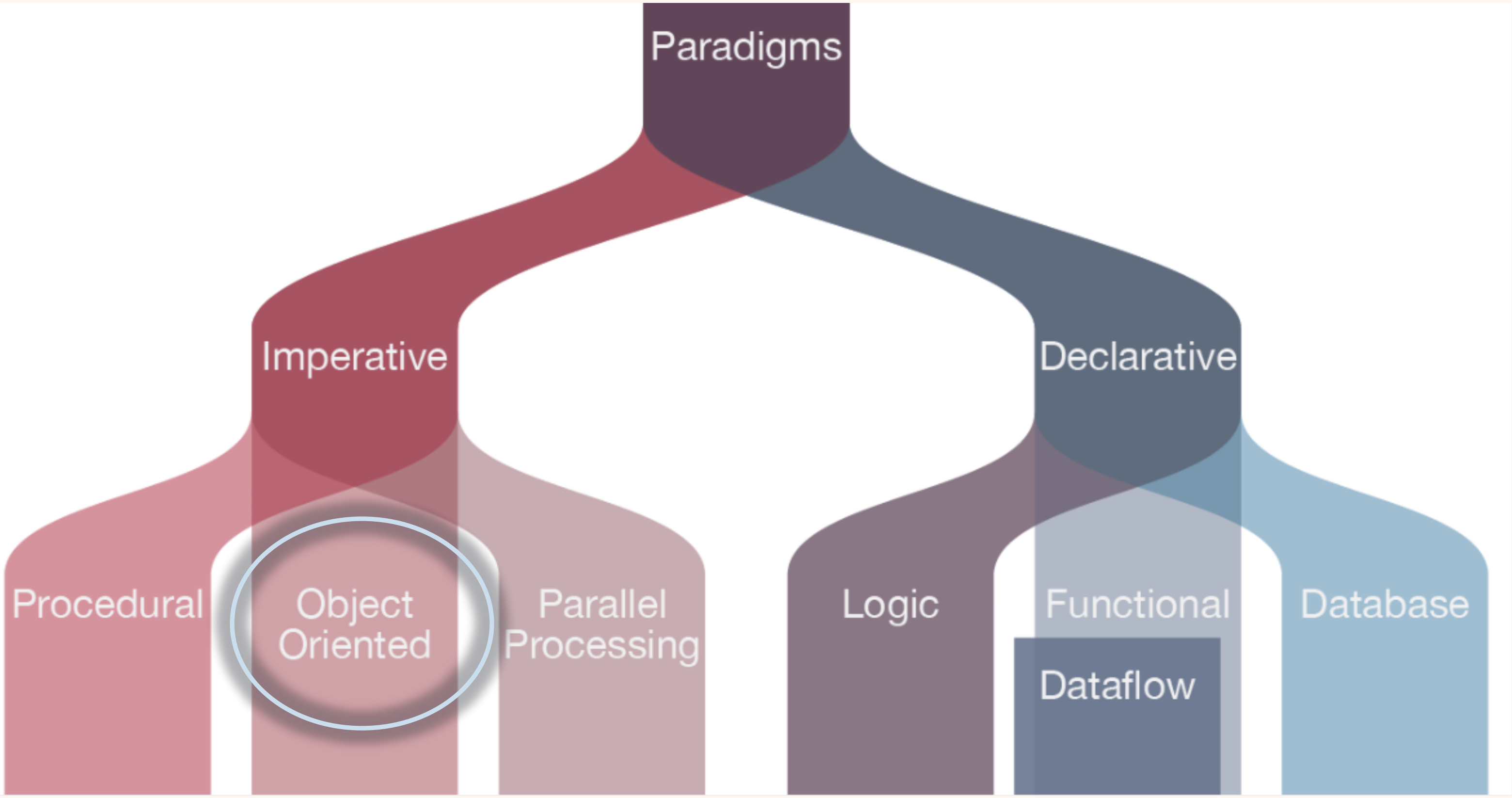
- maschinennah
- anwendungsorientiert

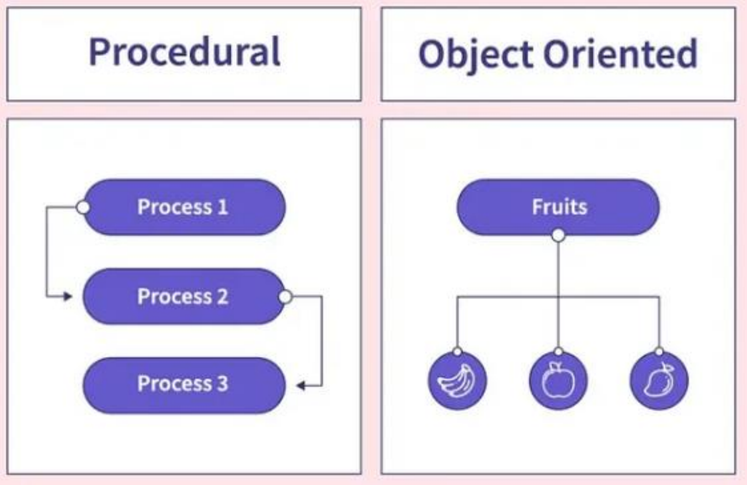
Komplexität

- problemspezifisch
- „general-purpose“

Paradigmen

- imperativ
- deskriptiv





https://medium.com/@kalyanasundaramthivaharan/object-oriented-
programming-oop-and-procedural-oriented-programming-pop-two-p
one-7a11cc143f0c

Classification

Programming Languages

Level of abstraction

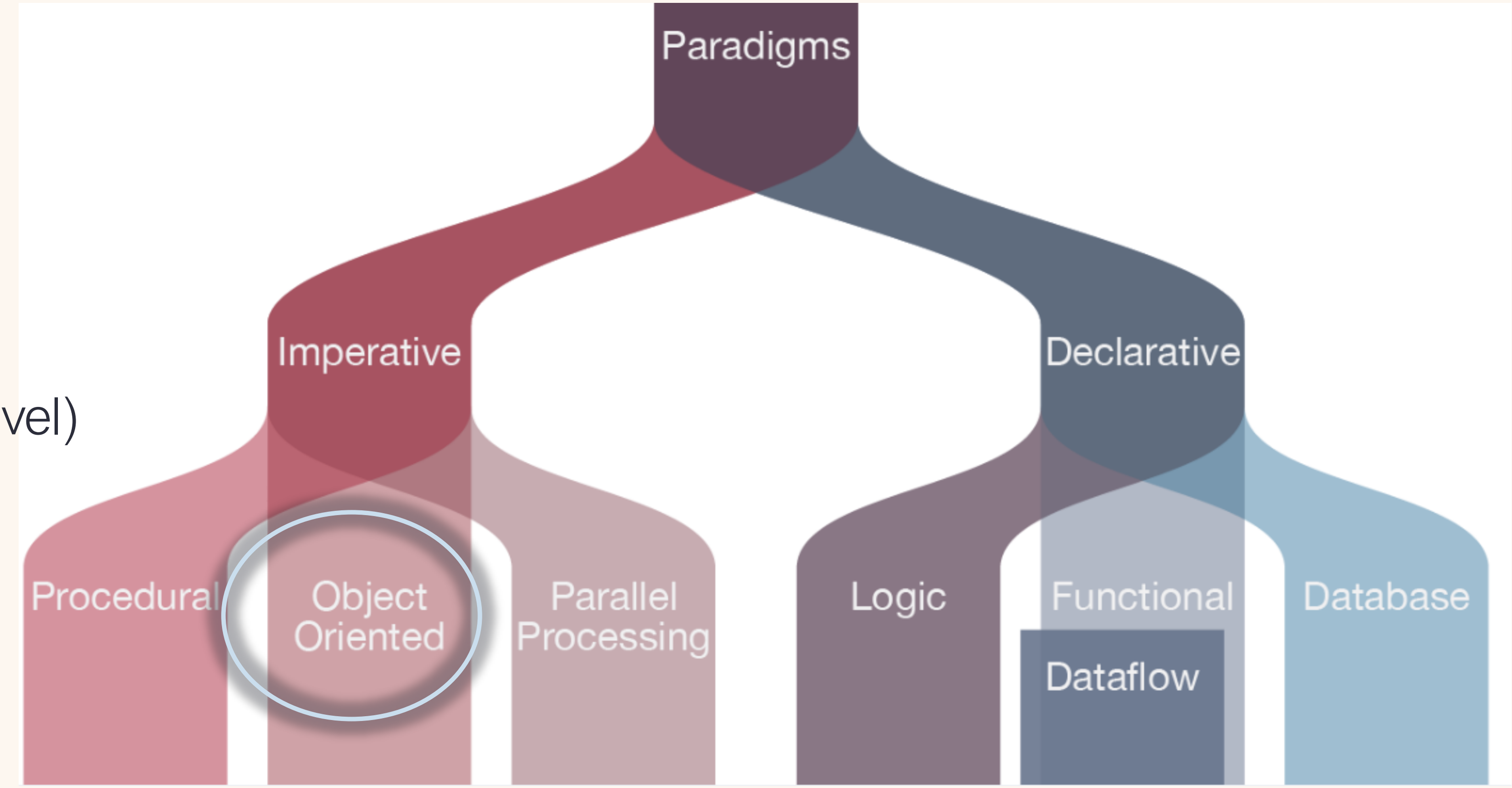
- close to the machine (low-level)
- application-oriented

Complexity

- problem-specific
- „general-purpose“

Paradigmen

- imperative (directing)
- declarative (descriping)



Digital objects represent reality and control real processes

OO-Programmierung

A More Natural Approach to Modeling Tasks/Processes

Objects (from everyday life to be automated)

- are items/living beings/situations/processes/abstract concepts…
- have properties, possess characteristics
- perform actions and offer services
- interact or maintain relationships with other objects.
- **solve sub-problems**, represented by their electronic counterparts.
- represent and address specific parts of a larger problem.

Advantages

- More natural modeling of problems
- Modularization / Data encapsulation
- Reusability of programs
- Easy extensibility of existing programs

Digitale Objekte bilden Realität ab und steuern reale Prozesse

OO-Programmierung

Natürliche(re) Vorgehensweise zum Modellieren von Aufgaben / Prozessen

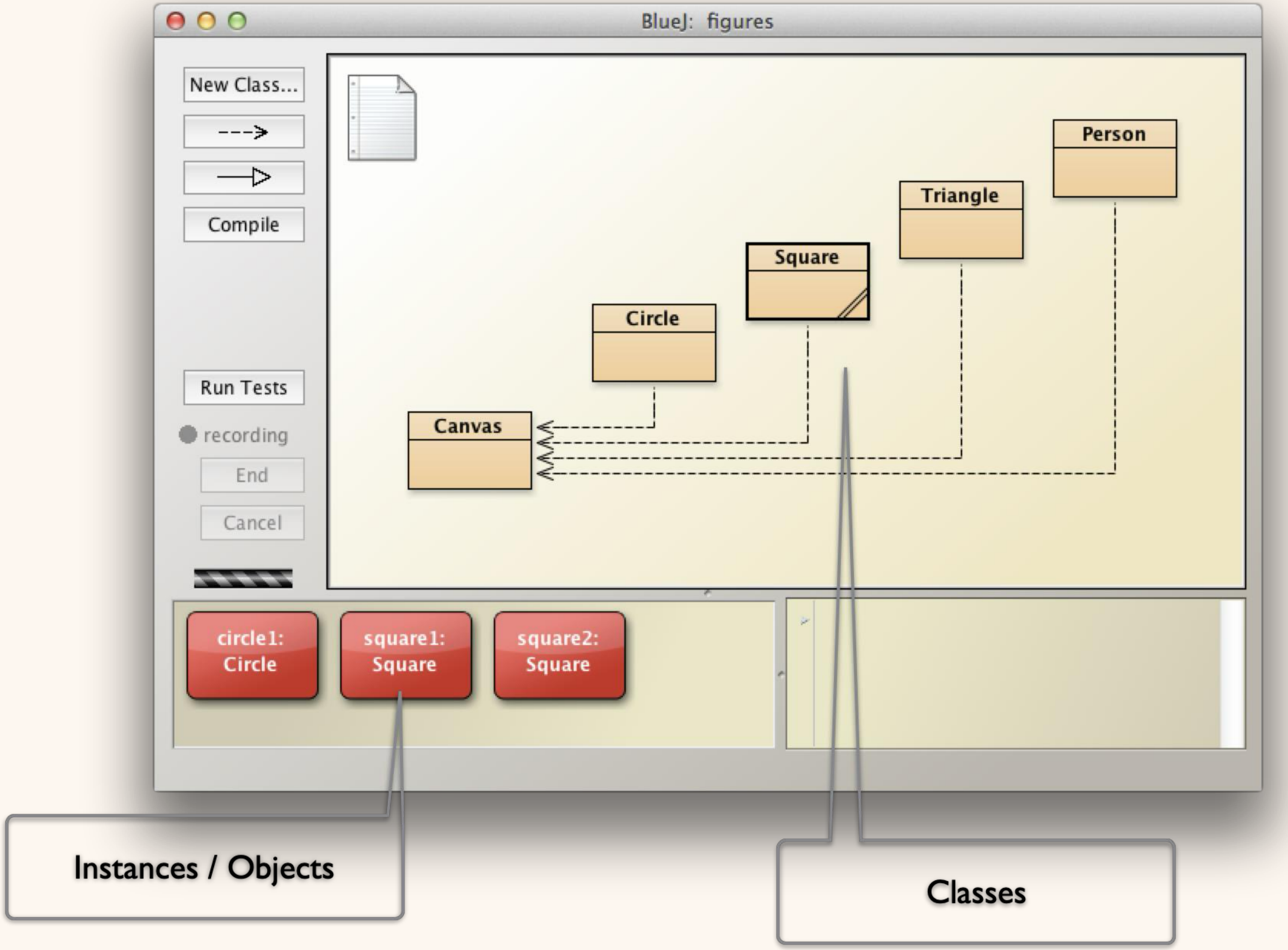
Objekte (des zu automatisierenden Alltags)

- sind Gegenstände / Lebewesen / Sachverhalte / Abläufe / abstrakte Konzepte...
- besitzen Eigenschaften
- haben Verhaltensweisen, können Dienste leisten
- kennen andere (auch andersartige) Objekte
- **Lösen Teilprobleme,** abgebildet auf elektronische Vertreter

Vorteile

- natürlichere Modellierung von Problemen
- Modularisierung / Datenkapselung
- Wiederverwendbarkeit von Programmen
- einfache Erweiterbarkeit existierender Programme

Class vs Instance



Elements of a Class

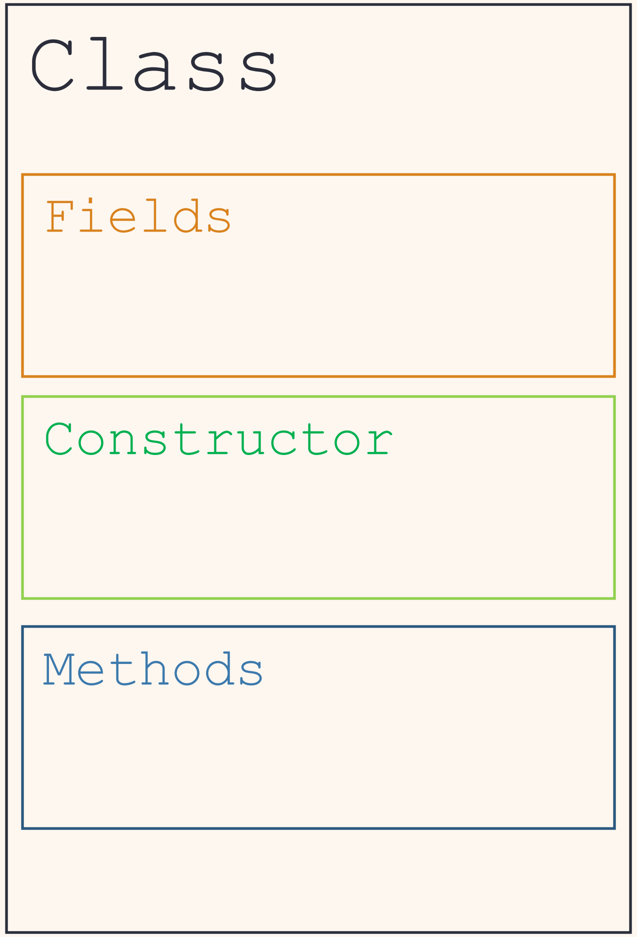
```
public class ClassName
{
    Fields

    Constructors

    Methods



}
```



Class

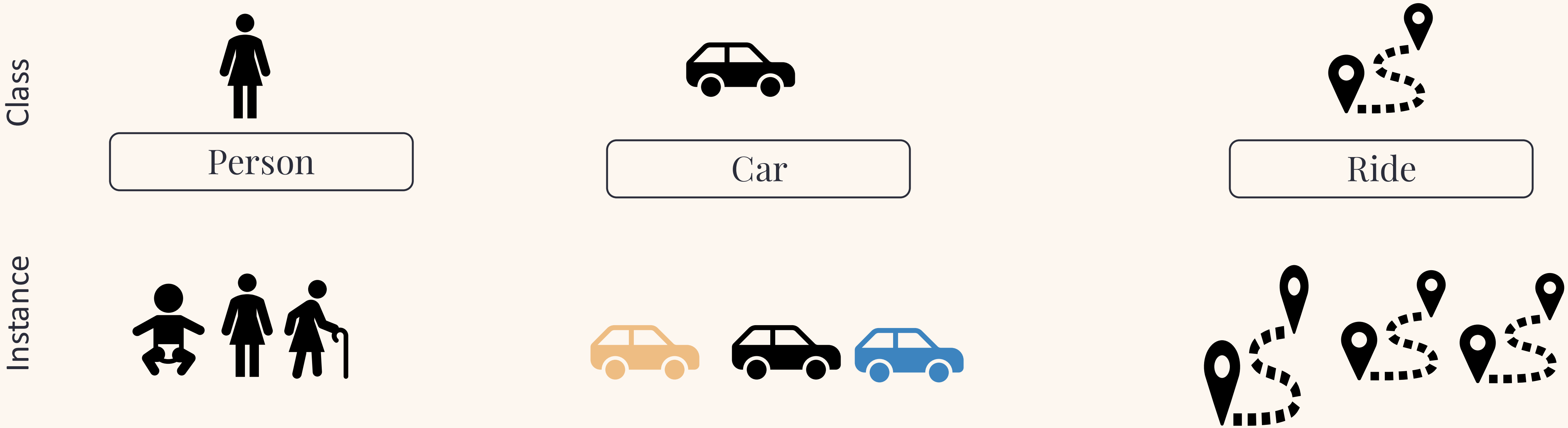
Fields

Constructor

Methods

OO-Programming

Class vs Object / Instance



Class

Person

Car

Ride

Instance

Instances of the same class share the same attributes (instance variables) and methods; they differ only in the values of their attributes (state)!