

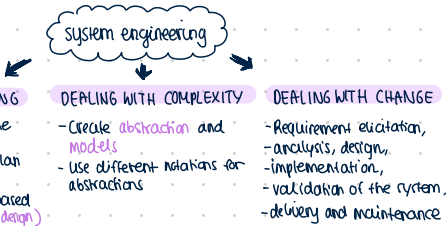
1. lecture : 15.04.21

OUTLINE:

- 1) Problem solving
- 2) Software engineering
- 3) Object-oriented programming

1) PROBLEM SOLVING

- get requirements from customer
- divide complex problem in smaller pieces (divide and conquer)
- then put pieces back together into a larger system



ABSTRACTION:

- complex systems hard to understand
- \neq 2 phenomena
- chunking (Group collections to reduce complexity)

- Allows us to ignore unessential details
- 1. - thought process (activity) where ideas are distanced from objects
- 2. = result (entity) of thought process

⇒ ABSTRACTION CAN BE EXPRESSED WITH A **MODEL**
= Abstraction of system



analysis

synthesis

→ understanding the problem's nature and breaking the problem into pieces

→ Putting the pieces into a larger structure

TECHNIQUES:

= formal procedures for producing results using some well-defined notation
Example: recipe, quick sort alg.

METHODOLOGIES:

= collection of techniques unified by philosophical approach
Example: cook book, functional decomposition

TOOLS:

= instruments or automated systems to accomplish a technique
Example: compiler, editor, debugger

2) Software Engineering

= collection of techniques, methodologies and tools that help with the production of a high quality software system development with a given budget before a given deadline while change occurs

PHENOMENON

= an object in the world of a domain as you perceive it

CONCEPT

= describes common properties of phenomena

A concept is a 3-Tuple

- **NAME**: distinguishes concepts from other ones
- **PURPOSE**: Properties that determine if a phenomenon is a member of a concept
- **MEMBERS**: set of phenomena which are part of the concept

Classification of phenomena into concepts
development of abstractions to answer specific questions about a set of phenomena

= ABSTRACTION

= MODELING

SYSTEM:

= organized set of communicating parts (natural system, engineered system)
- parts of system = systems (subsystems)

MODEL:

= Abstraction describing a system

VIEW:

= depicts selected aspects of a model

NOTATION:

= set of graphical or textual rules for depicting models and views
(Informal, Formal (UML))

3) object oriented programming

Type: int, boolean, string,...

Instance: member of specific type

Encapsulating: = creating classes for objects to define:

- Structure/State by using attributes
- Functionality/behavior by providing methods

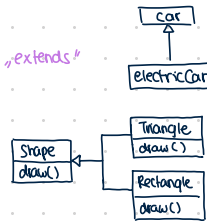
INHERITANCE:

• Java: Inheritance by defining subclasses using „extends“

ABSTRACT CLASS:

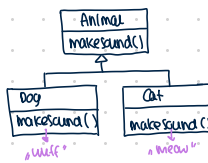
= class that cannot be instantiated

- ↳ holds common structure (state) or functionality
- can have multiple subclasses
- concrete objects to override methods



POLYMORPHISM:

= ability of object reference to take on many forms
• A super class reference is used to refer to any specific subclass



Animal = new Dog();
Animal.makeSound();

=> „wuff“

Animal = new Cat();
Animal.makeSound();

=> „meow“

