Introduction to UML

Martin Nečaský
Dept. of Software Engineering
Faculty of Mathematics and Physics
Charles University in Prague
What is UML?

- Unified Modeling Language
- standard graphical notation for modeling software systems from different viewpoints
  - architecture
  - data structures
  - processes
  - functions
  - states
  - communication
  - etc.
- and at different levels of abstraction
  - conceptual
  - implementation
What UML is not?

- replacement for textual programming languages
  - at least nowadays
- methodology
  - it does not say how to use it
- universal language
  - it is not a single language for any problem
Why UML?

- (standardized) graphical notation is a natural way of communication and understanding between different stakeholders and inside a development team
  - ISO standard
- allows to highlight important properties and features while avoiding unnecessary details
  - abstraction of software/data/processes
- analysts, designers, architects, developers started to use it widely
- well supported by different CASE tools
Classification of UML Diagram Types

Class Diagram
Profile Diagram
Composite Structure Diagram
Component Diagram
Deployment Diagram
Object Diagram
Package Diagram
Structure Diagram

Diagram
Behavior Diagram

Interaction Diagram
Timing Diagram
Interaction Overview Diagram
Communication Diagram
Use Case Diagram
Activity Diagram
Sequence Diagram
Basic Terminology
Model vs. Schema vs. Diagram

- **model** = modeling language
  - a set of constructs you can use to express something
  - e.g. the UML class model = \{class, attribute, association\}
  - e.g. the relational model = \{relation schema, attribute\}

- **schema** = expression in a given modeling language
  - an instance of a model
  - e.g. a relational schema = \{Person(name, email)\}

- **diagram** = schema visualization