# Advanced Usage of Z: Objects & Refinement

**Department of Distributed and Dependable Systems**

**D3S**

*Pavel Parízek*

CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

# Object-Z

- Main features
    - Classes & instances
    - Operations (methods)
    - Inheritance
    - History invariants
    - Dot notation

- Benefits
    - OOP: structure, modularity, reuse

# Refinement

- Goal: specification ➜ design ➜ code

- Operation refinement
- Data refinement

# Operation refinement

- Abstraction operation OpA
- Concrete operation OpC

- Weaker precondition
  - `pre OpA => pre OpC`

- Stronger postcondition
  - `post OpC => post OpA`

- Analogy: inheritance & method overriding
  - Object-oriented development

# Data refinement

- Goal: design concrete data structures

- Abstract schemas ➔ abstract states
- Concrete schemas ➔ concrete states

- Abstraction schema: abstract ↔ concrete

- Correct data refinement
  - `pre OpA ∧ Abs => pre OpC`
  - `pre OpA ∧ Abs ∧ OpC => Abs' ∧ post OpA`
  - `InitC => InitA ∧ Abs`

# Iterative step-wise refinement

- Target: complex systems

- Step
  - Refine some parts of the system model
  - Create procedures ➔ modular design

# Example

- Bank account system

- Abstract data structures
  - Mathematical model (clarity)

- Concrete data structures
  - Computer representation (performance)

# Literature

- G. Smith. The Object-Z Specification Language
    - [http://doi.org/10.1007/978-1-4615-5265-9](http://doi.org/10.1007/978-1-4615-5265-9)