

Extreme objects

Summary: This set of lectures gives full introduction into pure object-oriented approach to programming. Everything is an object – code, data, and functions. Minimalistic approach based on unification allows keeping minimal basis of key concepts for clarity and conciseness of the approach. It covered both passive and active behaviors of objects (sequential and concurrent executions) as well as approaching to the fully verified software using structured assertions (extended Design by Contract (c)). Proper type definition which comes from objects, multiple-inheritance with conflicts and multiple overriding coexists with simple atoms as 0 and 1 which are the foundation of everything

Lecture 1: Introduction to objects. Object structure. Attributes and fields

- Examples of objects
- Definition of object term
- Relation between computer memory and objects
- Key characteristics of any object
- Object structure. Attributes. Kinds of attributes

Lecture 2: Object life cycle and its operations

- Object creation and attributes initialization
- Life time loop – activations
- Destruction

Lecture 3: Group of objects form a type. Introduction of type concept

- Objects with identical structure form a type
- Special case of constants of different kinds
- Persistence of objects. Backbone of dynamic loading
- Kinds of type parametrization. Genericity

Lecture 4: Object equality. Object immutability. Shallow and deep

- How to compare objects
- What can be immutable
- Shallow and deep immutabilities

Lecture 5: Relations between objects. Refers and includes

- Reference and value semantics
- Cycles

Lecture 6: Object-class-module-type hierarchy

- Compile time and runtime relations
- Class-module difference and commonality

- Kinds of types

Lecture 8: Inheritance, overriding and member adaptations

- What is inheritance
- What is member adaptation while inheriting
- Kinds of overriding. Function to variable or constant
- Conformance

Lecture 9: Systematic assertions

- Predicates. Kinds of predicates
- Preconditions, postconditions, invariants and variants
- Alignment with inheritance
- Implications for practical usage. No more issue root cause triaging

Lecture 7: Active (concurrent) objects. Interactions between them

- Concept of processing element – thread, process, service
- Active objects never sleep
- How to support synchronous and asynchronous interactions between objects

Lecture 10: Control structures

- Member activation
 - o Assignment is a kind of special case
- Conditional
- Loops
- Exceptions. Kinds of exceptions
- Block
- Integration with assertions

Lecture 11: Tuples as a basic concept

- What is a tuple
- Every routine has 1 parameter
- Conformance

Lecture 12: Kernel classes

- 0 and 1 as two basic atoms
- Constant objects
- Regular expressions

Lecture 13: Overloading

- Routine names overloading

- Attribute names overloading
- Class names overloading
- Resolving ambiguities

Lecture 14: Compilation units and separate compilation

- Kinds of compilation units: script, program and library
- Clusters as areas for search
- Names controls
- Concept of the compilation context

Examination