Functional Programming

Conclusion

Functional Programming Principles

- Pure functions which are referential transparent
  - no external side effects
  - parameters and return values only
  - expressions (function applications) returning values
- Generics
  - generic data types abstract from concrete element types
  - generic functions abstract from concrete parameter types
- Function parameters
  - for parametrizing functions with functions
  - higher-order functions
- Working with functional data structures
  - immutable
  - value semantics
  - higher-order functions
- Functional exception handling
  - exceptional values as returns

Functional Programming Principles

- Function composition
  - composition operators creating complex functions from simpler functions
- Reduction
  - general scheme of computation
  - based on Monoids
  - parallel reduction
- Monads
  - generic way of building chains of computations
  - Monad type determines how computations are composed
- Lazy evaluation
  - by-name parameter passing or function objects
  - Java Streams as lazy sequences of elements
    - allowing computation chains
    - powerful collect operation with mutable containers
    - sequential and parallel processing