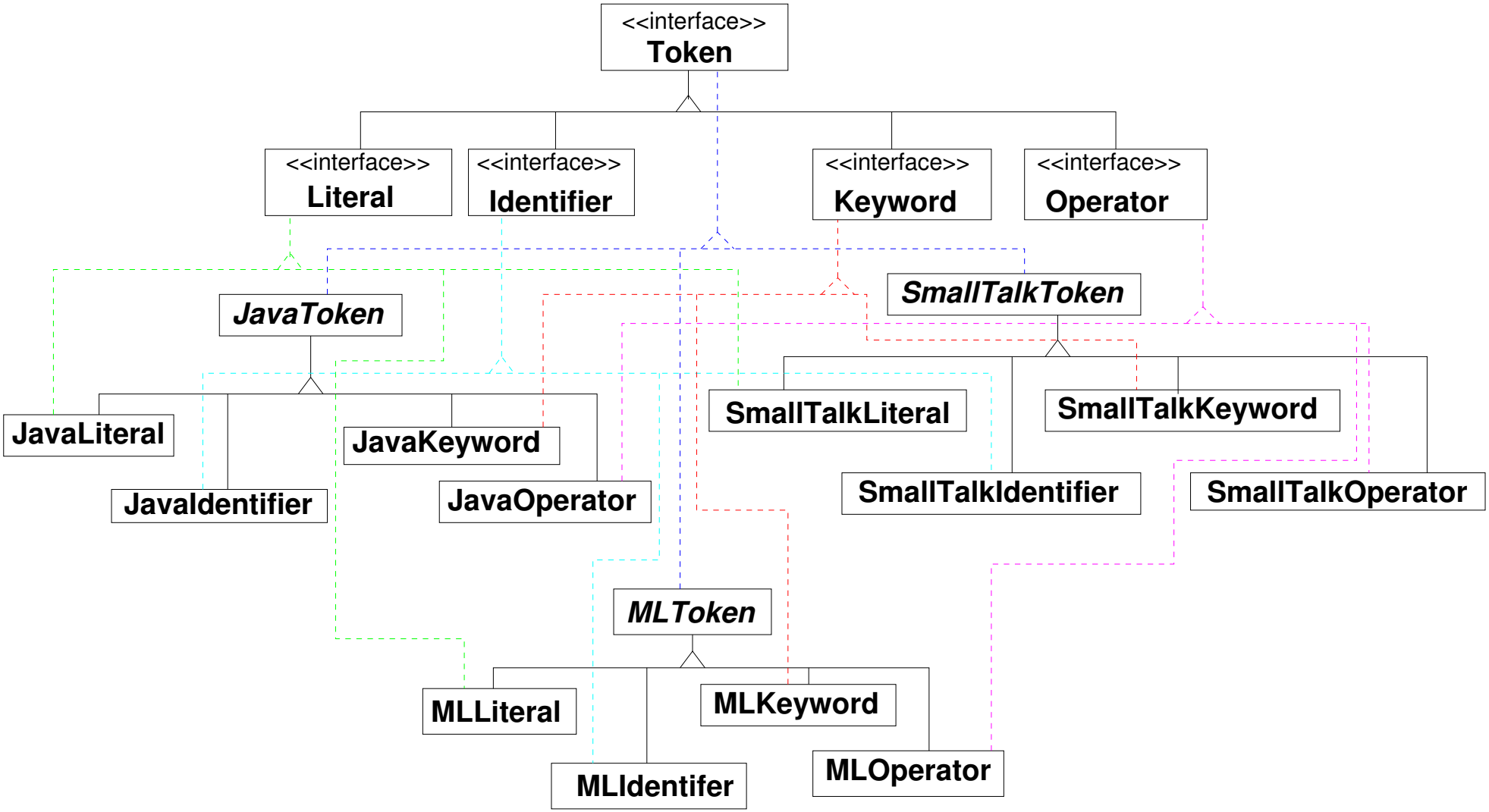
# CS 335 — Software Development

Creational Patterns (Abstract Factory and Builder)

Feb 26, 2014


<<interface>>
Token
<<interface>>
Literal
<<interface>>
Identifier
<<interface>>
Keyword
<<interface>>
Operator
JavaToken
SmallTalkToken
JavaLiteral
JavaKeyword
JavaIdentifier
JavaOperator
SmallTalkLiteral
SmallTalkKeyword
SmallTalkIdentifier
SmallTalkOperator
MLToken
MLLiteral
MLKeyword
MLIdentifer
MLOperator

# Tagged classes

**Effective Java Item 20: Prefer class hierarchies to tagged classes**

Occasionally you may run across a class whose instances come in two or more flavors and contain a *tag* field indicating the flavor of the instance. Such *tagged classes* have numerous shortcomings: boilerplate clutter, multiple implementations jumbled together, increased memory footprint from irrelevant fields of other flavors, constructors needing to set the tag field and initialize the right data fields with no compile-time checking, etc. In short, a tagged class is verbose, error-prone, and inefficient, a pallid imitation of a class hierarchy.

Abridged from Bloch, *Effective Java*, pg 100–101.

# Reflection

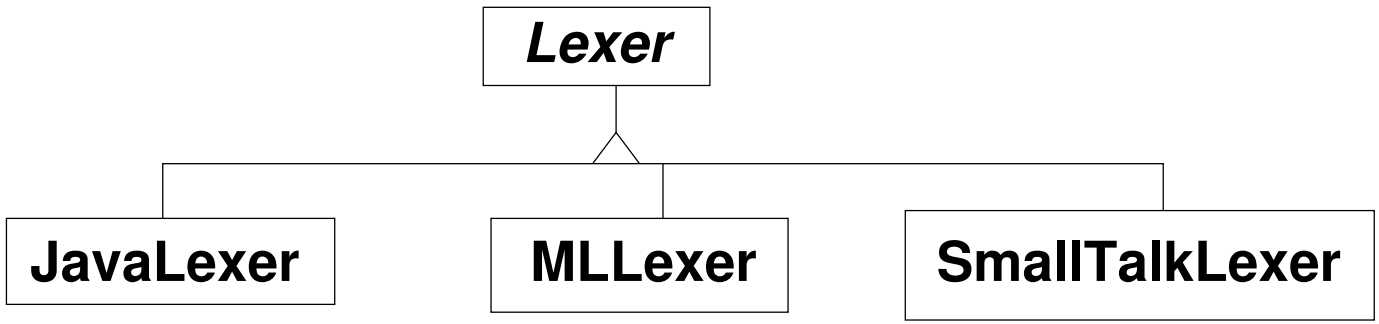
**Effective Java Item 53: Prefer interfaces to reflection**

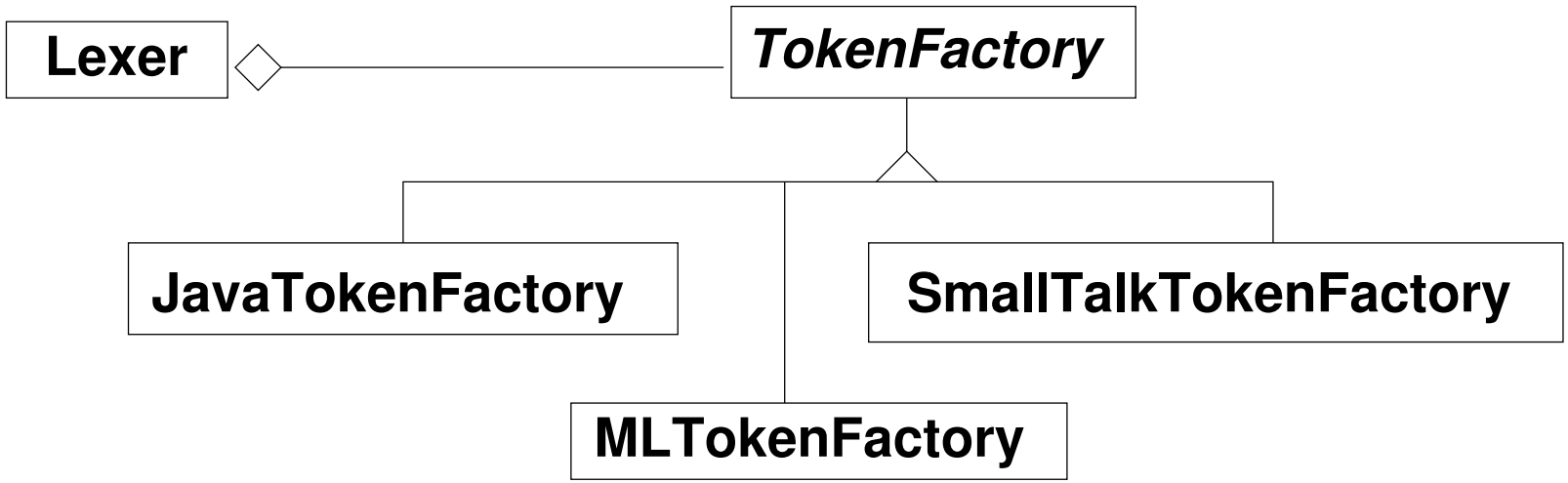
The power of reflection comes at a price: You lose all the benefits of compile-time type checking, the code required to perform reflective access is clumsy and verbose, and performance suffers.

Abridged from Bloch, *Effective Java*, pg 230.

# Template Method and Factory Method patterns

**Template Method.** Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template Method lets subclasses redefine (*refine*?) certain steps of an algorithm without changing the algorithm's structure. [DP, pg 235.]
**Factory Method.** Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses. [DP, pg 107.]


Lexer
JavaLexer
MLLexer
SmallTalkLexer


Lexer
TokenFactory
JavaTokenFactory
SmallTalkTokenFactory
MLTokenFactory

# Abstract Factory

Provide an interface for creating families of related or dependent objects without specifying their concrete classes. [DP, pg 87.]

# Builder

Separate the construction of a complex object from its representation so that the same construction process can create different representations. [DP, pg 97.]