Java vs. Ruby

Java	Ruby
Compiled and statically typed	Interpreted and dynamically typed
Object-oriented (but not pure)	Pure object-oriented
Single inheritance, multiple interfaces	Single inheritance, multiple mixin modules
Strongly typed	Strongly typed
C-derived syntax	Pascal-derived syntax
Statement-oriented	Expression-oriented
`Object` as universal super-class	Same
Fields and methods can be public, private, protected, or package	Instance and class variables are private; methods can be public (default), private, or protected
`this` reference	`self` reference
No special names for fields	Special names for instance and class variables, globals, and constants
Static fields and methods	Class variables and methods
Sub-type and parametric polymorphism	Duck-type polymorphism
Method overloading	No method overloading
Method overriding	Same
Operator `new` to create instances	Same
Abstract classes and methods	No abstract methods or classes
Final methods and classes	No final methods or classes; frozen objects
Nested classes	Nested classes are public
Anonymous classes	Singleton classes
No method aliasing	Method aliasing
No operator overloading	Operator overloading
Functional getters and setters	Attributes
No object extensions	Object extensions
Exception handling	Same
Packages for namespace control	Modules for namespace control
Regular expression library	Regular expression type
Standard control structures	Standard control structures plus iterators and statement modifiers
Large library	Same
No support for functional programming	Supports functional programming