Porting my programming language Onion from Java to Scala

Kota Mizushima
Some Company in Japan

About Myself

- Name: Kota Mizushima
- Country: Japan
- Got Ph.D. in University of Tsukuba
    - Parsing Algorithm
- Hired by some company (not related to Scala)
    - developing embedded software
- I'm interested in programming languages
    - Scala, Nemerle, Nice, Haskell, Ruby, and so on

Scala-chan and Java-co

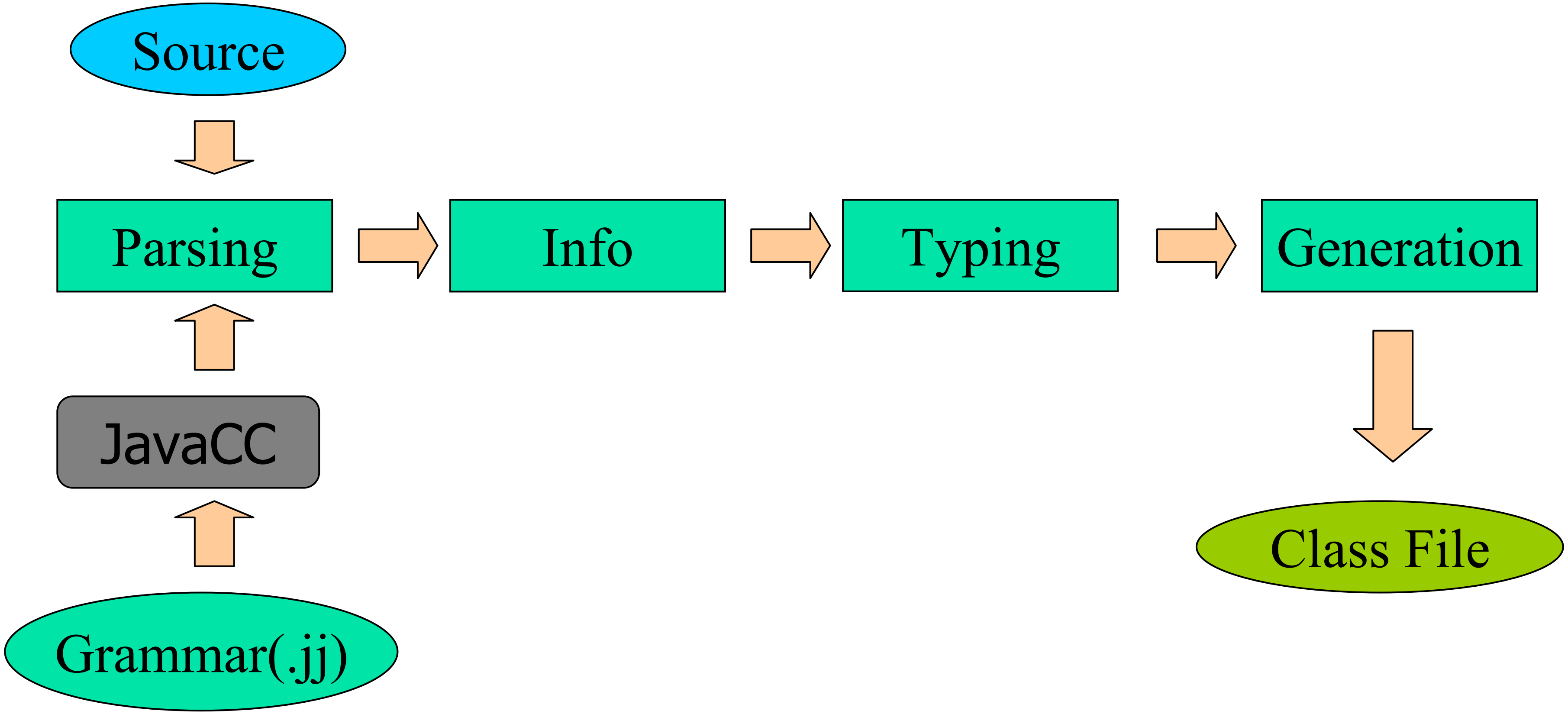
- Mascot characters of Scala and Java in Japan
- Java-co is a older sister of Scala-chan

Introduction to Onion

- Object oriented and statically typed
- Closures
  - #(arg: type){/* code */}
- Delegation
- Many syntax sugars and simple type inference
  - *x.y* → x.getY()
  - x = 1 → x: Int = 1
    - if there is no declaration such as x: Int above
- Compiled to Java class file
- Existing Java librarys is available

Structure of Onion Compiler

- LoC: 10000 lines over (was written in Java)



Why Scala?

- Writing compilers in Java is painful
    - Visitors lacks flexibility
    - No first class function, no pattern matching. etc.
- Scala is good for writing compilers
    - Pattern matching
    - First class functions and many higher-order methods
    - Algebraic data type by sealed trait + case class

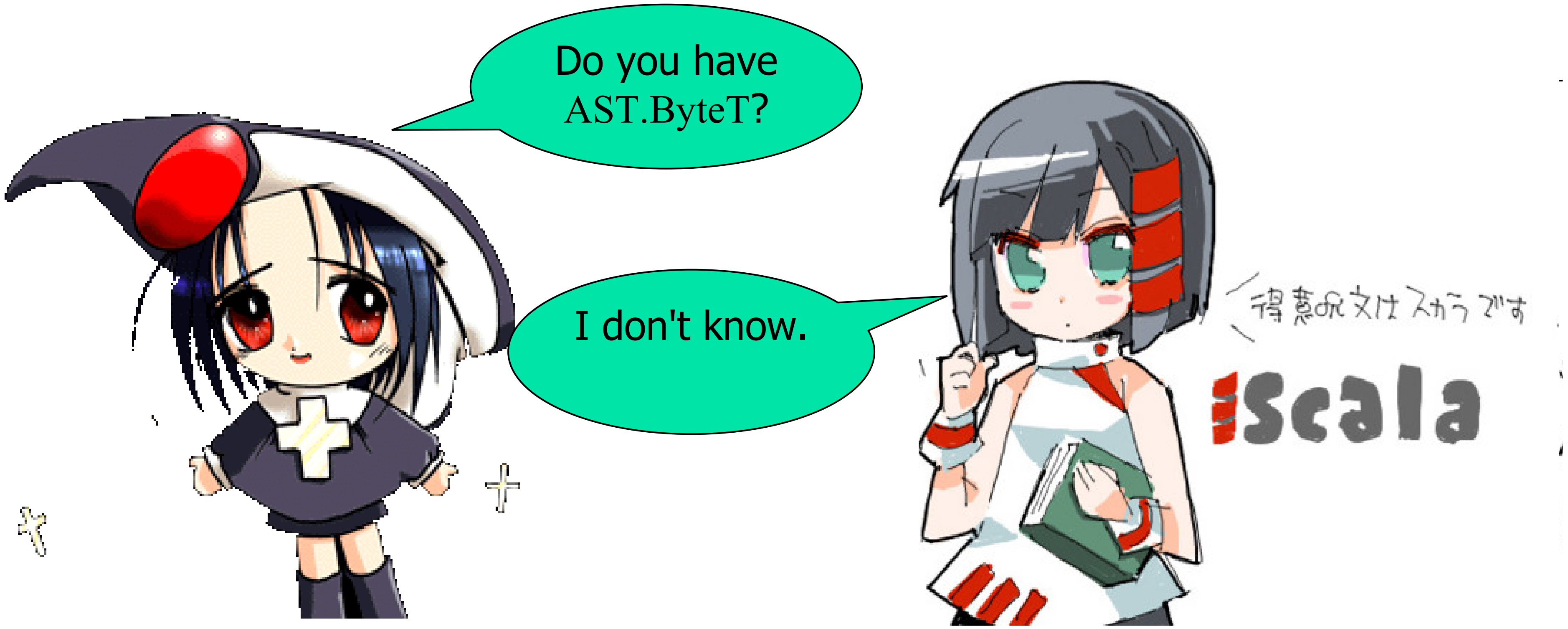
→ I decided to port my java code to scala code

How to port it

1. Create Scala classes for AST nodes
2. Rewrite the grammar(Parsing code) to use the Scala classes
3. Rewrite Info, Typing to use Scala classes
4. Port Java utility classes to Scala classes
   - Used from Info, Typing
5. Port Info, Typing phases
6. Port Generation phase (not completed yet)

Problem(1)
- object inside object from Java

- Assume object AST { case object ByteT }



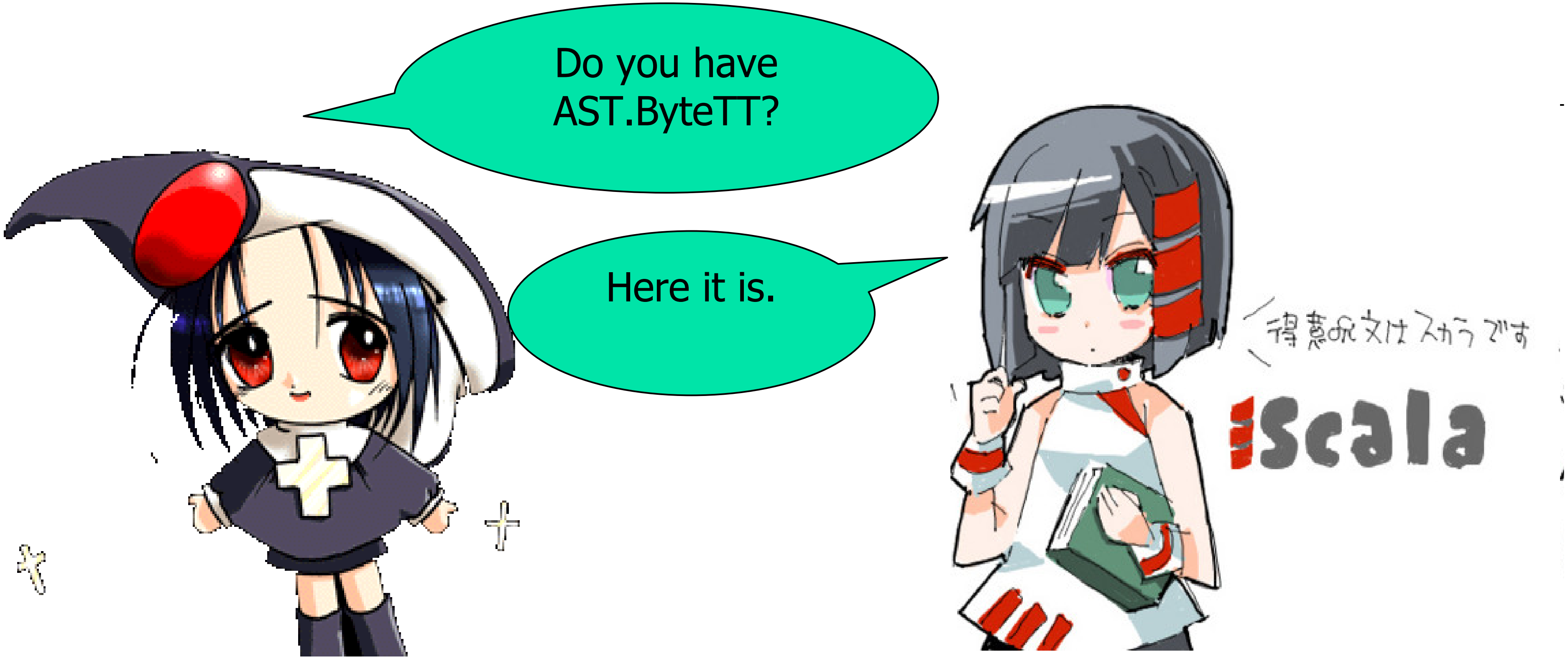
Solution(ugly)

- Assume object AST { case object ByteT }

Do you have AST$Byte$Module$?
Here it is.
得意呪文はスカラです
Scala

Workaround

- define acccessor val
  - object AST { case object ByteT_ val ByteTT = ByteT_ }



Problem(2)
operator called from Java

- Assume a List objects named x
- Java-co want to call :: method

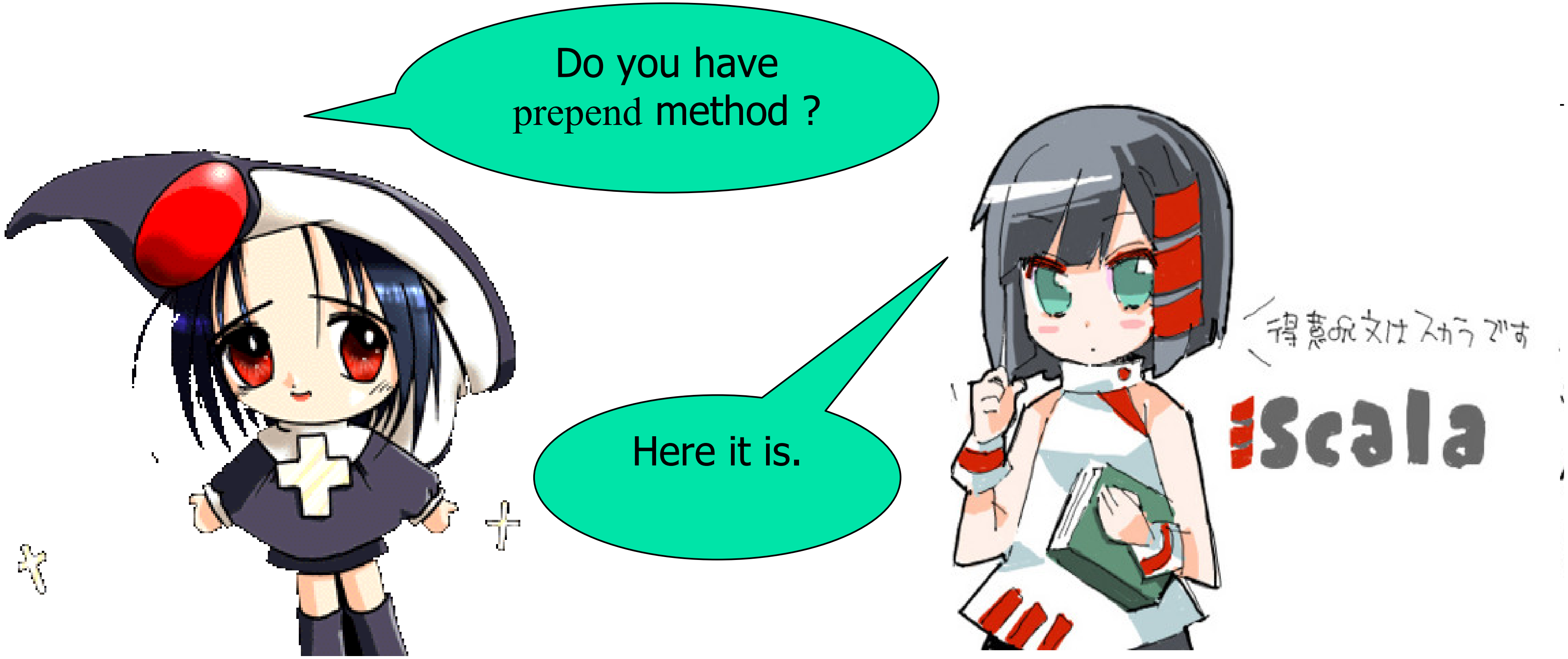
Solution(ugly)

Do you have
$colon$colon
method ?

Here it is.

Workaround

- Define alphabetical method in Scala
  - def prepend[T](x: T, xs: List[T]) = x::xs



Problem(3)

- IDEs sometimes see correct Scala code as error
  - Some Java <-> Scala cross reference
    - no workaround
  - Some nested Scala class/object from Java
  - Some Scala generics from Java
    - workaround: inserting cast to suppress error messsages

IDE evaluation

- Eclipse:
  - 2.0.0.beta4: Onion compiler compiles successfully
  - While the porting, There were many error markers
    - 12/2010
- IntelliJ IDEA:
  - Now: Onion compiler compiles successfully
  - While the porting, There were some error markers
  - Intelli's inter-language refactoring was useful
    - 「Renaming」 refactoring
  - 「Java to Scala」 functionality was useful
- NetBeans: I don't evaluate it

Result of porting

- Most Java code has been ported to Scala Code
  - Except Generation phase
- LoC was decreased about 20%
  - Because of straightforward porting
  - More "Scala-ish" code will decrease code size
- I've got more readable Onion compiler's code.
  - Easy to add some features

Advice on migration to Scala

- Code generation tool (e.g. parser generator) may make it difficult to port
- IntelliJ's 「Java to Scala」 is our friend
- Nested Java classes may be replaced to Scala's nested function definitions

Conclusions

- Introduction to Onion
- Structure of Onion
- How to port it
- Problems on the porting
- Advice on migration from Java to Scala

- Any Questions?