Java Generics

Parametric Polymorphism

UNBOUNDED WILDCARD GENERICS

DR. ERIC CHOU

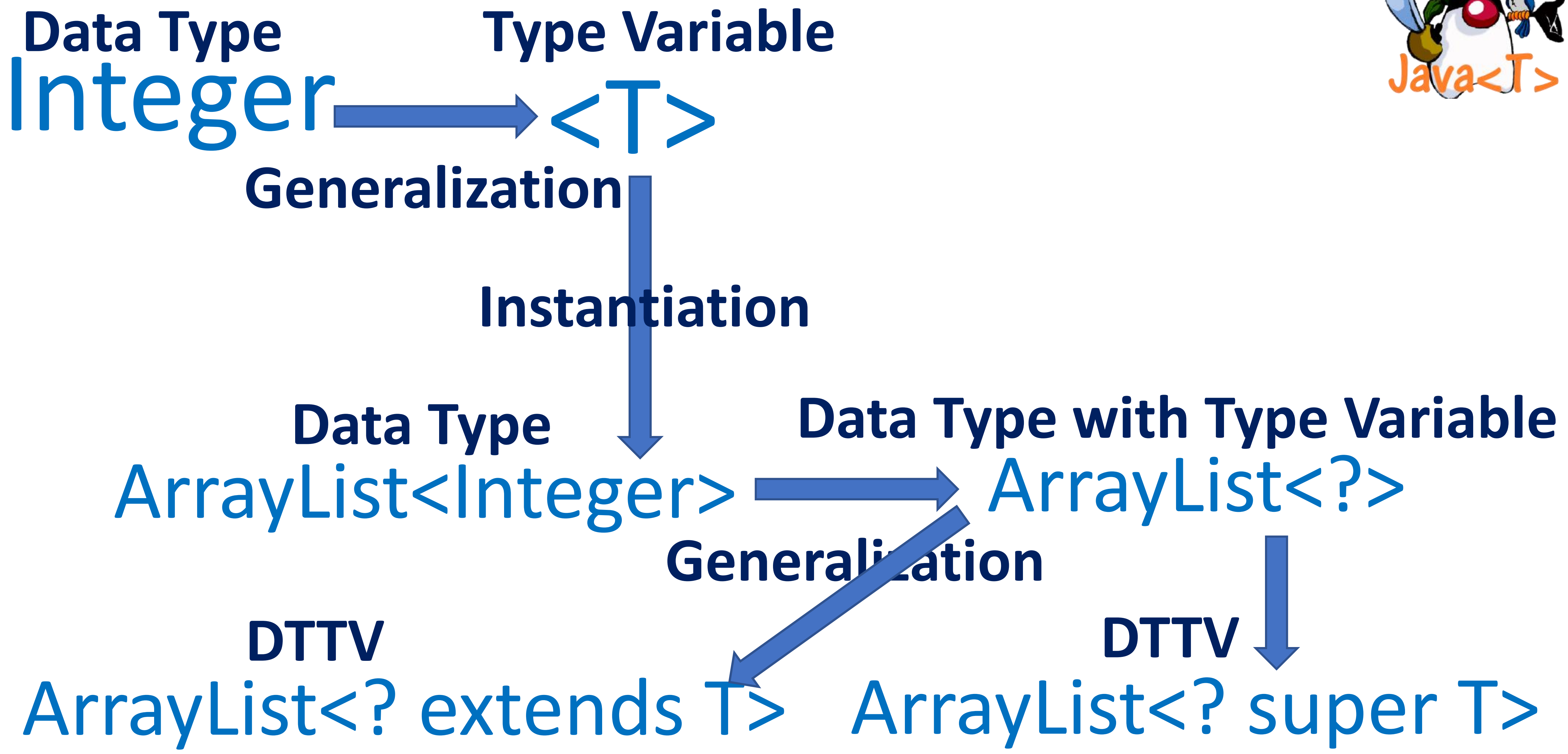
IEEE SENIOR MEMBER

Unbounded Wildcard

The first form, **?**, called an unbounded wildcard, is the same as **? extends Object**. The second form, **? extends T**, called a bounded wildcard, represents **T or a subtype of T**. The third form, **? super T**, called a lower-bound wildcard, denotes **T or a supertype of T**. You can fix the error by replacing line 12 in **WildCardNeedDemo.java** as follows:

```
public static double max(GenericStack<? extends Number> stack) {
```

<? extends Number> is a wildcard type that represents Number or a subtype of Number, so it is legal to invoke max(new GenericStack<Integer>()) or max(new GenericStack<Double>()).


Data Type
Integer
Type Variable
<T>
Generalization
Instantiation
Data Type
ArrayList<Integer>
Data Type with Type Variable
ArrayList<?>
Generalization
DTTV
ArrayList<? extends T>
DTTV
ArrayList<? super T>

Unbounded Wildcard

- **AnyWildCardDemo.java** shows an example of using the **?** wildcard in the print method that prints objects in a stack and empties the stack. **<?>** is a wildcard that represents any object type.
- It is equivalent to **<? extends Object>.** What happens if you replace **GenericStack<?>** with **GenericStack<Object>**? It would be wrong to **invoke print(intStack)**, because **intStack** is not an instance of **GenericStack<Object>**.
- Please note that **GenericStack<Integer>** is not a subtype of **GenericStack<Object>**, even though Integer is a subtype of Object.

Unbounded Wildcard

Demo Program: AnyWildCardDemo.java

Go BlueJ!