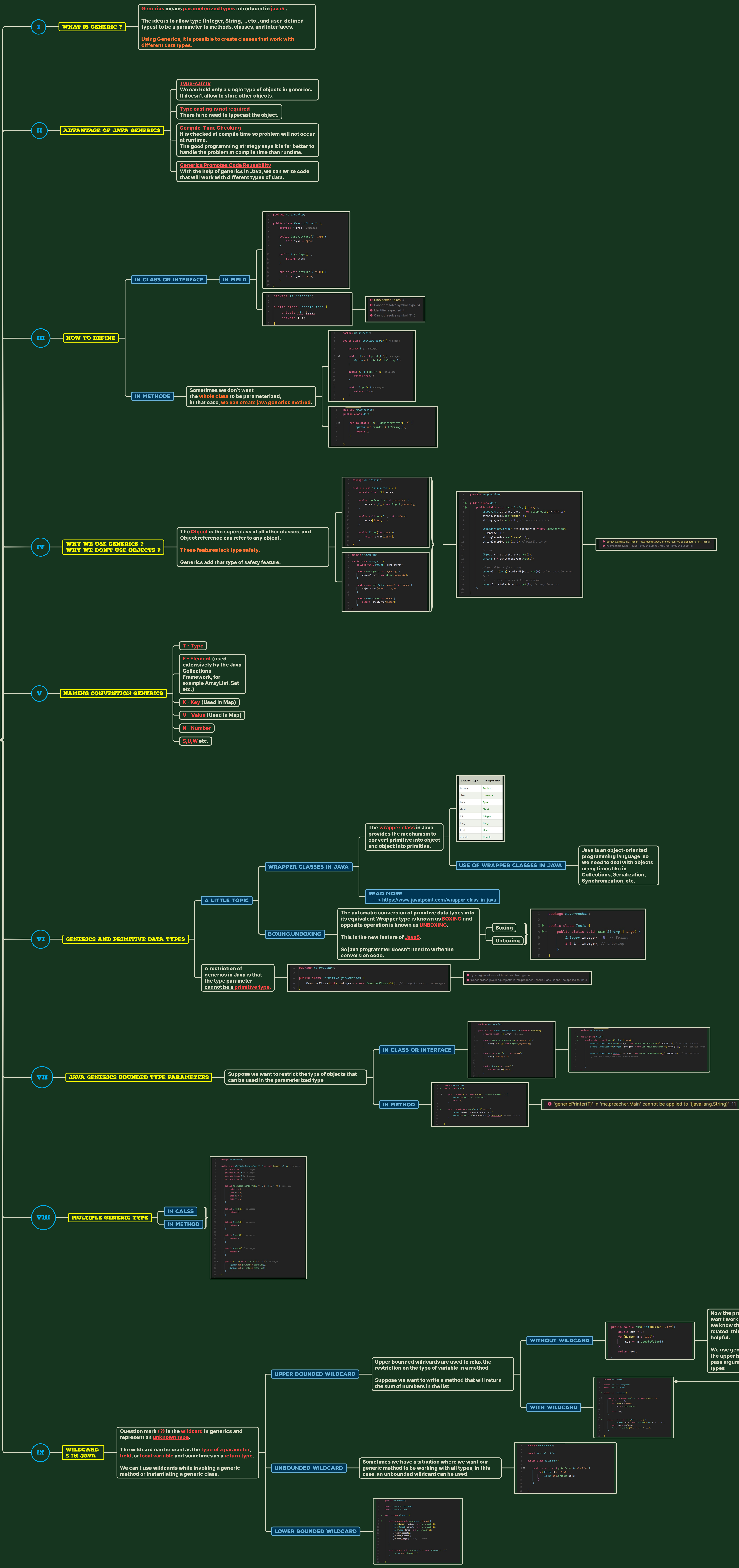


GENERICS<-> IN JAVA

Prepared by github.com/ali-mp



IXWILDCARD S IN JAVA

UPPER BOUNDED WILDCARD

UNBOUNDED WILDCARD

LOWER BOUNDED WILDCARD

Question mark (?) is the **wildcard** in generics and represent an **unknown type**.  
  
The wildcard can be used as the **type of a parameter, field, or local variable** and sometimes as a **return type**.  
  
We can't use wildcards while invoking a generic method or instantiating a generic class.

Upper bounded wildcards are used to relax the restriction on the type of variable in a method.  
  
Suppose we want to write a method that will return the sum of numbers in the list

WITHOUT WILDCARD

```
public double sum(List<Numbers> list) {  
    double sum = 0;  
    for(Numbers n : list) {  
        sum += n.doubleValue();  
    }  
    return sum;  
}
```

Now the problem with above implementation is that it won't work with List of Integers or Doubles because we know that List<Integer> and List<Double> are not related, this is when an upper bounded wildcard is helpful.

We use generics wildcard with extends keyword and the upper bound class or interface that will allow us to pass argument of upper bound or it's subclasses types

WITH WILDCARD

```
package me.preacher;  
  
import java.util.List;  
  
@ public class WildCard {  
  
    @ public static void printAndCalculateSum(List<? extends Number> list) {  
        for(Numbers n : list) {  
            System.out.println(n);  
            sum += n.doubleValue();  
        }  
        return sum;  
    }  
  
    @ public static void calculateSum(List<? extends Number> list) {  
        for(Numbers n : list) {  
            sum += n.doubleValue();  
        }  
        return sum;  
    }  
}
```

Sometimes we have a situation where we want our generic method to be working with all types, in this case, an unbounded wildcard can be used.

```
package me.preacher;  
  
import java.util.List;  
  
@ public class UnBound {  
  
    @ public static void printAndCalculateSum(List<?> list) {  
        for(Numbers n : list) {  
            System.out.println(n);  
            sum += n.doubleValue();  
        }  
        return sum;  
    }  
}
```

LOWER BOUNDED WILDCARD

```
package me.preacher;  
  
import java.util.List;  
  
@ public class LowerBound {  
  
    @ public static void printAndCalculateSum(List<? super Number> list) {  
        for(Numbers n : list) {  
            System.out.println(n);  
            sum += n.doubleValue();  
        }  
        return sum;  
    }  
}
```