

Multiple upper bounds

We briefly introduce the idea of multiple lower bounds. This is meant only for completeness, in case you run across something like this. It doesn't happen too often.

A class can extend one other class and implement several interfaces, as shown to the right. Therefore, it makes sense to allow a generic type to have several upper bounds, as shown by the following two examples.

```
public class D<T> extends C & I2 & I1 { ... }  
public <T> extends C & I1 & I2 void m5(T p) { ... }
```

Character '&' is used to separate the bounds, not a comma, because

```
<T> extends C, I1, I2>
```

denotes three different type parameters (T, I1, and I2) and not one type parameter with three upper bounds.

The most frequent use of multiple upper bounds is to implement two or more interfaces, as in this example:

```
public static <T> extends Comparable<T> & Iterator<T>> void m(T p) {...}
```

Here are two restrictions on multiple lower bounds.

1. At most one class may be included in the upper bounds and if it is included, it must be the first one.
2. A wildcard may not have more than one upper bound, e.g. <?> extends I1, I2> is syntactically incorrect.

```
public interface I1 { ... }  
public interface I2 { ... }  
public class C { ... }  
public class S extends C  
    implements I1, I2 { ... }
```