**Contravariance**

class GarbageCan[-A] {

// .. don't worry about implementation yet

}

def setGarbageCanForPlastic(gc: GarbageCan[PlasticItem]): Unit = {

// sets garbage can for PlasticItem items

}

The method setGarbageCanForPlastic accepts a GarbageCan of type **PlasticItem or supertype of PlasticItem**. **It accept only Item**. This is possible because type parameter A is prefixed with a -. It indicates that subtyping is contravariant in that parameter. It can be said that class GarbageCan is contravariant in its type parameter. **[-A] it means that takes A and all super classes of A.**

// contravariant subtyping

setGarbageCanForPlastic(new GarbageCan[Item])

// invariant

setGarbageCanForPlastic(new GarbageCan[PlasticItem])

// Compile error ! covariant subtyping

setGarbageCanForPlastic(new GarbageCan[PlasticBottle])

**Contravariant subtyping**

A <: B

GarbageCan[B] <: GarbageCan[A]

If A is a subtype of B then GarbageCan[B] should be a subtype of GarbageCan[A]. This property is called contravariant subtyping.