

## Re: Generics error messages

---

*Source:* <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2005-08/msg00234.html>

---

- *From:* Chris Smith <[cdsmith@xxxxxxx](mailto:cdsmith@xxxxxxx)>
  - *Date:* Sun, 31 Jul 2005 00:23:04 -0600
- 

Roedy Green <[look-on@xxxxxxxxxxxxxxxxxxxxxx](mailto:look-on@xxxxxxxxxxxxxxxxxxxxxx)> wrote:

> Is there distinction between <E> and <? extends E>?  
>

Yes, but it's kind of subtle.

Let's say I've got a class `Automobile` and two classes `Car` and `Truck` that both derive from `Automobile`. A `Collection<Automobile>` can contain instances of `Automobile`, or `Car`, or `Truck`. I can, for example, add one of each and then read them back out into references of type `Automobile`, and this is all safe to do. A `Collection<Car>`, on the other hand, may only contain `Cars`. It is an error to try to add a `Truck` to such a collection. Note that the two generic classes `Collection<Automobile>` and `Collection<Car>` are NOT related by inheritance, even though their type parameters are related.

Note that up to this point, we've only been talking about the (pre-erasure) runtime class of various collection objects. The references used to refer to those collection objects can have a range of other types. A reference of type `Collection<Automobile>` is somewhat limited. It can ONLY refer to a `Collection<Automobile>`, and can NOT refer to a `Collection<Object>` or a `Collection<Car>`. (Of course, an instance of `Collection<Automobile>` can contain objects of class `Car`, but that's not the same as having an object of class `Collection<Car>`.)

There are also wildcard types. A `Collection<? extends Automobile>` is a reference type that can refer to any of a `Collection<Automobile>`, `Collection<Car>`, or `Collection<Truck>`. Even though the three classes are not related to each other by inheritance, they are all assignment compatible with that wildcard type.

So what's the practical difference? Well, a reference of type `Collection<? extends Automobile>` MIGHT be referring to an object of class `Collection<Truck>`, so you can't safely add a `Car` to it. It also might be referring to an object of `Collection<Car>`, so you can't add a `Truck` to it. In fact, it turns out you can never add anything to a collection of a lower-bound wildcard type, because you can never guarantee that the object you're adding is compatible with the object's actual type parameter. This is quite different from the simpler

## Re: Generics error messages

Collection<Automobile>, where you know that the type parameter is Automobile and therefore it's definitely safe to add a Car or Truck object.

(Note that this can sometimes be counter-intuitive. Many people initially expect that you would be able to add a Car object to a Collection<? extends Automobile> but would not be able to add it to a Collection<Automobile>. That's an unfortunate confusion of words, and it's almost exactly backwards from the truth.)

> If the underlying arrayList has a method addAll( Collection ) types  
> aside, why would not my method override it.?

Because someone might pass a Collection<Car> to addAll, and then your method would not be appropriate. Essentially, you're writing a more specific method than the superclass; one that is pickier about its parameters. You can't do that without an overload, and you can't overload without changing the type erasure.

--

www.designacourse.com

The Easiest Way To Train Anyone... Anywhere.

Chris Smith – Lead Software Developer/Technical Trainer  
MindIQ Corporation

---

### • *Follow-Ups:*

- ◆ **Re: Generics error messages**  
    ◇ From: Roedy Green
- ◆ **Re: Generics error messages**  
    ◇ From: Roedy Green
- ◆ **Re: Generics error messages**  
    ◇ From: Roedy Green
- ◆ **Re: Generics error messages**  
    ◇ From: jan V

### • *References:*

- ◆ **Generics error messages**  
    ◇ From: Roedy Green
- Prev by Date: **Re: How easy is it to reverse engineer a java program?**
- Next by Date: **Re: new FileWriter hangs**
- Previous by thread: **Generics error messages**
- Next by thread: **Re: Generics error messages**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**