What's left to know about Generics?

In the next few videos, I want to cover the following topics.

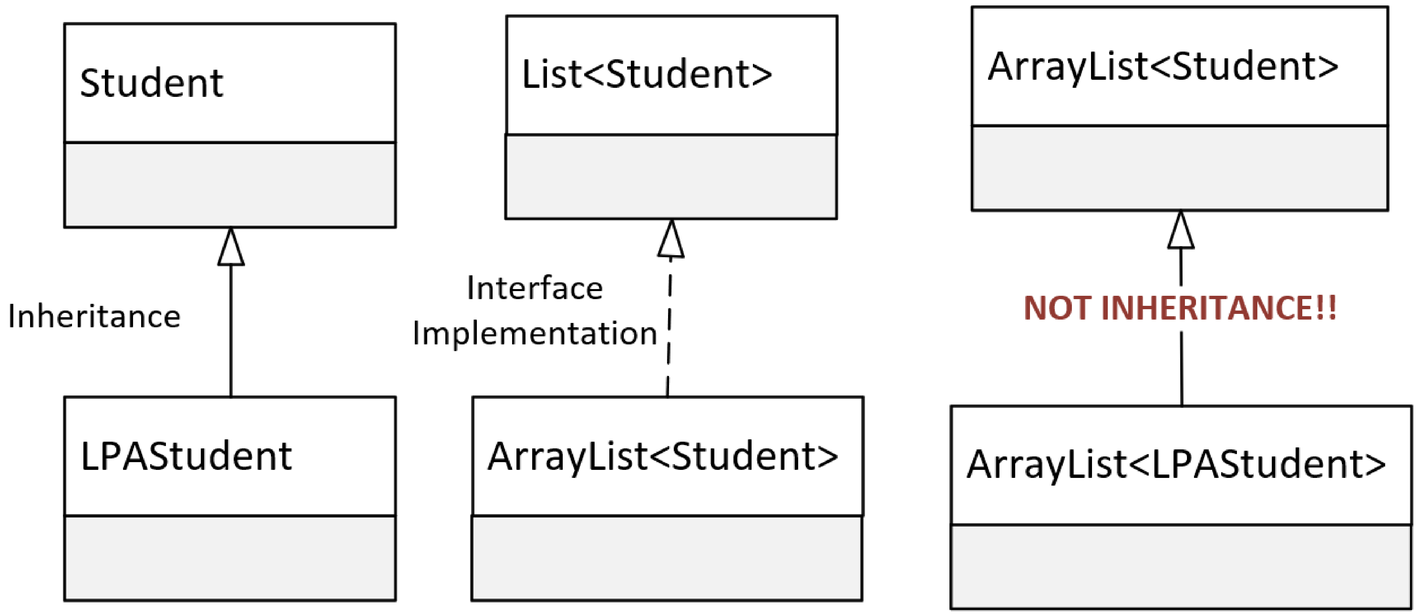
- Using generic references that use type arguments, declared in method parameters and local variables.
- Creating generic methods, apart from generic classes.
- Using wildcards in the type argument.
- Understanding static methods with generic types.
- Using multiple upper bounds.

This isn't inheritance

We know LPAStudent inherits from Student, and we can pass an instance of LPA Student to any method, or assign it to any reference type, declared with the type Student.

We also know that ArrayList implements List, and we can pass an ArrayList to a method or assign it to a reference of the List type.

Why can't we pass an ArrayList of LPA Student, to the method parameter that's declared as a List of Student?



This isn't inheritance

Surely, if an LPAStudent is a Student, a List of LPAStudent is ultimately a List of Student.

It's very natural to assume that a method that takes a List with Students should accept a List with LPAStudents, because LPAStudent is a Student after all.

But that's not how it works.

When used as reference types, a container of one type has no relationship to the same container of another type, even if the contained types do have a relationship.