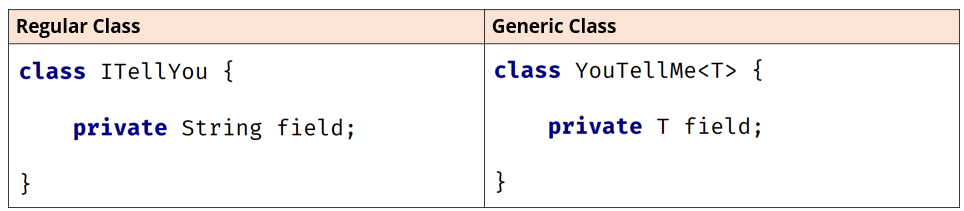
Generics

## Introduced from JDK 5.0 onwards

* The thing to notice with the generic class, is that the class declaration has angle brackets with a T in them, directly after the class name. Where ‘T’ is the placeholder for a type that will be specified later.
* This is called a type identifier, and it can be any letter or word, but T which is short for Type is commonly used.



For the generic class, the field's type is that placeholder, just T, and this means it can be any type at all.

The T in the angle brackets means it's the same type as the T, specified as the type of the field.

On this slide, I have a variable declaration of the generic ArrayList.

In the declaration of a reference type that uses generics, the type parameter is declared in angle brackets.

The reference type is ArrayList, the type parameter (or parameterized type) is String, which is declared in angle brackets, and listOfString is the variable name.

Text

Description automatically generated

Note: when working with generics, you cannot directly use primitive types like int, char, boolean, etc., as type arguments. Instead, you need to use their corresponding wrapper classes: Integer, Character, Boolean, etc. This restriction exists because generics in Java are implemented using **type erasure**.

To ensure compatibility with existing code and maintain backward compatibility, Java introduced **autoboxing** and **unboxing**, which automatically converts between primitive types and their corresponding wrapper classes