

Wrapper Classes

Chapter 2: Wrapper Classes · Topic 1

Key Concept

Each primitive type has a corresponding wrapper class, like Integer for int, Double for double, and Character for char.

Autoboxing automatically converts a primitive value to its wrapper object, while unboxing converts a wrapper object back to a primitive.

Wrapper classes provide useful methods, like Integer.parseInt() to convert a String to an int, useful for processing user input.

Understanding wrapper classes is important for working with Java collections and other features that require objects rather than primitives.

Quick Revision Notes

- Wrapper classes convert primitives to objects.
- Integer wraps int; Double wraps double; Character wraps char.
- Autoboxing converts primitive to wrapper automatically.
- Unboxing converts wrapper back to primitive.
- Integer.parseInt() converts String to int.
- This is useful for processing user input.
- Wrapper classes are needed for Java collections.
- They bridge primitives and object-based features.

Practice Questions

- 1 Convert an int to an Integer object.
- 2 Use Integer.parseInt() to convert a String to an int.
- 3 Discuss autoboxing and unboxing.
- 4 Research why wrapper classes are needed for collections.
- 5 Create an example using a wrapper class method.
- 6 Explain wrapper classes in your own words.