Task : Wrapper Classes, Autoboxing & Unboxing

Task 1: Convert Primitive to Wrapper and Vice Versa

Objective: Practice `valueOf()` and `xxxValue()` methods.

Task:

- Take an `int`, `double`, and `char` variable.
- Convert each to their respective wrapper classes using `valueOf()`.
- Convert back to primitive using `xxxValue()` methods.
- Print all values.

Task 2: Autoboxing and Unboxing

Objective: Understand implicit boxing/unboxing.

Task:

- Declare a `List<Double>` and add primitive `double` values.
- Retrieve values using a loop and perform sum of all elements.
- Display the average using unboxed values.

Task 3: Parsing from Strings

Objective: Use `parseXxx()` methods.

Task:

- Take input strings like `"123"`, `"45.67"`, `"true"`.
- Convert them into `int`, `double`, and `boolean` using `Integer.parseInt()`, `Double.parseDouble()`, and `Boolean.parseBoolean()`.
- Print the converted primitives and their types.

Task 4: Comparing Wrapper Objects

Objective: Understand how wrapper objects behave with `==` and `.equals()`.

Task:

- Create two `Integer` objects with same value using:
  - Autoboxing
  - `new Integer(100)`
- Compare using `==` and `.equals()` and explain the difference.

Task 5: Wrapper Classes in Generics

Objective: Use wrapper classes in a generic class.

Task:

- Create a generic class `Box<T>`.
- Add a method to store a value and retrieve it.
- Instantiate it for `Integer` and `Double`.
- Autobox values into the generic class and retrieve them.

Task 6: Wrapper with ArrayList (Grades Tracker)

Objective: Work with `ArrayList<Integer>`.

Task:

- Create an `ArrayList<Integer>`.
- Add 5 student marks using autoboxing.
- Remove the lowest mark using Collections method.
- Print max, min, and average.

Task 7: Temperature Converter App

Objective: Use wrapper methods and collections.

Task:

- Input a list of temperature strings like `["36", "38", "40"]`.
- Convert to integers using `Integer.parseInt()`.
- Convert Celsius to Fahrenheit and store in `ArrayList<Double>`.
- Print both Celsius and Fahrenheit values.

Task 8: Bank Account Using Wrapper Class

Objective: Real-world use of wrapper classes.

Task:

- Create a class `BankAccount` with:
  - AccountNumber (String)
  - Balance (Double wrapper)
- Add methods: deposit(double amount), withdraw(double amount), and show balance.
- Demonstrate how wrapper handles null balance (set default to 0.0 if null).

Task 9: Null Handling in Wrapper Classes

Objective: Show wrapper class behavior with nulls.

Task:

- Declare a `Double` object and set it to `null`.
- Try unboxing it to `double` → Catch the `NullPointerException`.
- Use `Optional` or default value strategy to fix it.

Task 10: Wrapper Classes with Sorting

Objective: Use wrapper classes in sorting.

Task:

- Create an array of `Integer` objects.
- Sort using `Arrays.sort()`.
- Find the 2nd highest value using wrapper methods.

Student Scoreboard App

- Input student names and scores using a `Map<String, List<Integer>>`.
- For each student:
  - Add marks using `ArrayList<Integer>`.
  - Calculate average score using unboxing.
  - Find highest scorer using wrapper comparison.