A class variable is used to maintain data for each object of the class.

The keyword **PUBLIC** is an access modifier and every public class declaration must be stored in a file that as the same name with the class and ending with file extension .java. Also all class names,method names and variable names.When you declare a class, you can provide your own constructor to specify *custom initialization* for objects of your class. For example, you might want to specify a name for an Account object when the object is created. Method that does not return anything is a void method.

A primitive-type variable can hold exactly *one* value of its declared type at a time. For

example, an int variable can store one integer at a time. When another value is assigned

to that variable, the new value replaces the previous one—which is *lost*.

Instance variable exist before the method and they called on an object.Variables or method with access modifier private exist and accessible to method in the class where they are declared.

Variables declared inside a method are called local variables also parameters of a method are also local variables inside that method.Also if a method have the same name as the instance variable, the method body will refer to local variable. Method with empty parenthesis does not need additional information to perform task.

@BeforeEach

The @BeforeEach annotation in JUnit 5 is used to mark a method that should be executed before each test method in a test class. This is particularly useful for setting up common test fixtures or initializing test data required by the test methods.

@AfterEach

The **@AfterEach** annotation in JUnit 5 is used to indicate that a specific method should be executed ****after each test method**** in the test class. It is commonly used for cleanup tasks, such as releasing resources, resetting states, or closing connections, to ensure that each test runs in isolation.