In Java, the combination of "static" and "void" is often used in method declarations. Let's understand the meaning of each term:

1. Static: When a method or variable is declared as "static", it means that it belongs to the class itself rather than an instance of the class. This means that the method or variable can be accessed and used without creating an object of the class. Static members are shared among all instances of the class and can be accessed directly using the class name.

2. Void: "Void" is a keyword used as the return type of a method to indicate that the method does not return any value. In other words, a method with a "void" return type performs an action or task without producing a result.

When "static" and "void" are used together in a method declaration, it means that the method is a static method that does not return any value. Here are some characteristics of a static void method:

- Accessibility: Static methods can be accessed directly using the class name without creating an object of the class.
- No Return Value: A static void method does not return any value. It performs a specific task or action but does not produce a result.
- Class-level Scope: Static methods are associated with the class itself rather than individual instances of the class. They can access other static members of the class but cannot directly access instance variables or methods.
- Usage: Static void methods are commonly used for utility methods, helper methods, or actions that don't require a return value. They can be called directly using the class name followed by the method name.

Here's an example of a static void method:

```java
public class MyClass {
    public static void printMessage() {
        System.out.println("Hello, World!");
    }
}
```

In the example above, the `printMessage()` method is a static void method. It can be called using the class name, like `MyClass.printMessage()`, without creating an object of the `MyClass`. It simply prints the message "Hello, World!" to the console.