Great question! protected and public might seem similar because they both allow access beyond the class where they're defined, but there are key differences in scope and intended use. Here's a breakdown:

Key Differences between public and protected

1. Visibility Across Packages

Public: Members marked as public can be accessed from anywhere in the program. Any class, in any package, can access a public member if it has visibility of the class itself.

Protected: protected members can be accessed only by subclasses (even if they are in a different package) and by classes in the same package. So, if another unrelated class (not a subclass and not in the same package) tries to access a protected member, it will not have access.

2. Intended Use with Inheritance

Public: The public modifier is intended for members that should be completely open for access, regardless of the relationship between classes.

Protected: The protected modifier is specifically designed for inheritance. It allows subclasses to access the parent's protected members, which lets subclasses extend and reuse the functionality of the superclass.

3. Access by Unrelated Classes in Other Packages

Public: Any unrelated class in any package can access public members.

Protected: Only subclasses can access protected members if they're in different packages. Unrelated classes in other packages cannot access protected members.

Code Example to Illustrate the Difference

Consider the following example with both public and protected members.

Step 1: Superclass with Public and Protected Members

```
// File: animals/Animal.java
package animals;

public class Animal {
    public String publicName = "Public Animal";
    protected String protectedName = "Protected Animal";
```

```
    public void showPublic() {
        System.out.println("Public method: " + publicName);
    }

    protected void showProtected() {
        System.out.println("Protected method: " + protectedName);
    }
}
```

In this Animal class:

publicName and showPublic() are public, so they're accessible everywhere.

protectedName and showProtected() are protected, so they're accessible in subclasses and in the same package.

Step 2: Subclass in a Different Package

```
// File: zoo/Lion.java
package zoo;

import animals.Animal;

public class Lion extends Animal {
    public void accessParent() {
        System.out.println(publicName);  // Accessible (public)
        System.out.println(protectedName);  // Accessible (protected via inheritance)

        showPublic();     // Accessible (public)
        showProtected();  // Accessible (protected via inheritance)
    }
}
```

In this Lion class:

Lion can access publicName and showPublic() because they're public.

Lion can also access protectedName and showProtected() because it's a subclass of Animal, even though Lion is in a different package.

Step 3: An Unrelated Class in Another Package

```
// File: TestAccess.java
import animals.Animal;

public class TestAccess {
    public static void main(String[] args) {
        Animal animal = new Animal();
```

```
        System.out.println(animal.publicName);  // Accessible (public)
        animal.showPublic();  // Accessible (public)

        // The following lines would cause an error because protectedName and showProtected()
are not accessible here:
        // System.out.println(animal.protectedName);  // Error
        // animal.showProtected();  // Error
    }
}
```

In TestAccess:

publicName and showPublic() are accessible because they are public.

protectedName and showProtected() are not accessible because TestAccess is not a subclass of Animal and is in a different package.

Summary Table

Key Takeaway

Public: Universal access, no restrictions.

Protected: Intended for inheritance, accessible in subclasses (even in different packages) and same-package classes only. Not accessible by unrelated classes in different packages.

protected is more restrictive and is meant to support inheritance by allowing subclasses to access certain members without exposing them fully like public.

Summary Table

Access Modifier	Access Level in Same Package	Access in Subclass (Different Package)	Access by Unrelated Class (Different Package)
`public`	Yes	Yes	Yes
`protected`	Yes	Yes	No