Getter and Setter method in Java

Best practice
- Implement a clone() method for the custom type
- Return a cloned object from the getter
- Assign a cloned object in the setter

When to use
- Achieve read-only or write-only access to the fields outside the class
- Validate the values of class fields
- Want to provide different access levels
- Perform additional logic
- Implement the JavaBeans specification for framework or tool compability
- Design a public API with a clear, consistent interface

Avoiding mistakes
- Using getters and setters with public fields
- Assigning objects references in the setter methods
- Returning object references directly from the getter methods
- Adding unncessary getters and setters