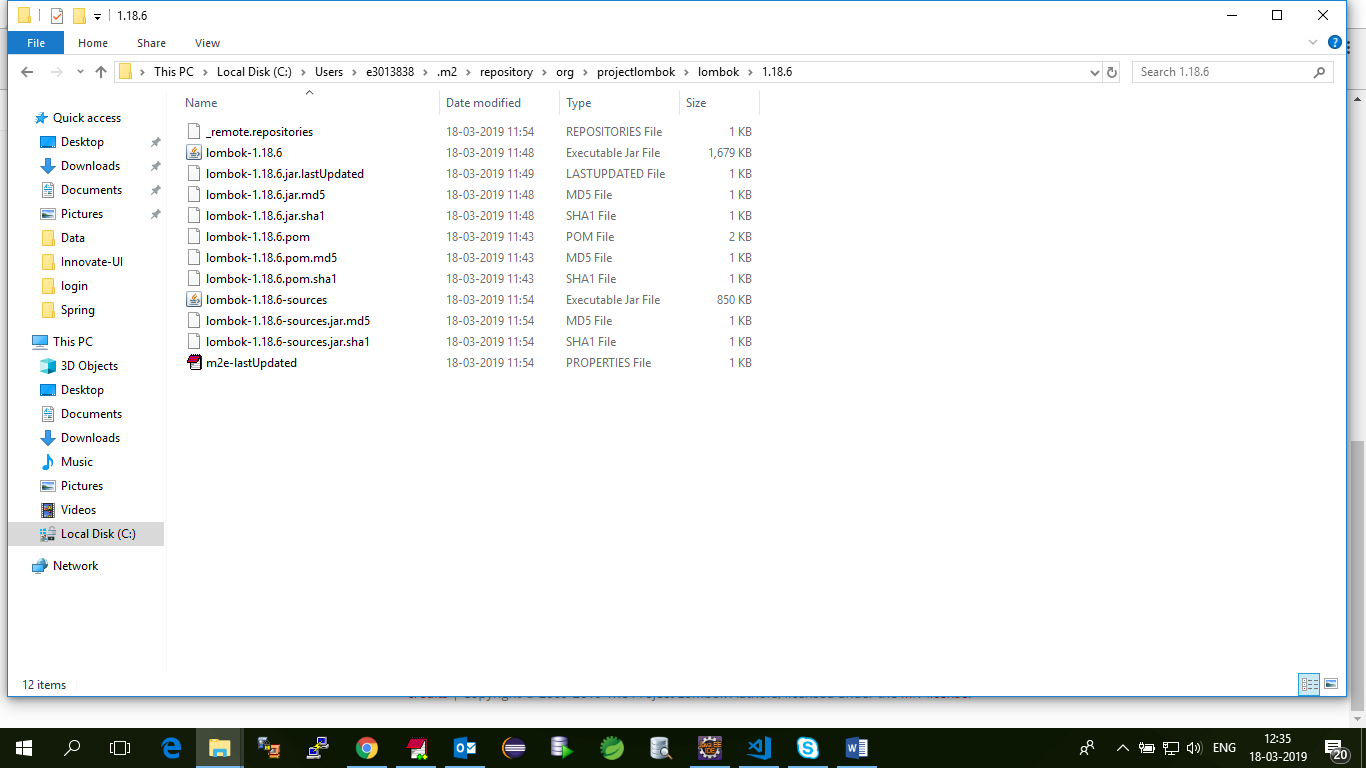
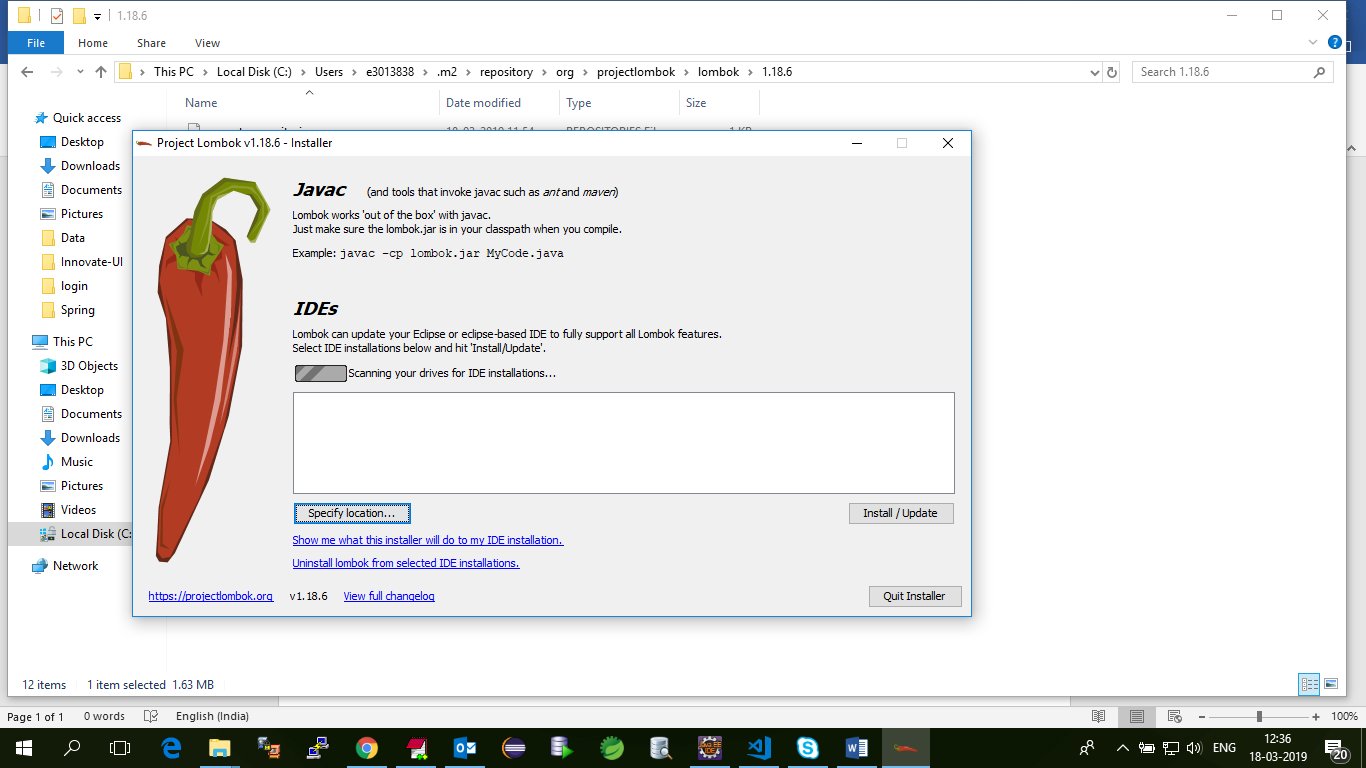
Set up Lombok in eclipse

Add this jar in maven [or download manually]





Give eclipse.exe location and click install/update.

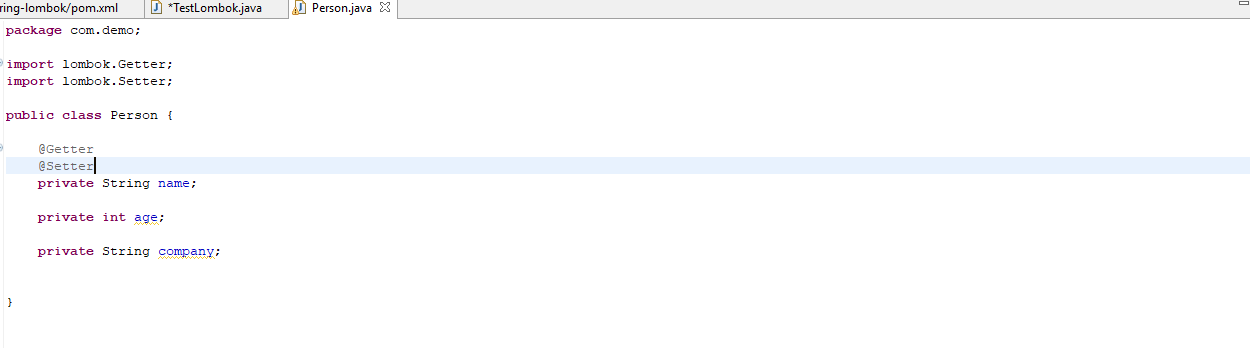
**LOMBOK IN JAVA**

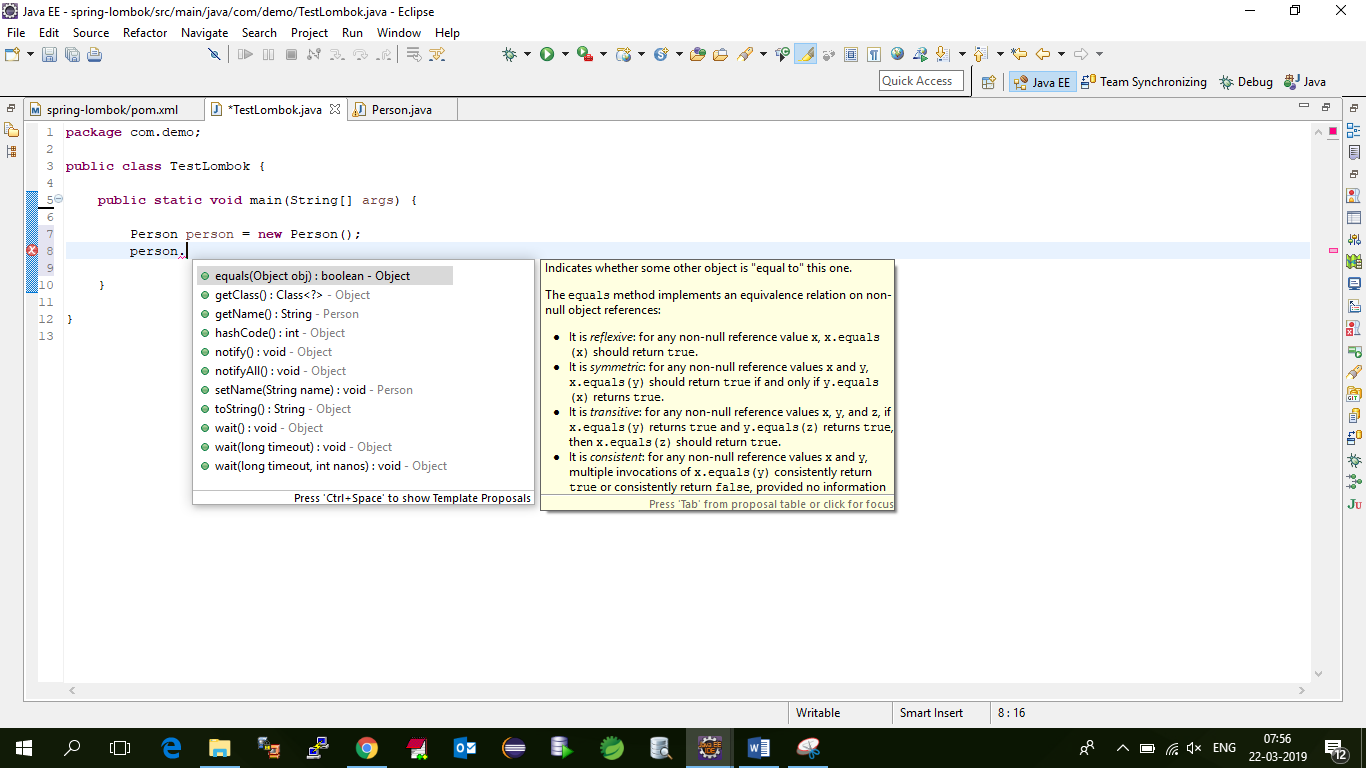
**https://projectlombok.org/features/all**

Project Lombok is a small library that can be used to reduce the amount of boilerplate Java code that is commonly written for Java classes. Project Lombok does this via annotations that can be added to the Java class for which common methods are desired. Most of the annotations are self-descriptive in their names: @Getter, @Setter, @EqualsAndHashCode, @ToString, and @NoArgsConstructor are examples

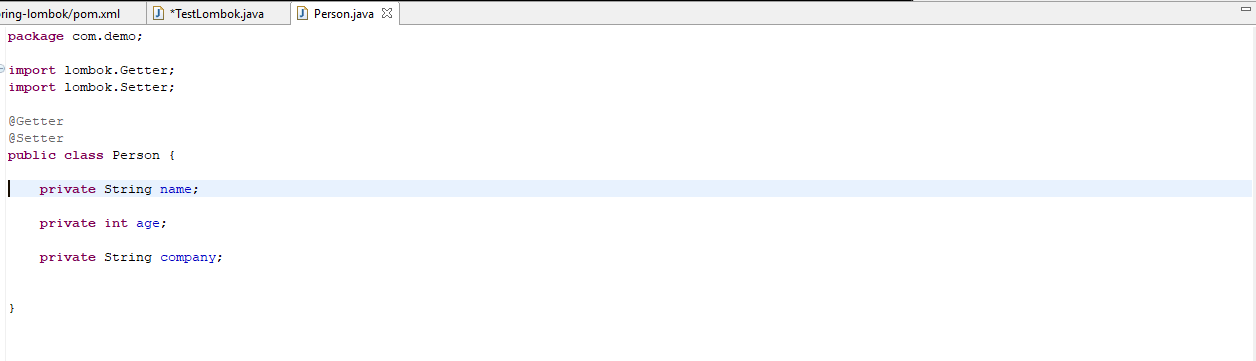
**@Getter and @Setter**

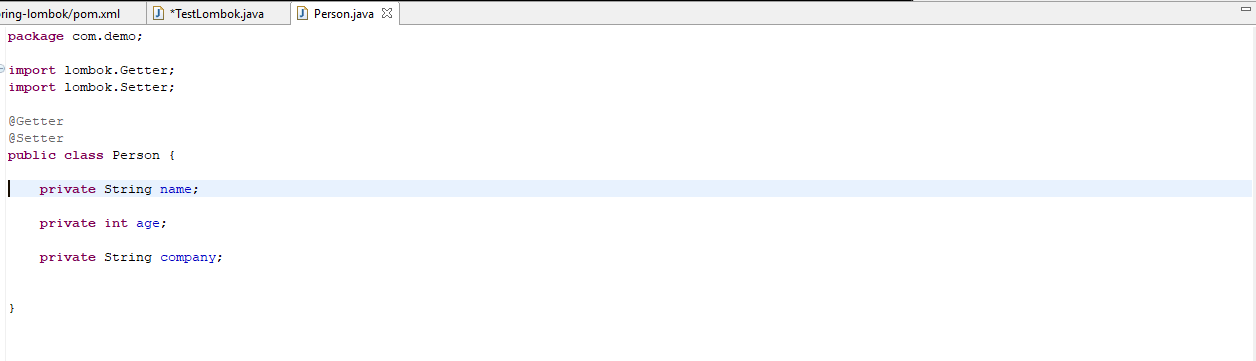
You can annotate any field with @Getter and/or @Setter, to let lombok generate the default getter/setter automatically.





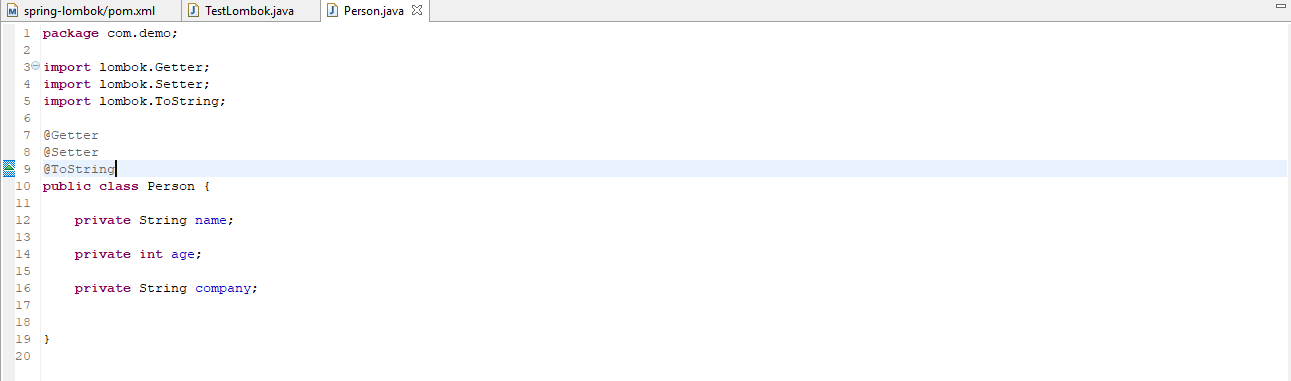
This annotation can also be applied to a class, in which case it'll be as if all non-static fields that don't already have

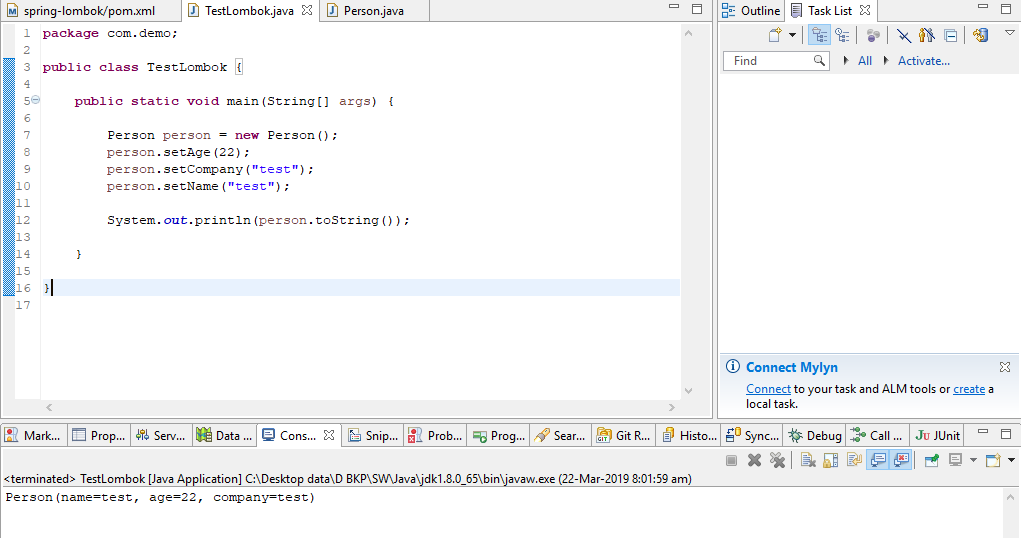




**@ToString**

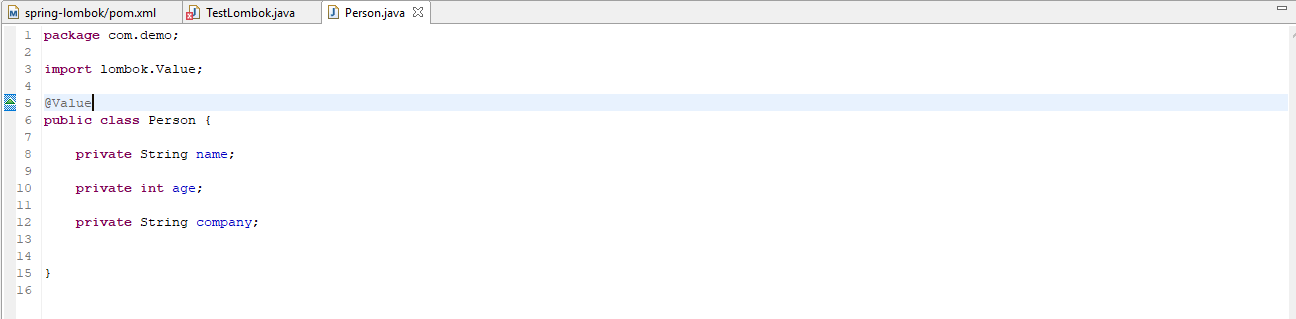
Generates an implementation for the {@code toString} method inherited by all objects, consisting of printing the values of relevant fields.

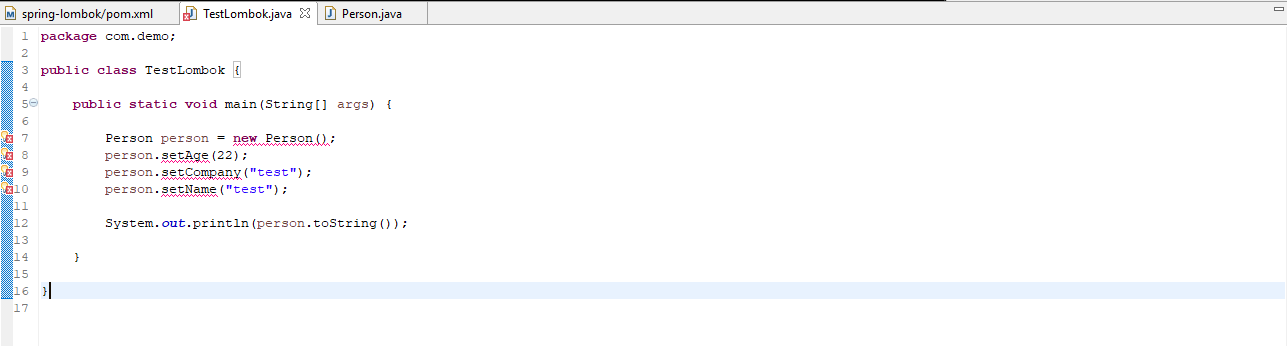


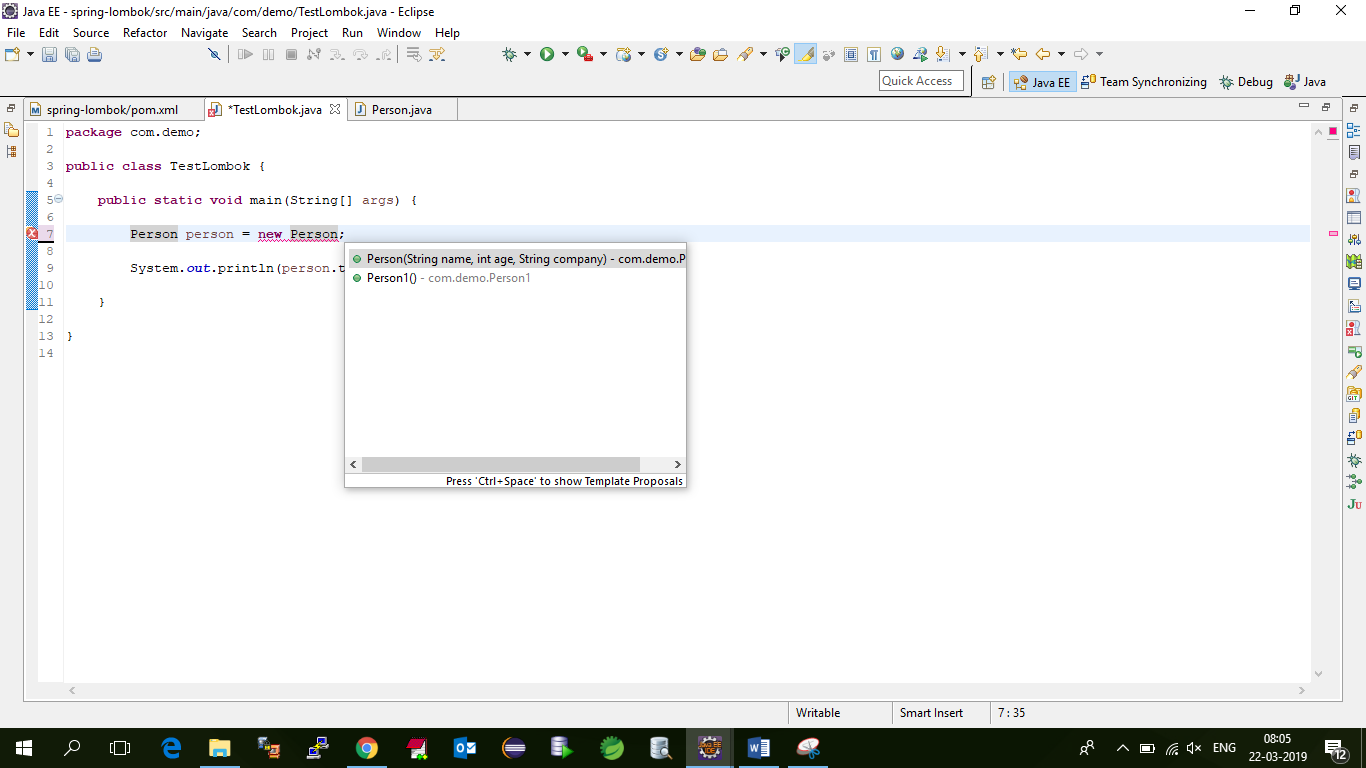


**@Value**

@Value is the immutable variant of @Data; all fields are made private and final by default, and setters are not generated. The class itself is also made final by default, because immutability is not something that can be forced onto a subclass. Like @Data, useful toString(), equals() and hashCode() methods are also generated, each field gets a getter method, and a constructor that covers every argument (except final fields that are initialized in the field declaration) is also generated.

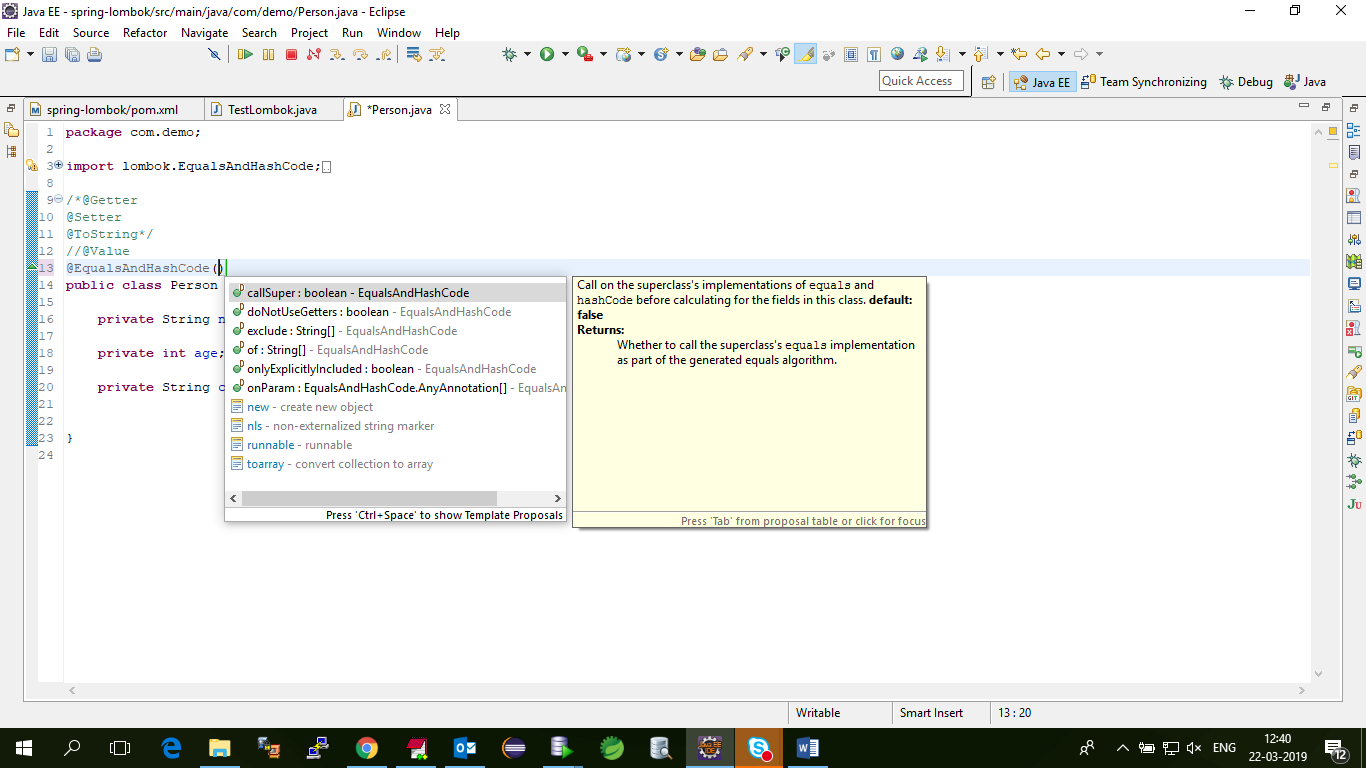






**@EqualsAndHashCode**

Any class definition may be annotated with @EqualsAndHashCode to let lombok generate implementations of the equals(Object other) and hashCode() methods. By default, it'll use all non-static, non-transient fields, but you can modify which fields are used (and even specify that the output of various methods is to be used) by marking type members with @EqualsAndHashCode.Include or @EqualsAndHashCode.Exclude. Alternatively, you can specify exactly which fields or methods you wish to be used by marking them with @EqualsAndHashCode.Include and using @EqualsAndHashCode(onlyExplicitlyIncluded = true).



**@NoArgsConstructor**

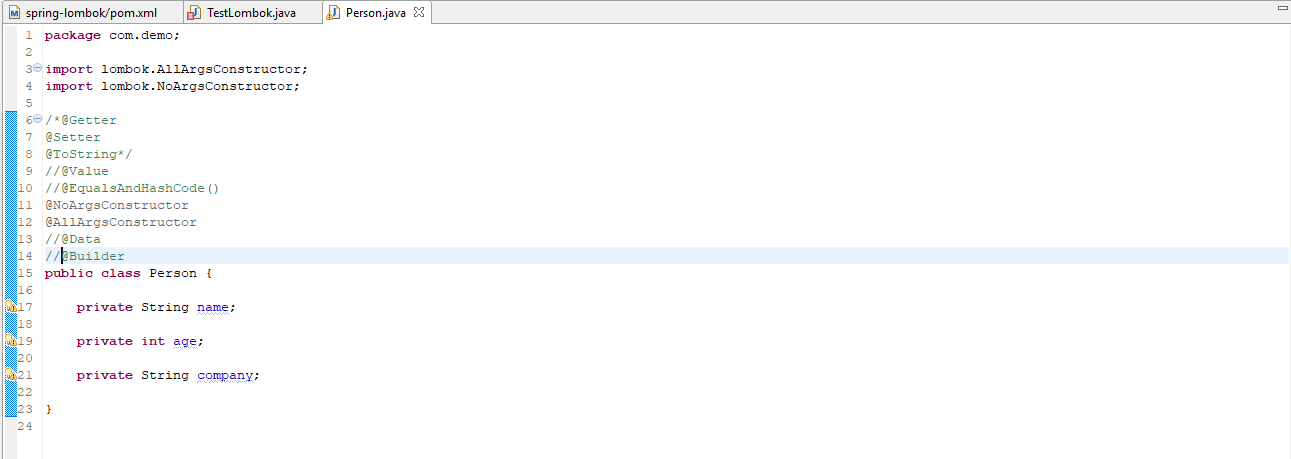
@NoArgsConstructor will generate a constructor with no parameters. If this is not possible (because of final fields), a compiler error will result instead, unless @NoArgsConstructor(force = true) is used, then all final fields are initialized with 0 / false / null.

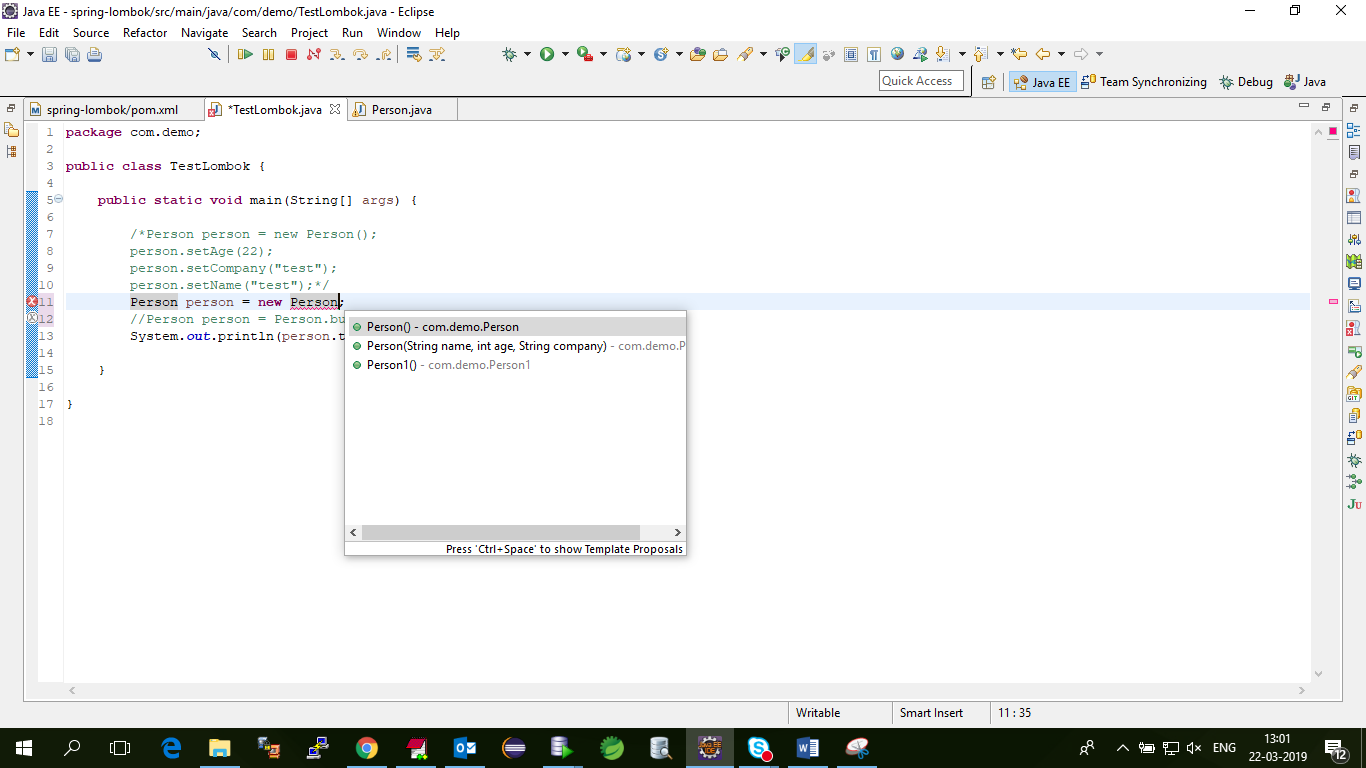
**@RequiredArgsConstructor**

Generates a constructor with required arguments. Required arguments are final fields

**@AllArgsConstructor**

@ AllArgsConstructor generates a constructor with 1 parameter for each field that requires special handling

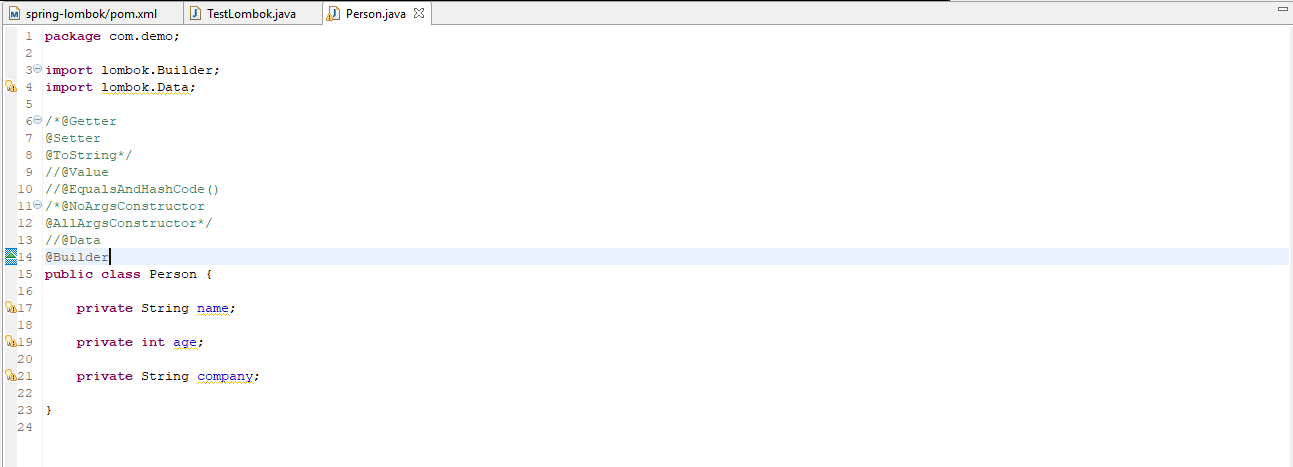


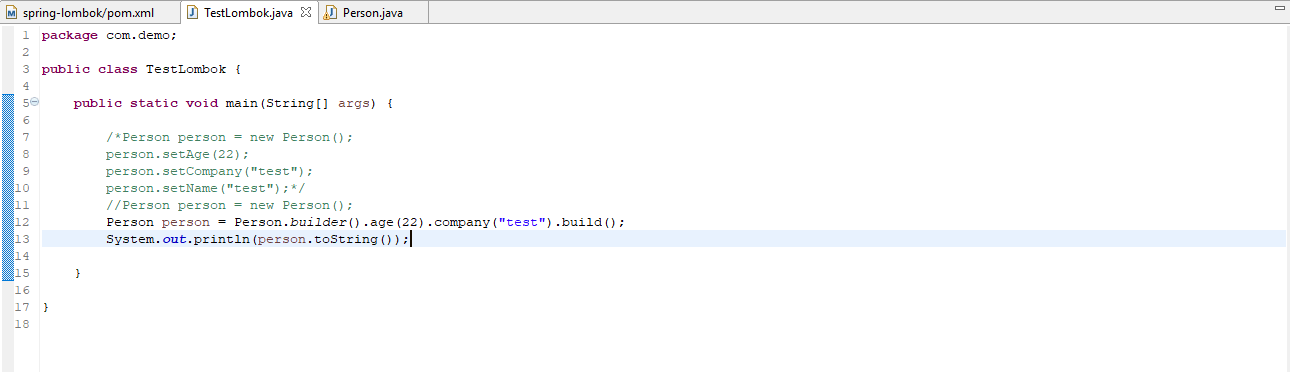


**@Data**

@Data is a convenient shortcut annotation that bundles the features of @ToString, @EqualsAndHashCode, @Getter / @Setter and @RequiredArgsConstructor together

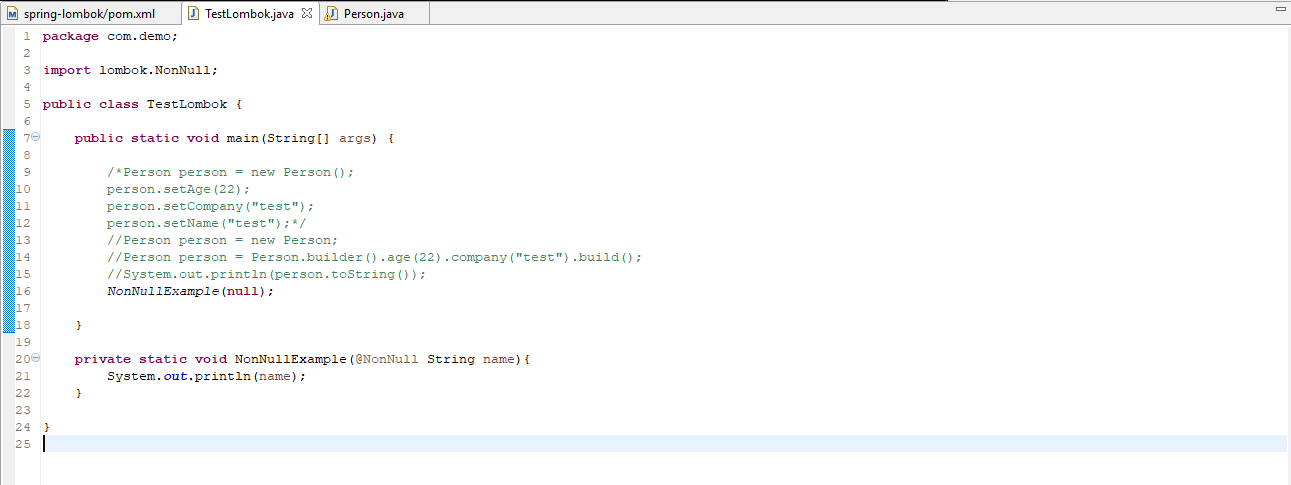
**@Builder**





**@NonNull**

You can use @NonNull on the parameter of a method or constructor to have lombok generate a null-check statement for you.



**@Log**

