Implementing Methods of an Interface

If a class is declared as implementing a certain interface or extending a class with abstract methods, it has to implement the methods of such interface or class. IntelliJ IDEA creates stubs of the implemented methods, with the default return values for the primitive types, and null values for the objects.

To implement methods of an interface or abstract class

1. Do one of the following:
   - On the main menu, choose **Code | Generate**, or press `Ctrl+I`.
   - Right-click the editor and choose **Generate** on the context menu, or press `Alt+Insert`, and choose **Implement Methods**.

   The **Select Methods To Implement** dialog appears, displaying the list of classes and interfaces with the methods that can be implemented.
2. Select one or more methods to implement. For multiple selection, use `Ctrl` and `Shift` keys.
3. If necessary, check the option **Copy JavaDoc** to insert JavaDoc comments from the implemented interface of abstract methods (if any).
4. Click **OK**.

File template responsible for implementing a method (*Implemented method body*) accepts predefined template variables from "File Header" (**File | Settings - IDE Settings - File Templates - Code - File Header**), e.g. ${USER}, ${DATE}, etc.

For example, consider the following file template:

```
    #if ( $RETURN_TYPE != "void" )return $DEFAULT_RETURN_VALUE;#end  // TODO ($USER, $DATE):To c
```

Provided that an implemented interface contains two methods, this template expands into the following code:

```
            @Override
            public void hunt() {
// TODO (mio, 9/21/12): To change the body of an implemented method, use File | Settings | File T
}
@Override
            public String sniff() {
return null;  // TODO (mio, 9/21/12): To change body of implemented methods use File | Settings |
            }
```

See Also

Reference:

- File and Code Templates

Web Resources:

- Developer Community