Rex Jones II

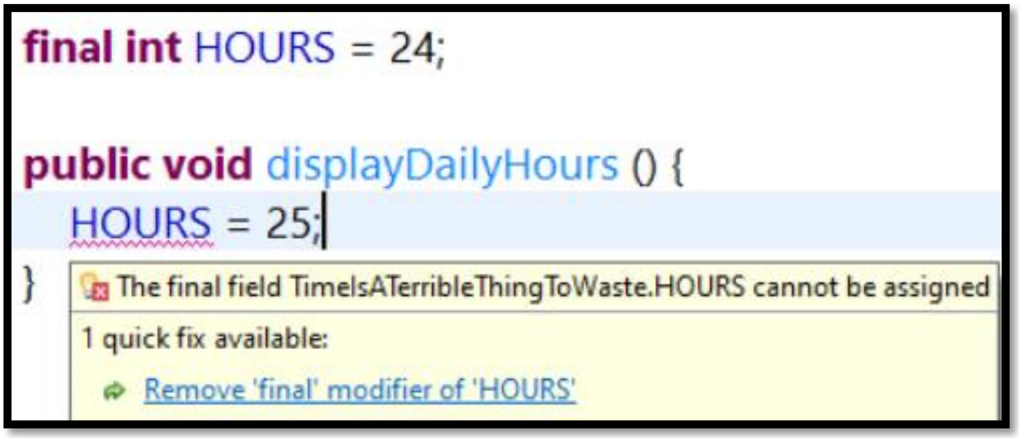
(Transcript)
Final / Constant

Let’s talk about the keyword final. Final is a way to prevent variables, methods, and classes from changing in our program. Variables are considered constants after applying final. A method cannot be overridden after applying final. When final is applied to classes, the class cannot be inherited by another class.

Starting with variables, we write final before the data type int. Conventionally, the name is written in all capital letters. HOURS = , capital letters identify the variable as a constant meaning it will not change. There are 24 hours in a day. The difference between final and non-final variable is a final variable cannot be reassigned. For example, let’s create a method called:

```
public void displayDailyHours () {
HOURS = 25;
}
```

Do you see the error? The error states “HOURS cannot be assigned”.



```
final int HOURS = 24;

public void displayDailyHours () {
    HOURS = 25;
}
```

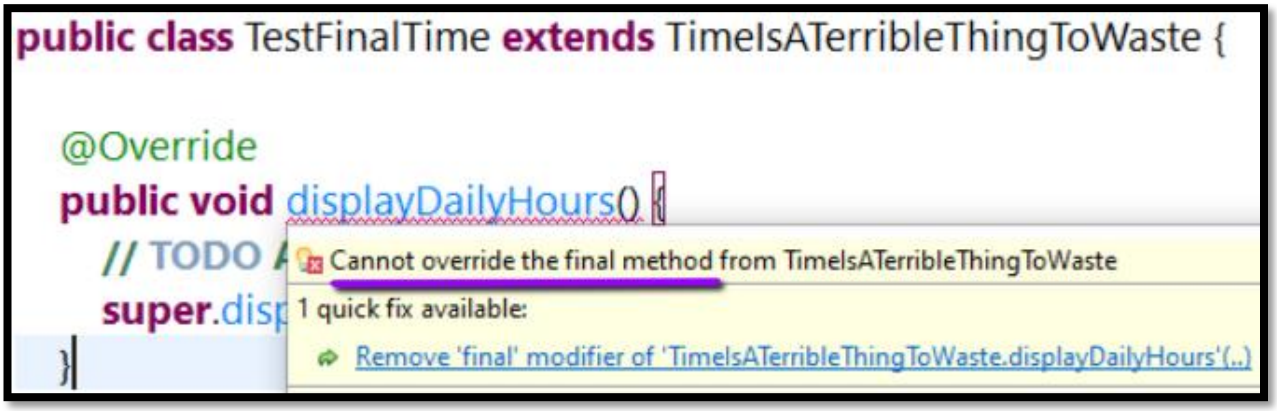
The final field TimeIsATerribleThingToWaste.HOURS cannot be assigned

1 quick fix available:

Remove 'final' modifier of 'HOURS'

Look what happens when I remove final. The error goes away. Change the value from 25 to 24 then Add final back and the error comes back. We are only allowed to initialize a final variable one time. Either directly or inside a method.

Final also prevents method overriding. Let's see what happens when I go to this other class TestFinalTime. Extend the parent class TimeIsATerribleThingToWaste then override the method by clicking CTRL + SPACE and navigating to displayDailyHours. You see there is no problem but if I go back to the parent class then add final to the method after public and save. Now, we see an error in the child class for method.
Hover the error and the error says Cannot override the final method.



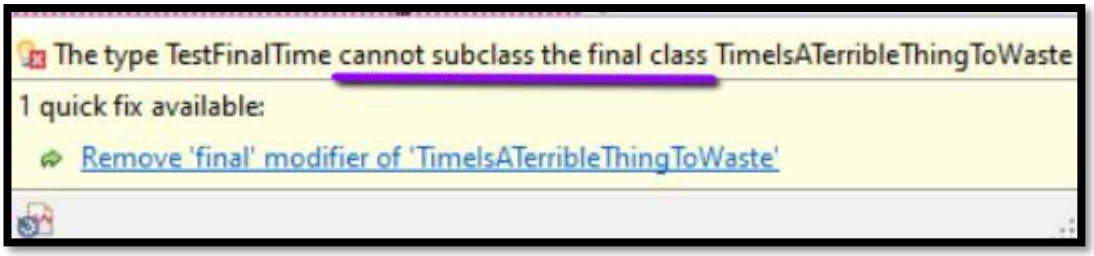
```
public class TestFinalTime extends TimeIsATerribleThingToWaste {

    @Override
    public void displayDailyHours() {
        // TODO A
        super.disp
    }
```

Cannot override the final method from TimeIsATerribleThingToWaste

1 quick fix available:

Remove 'final' modifier of 'TimeIsATerribleThingToWaste.displayDailyHours'(..)

Declaring a class as final, implicitly declares all of its methods as final. A final class prevent inheritance. If I add final between public and class, then we see an error that's different from the previous error and it says Cannot subclass the final class.

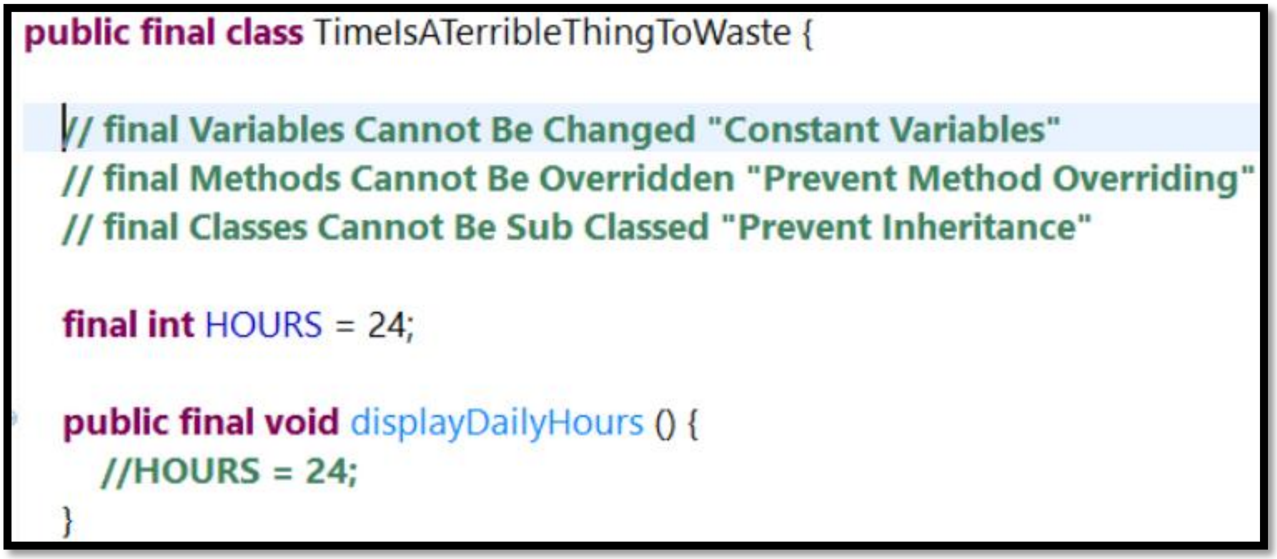


The type TestFinalTime cannot subclass the final class TimeIsATerribleThingToWaste

1 quick fix available:

Remove 'final' modifier of 'TimeIsATerribleThingToWaste'

To recap, let's go over this again. Final Variables Cannot Be Changed because they are considered Constant Variables. Final Methods Cannot Be Overridden Because so they Prevent Method Overriding, and final Classes Cannot Be SubClassed so they Prevent Inheritance.



```
public final class TimeIsATerribleThingToWaste {

    // final Variables Cannot Be Changed "Constant Variables"
    // final Methods Cannot Be Overridden "Prevent Method Overriding"
    // final Classes Cannot Be Sub Classed "Prevent Inheritance"

    final int HOURS = 24;

    public final void displayDailyHours () {
        //HOURS = 24;
    }
```

Thanks for watching. If you like the video then Connect with me on LinkedIn, follow me on Twitter, and subscribe to my YouTube channel. See you next time.

Social Media Contact

- YouTube https://www.youtube.com/c/RexJonesII/videos
- Twitter https://twitter.com/RexJonesII
- LinkedIn https://www.linkedin.com/in/rexjones34/
- GitHub https://github.com/RexJonesII/Free-Videos
- Facebook http://facebook.com/JonesRexII