

Folding

[Previous](#) | [Refactoring Basics](#)

[Top](#) | [Quick Start](#)

[Next](#) | [Find Usages](#)

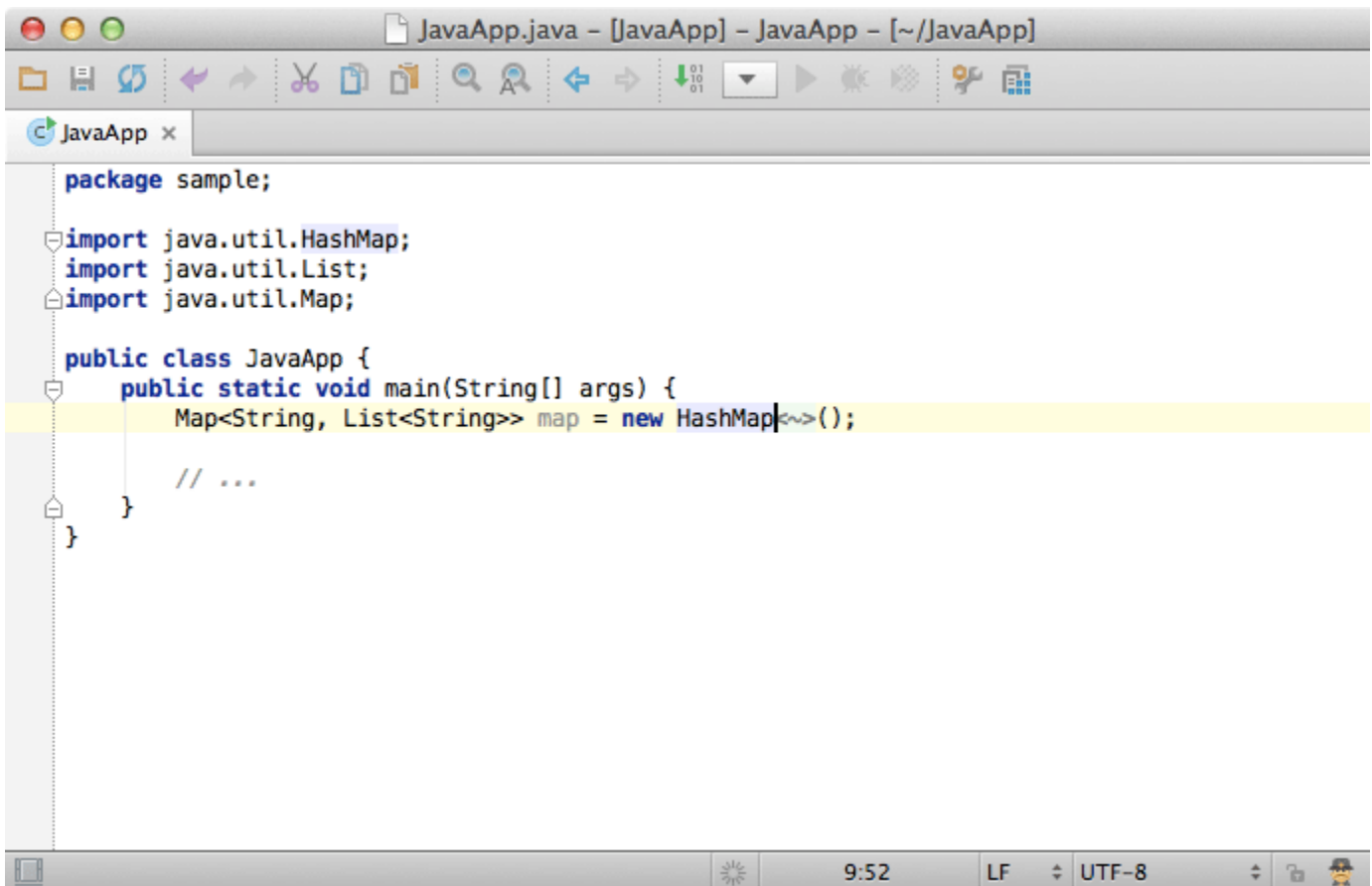
 **Redirection Notice**
This page will redirect to <https://www.jetbrains.com/idea/help/folding-and-expanding-code-blocks.html>.

Folding helps you keep your code neat and readable, by automatically collapsing the boilerplate or by replacing it with more relevant statements.

You can see the original code behind the folded regions by placing the caret on it, or by moving the mouse over the collapsed pieces of code. Additionally you can always expand or collapse the folded blocks at the caret by calling the Toggle folding action via the shortcut `Ctrl + .` (`Cmd + .` for Mac).

1. Generics

A very basic example of code folding is when the editor replaces parameter types in generics with the diamond operator added in Java 7. The code itself remains the same but it looks shorter and clearer.



```
package sample;

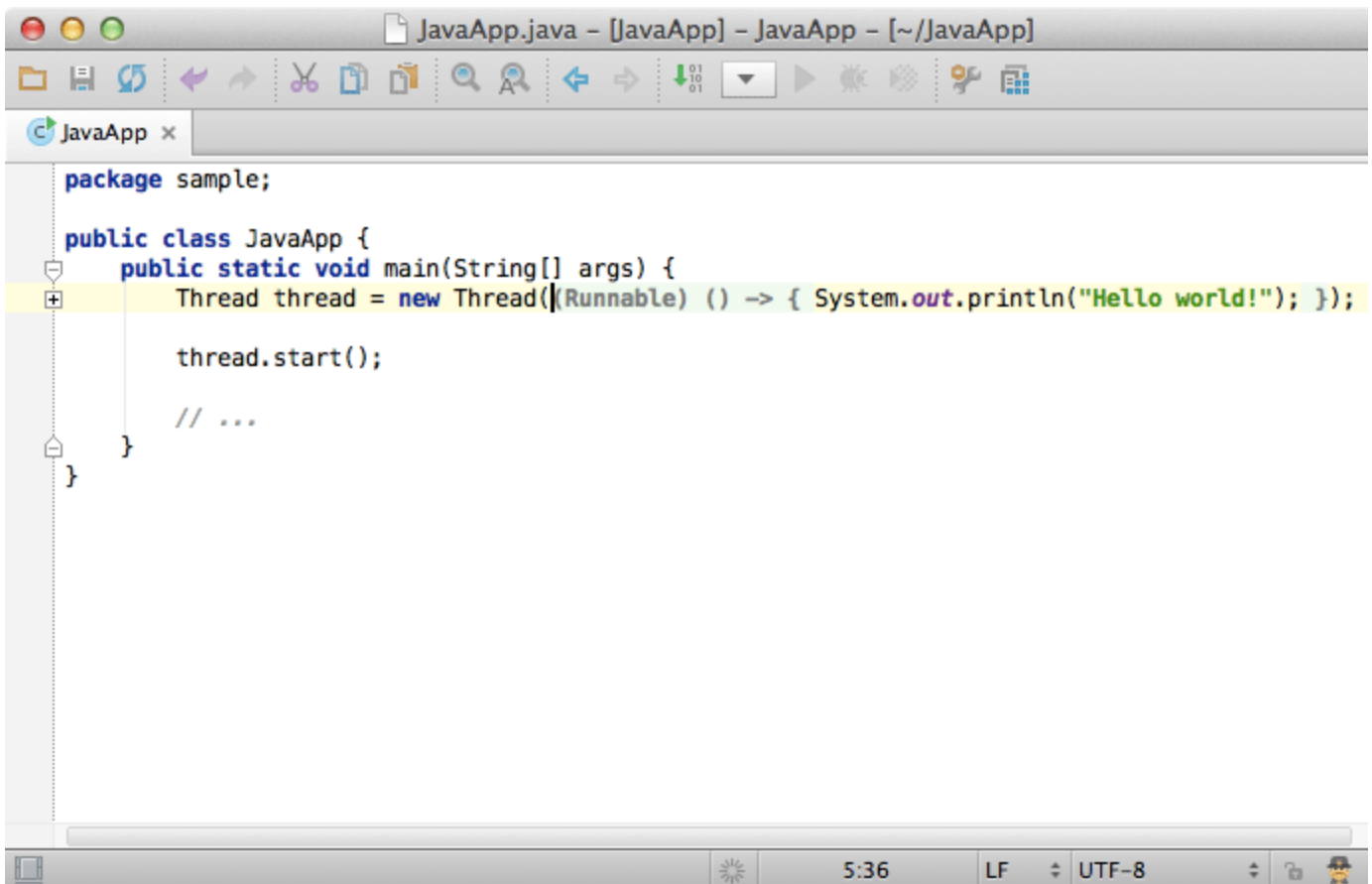
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class JavaApp {
    public static void main(String[] args) {
        Map<String, List<String>> map = new HashMap<>();

        // ...
    }
}
```

2. Anonymous classes

Another sample of code folding is when anonymous classes definition is replaced with the lambda notation added in Java 8. The lambda notation makes the code much shorter.



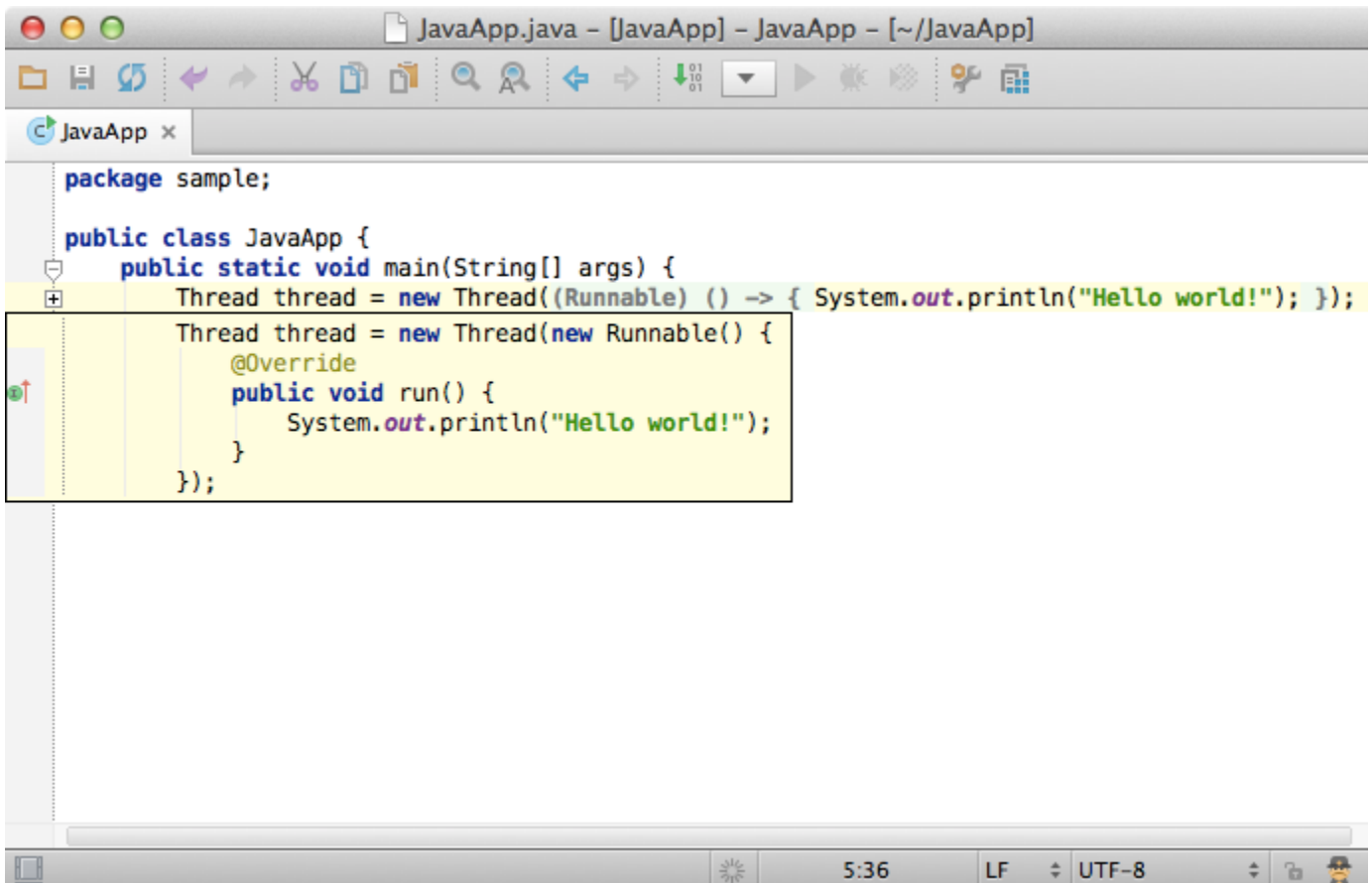
```
package sample;

public class JavaApp {
    public static void main(String[] args) {
        Thread thread = new Thread((Runnable) () -> { System.out.println("Hello world!"); });

        thread.start();

        // ...
    }
}
```

When you move the mouse over this or any other folded piece of code, you can still see the original code.



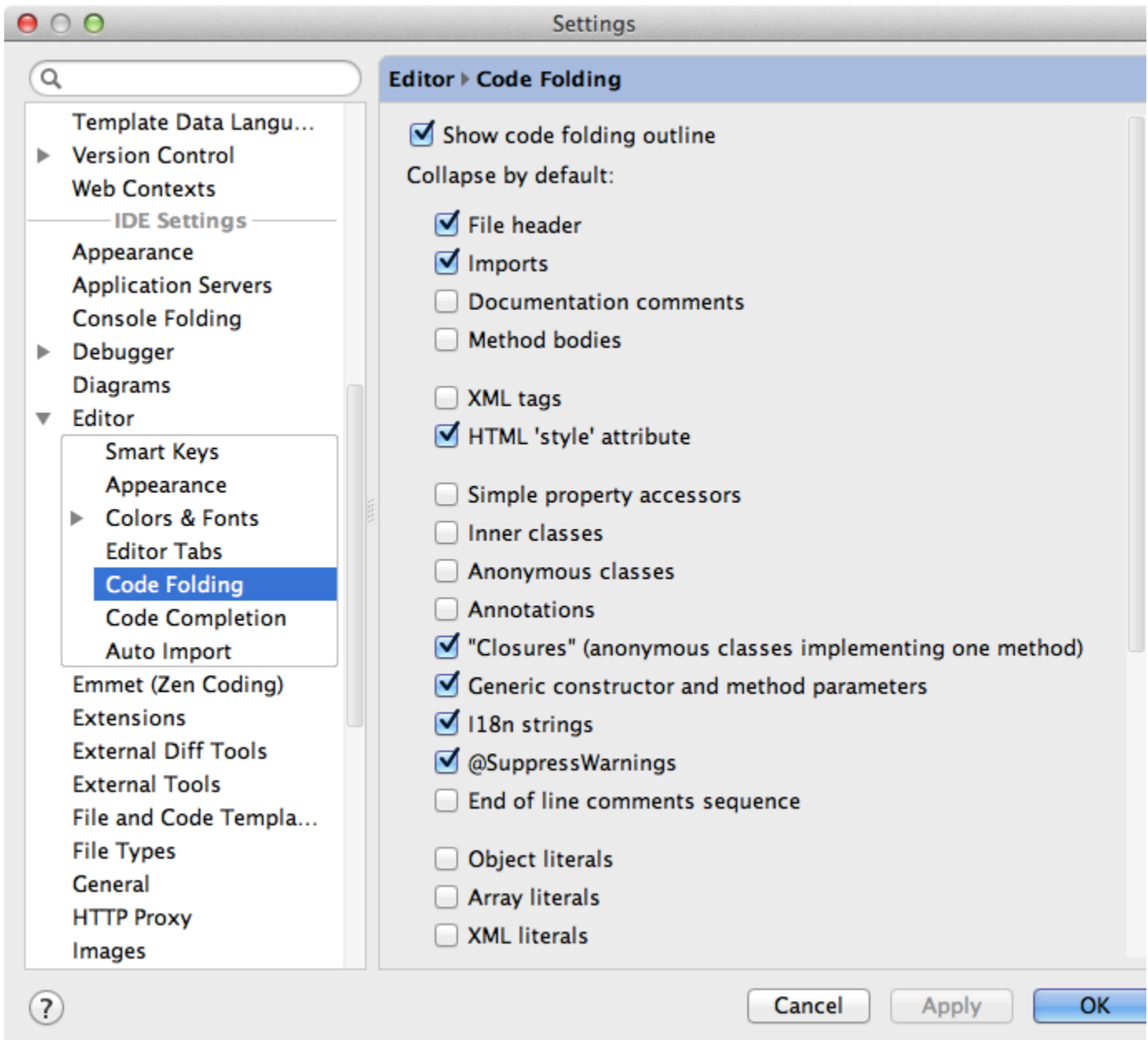
```
package sample;

public class JavaApp {
    public static void main(String[] args) {
        Thread thread = new Thread((Runnable) () -> { System.out.println("Hello world!"); });

        Thread thread = new Thread(new Runnable() {
            @Override
            public void run() {
                System.out.println("Hello world!");
            }
        });
    }
}
```

3. Folding settings

You can always change the default behavior of code folding for particular statements via Settings Editor Code Folding.



4. Other useful actions

- Expand All via Shift + Ctrl + + (Shift + Cmd + + for Mac) automatically expands all collapsed blocks in the opened file.