# Table of Contents

# Perface

Annotation are used to provide supplement information about a program. Java Annotation is a tag that represents the metadata (information) for class, interface, methods, fields to indicate some additional information which can be used by java compiler and JVM.

Annotation start with "@", annotation not change action of a compiled program.

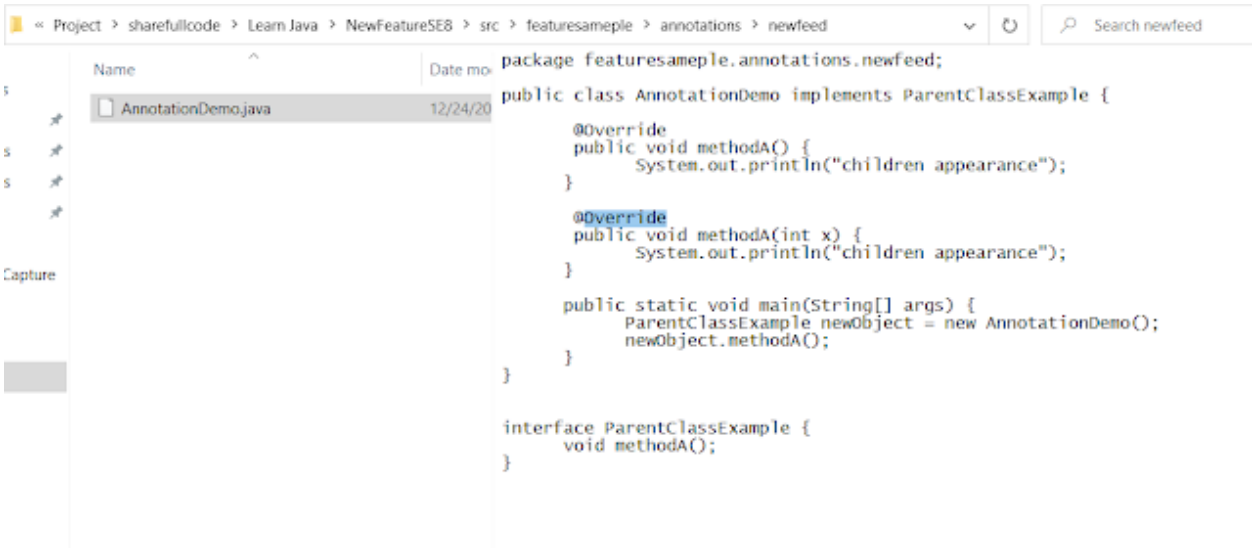
Built-in Java Annotations : Override, SuppressWarnings, Deprecated.

Built-in Java Annotations used in others annotations (customize annotation): Target, Retention, Inherited, Documented.

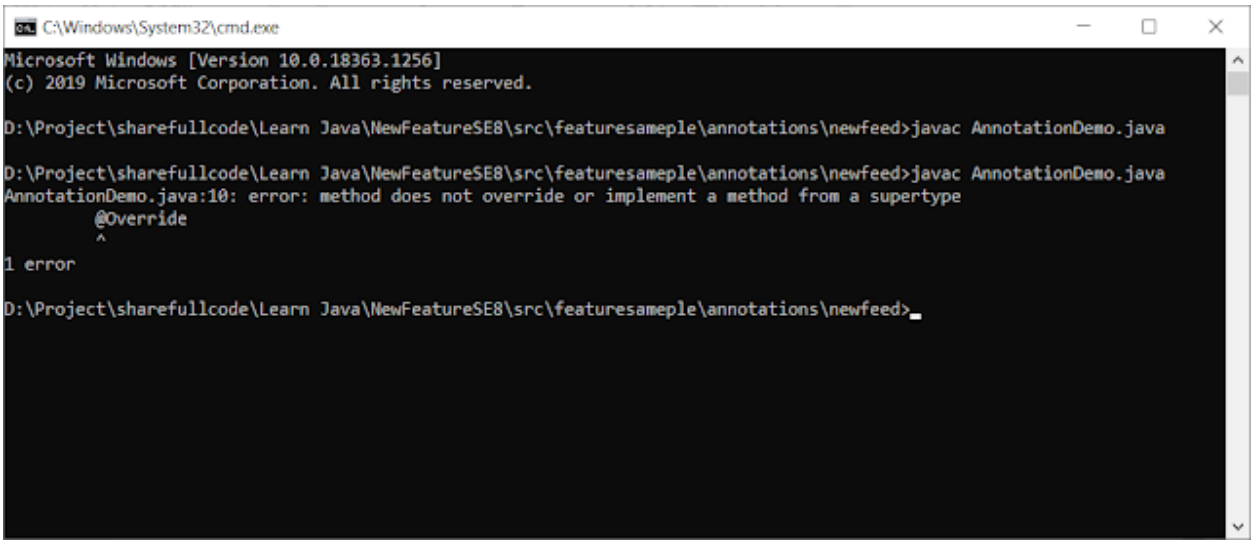
# Built-in Java Annotation:

## @Override

Annotation assures that the subclass method is overriding the parent method. if it not show, compile time error occurs. it example bellow show error complier with Overload one function not have in interface. if you want to remove error complie, you should remove @Override in function overload.




Error during complie Javac,




## @SuppressWarnings

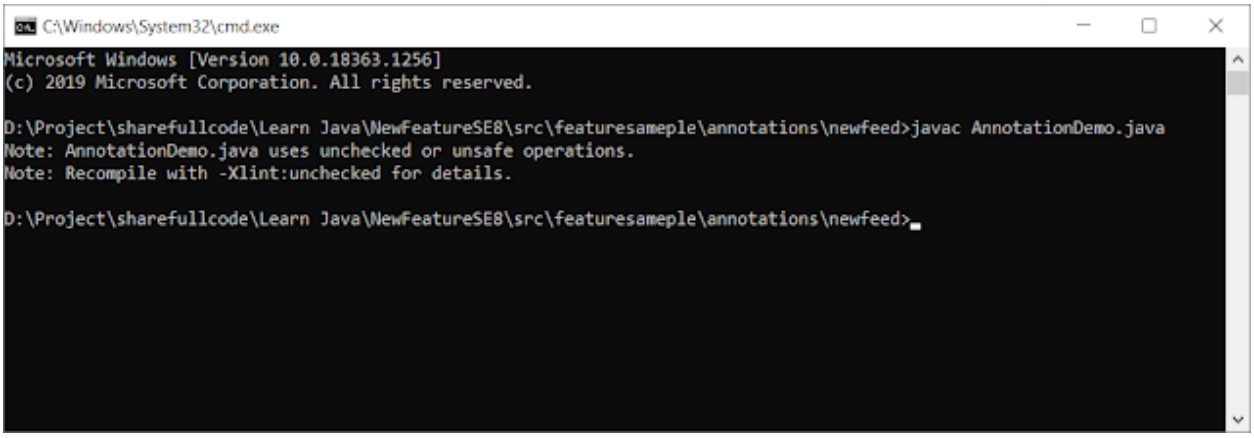
Annotation: is used to suppress warnings issued by the complier. This type of annotation can be applied to any type of declaration.

This example bellow illustration suppress warning during check javac

FlatMapSample.java | AnnotationDemo.java | ParallelStreamSample.java

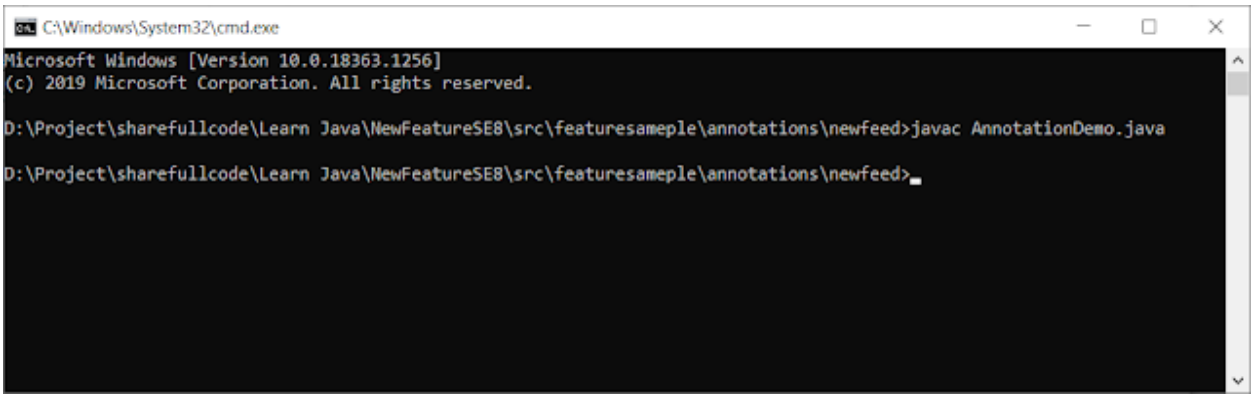
```
8  public class AnnotationDemo implements ParentClassExample {
9
10     @Override
11     public void methodA() {
12         System.out.println("children appearance");
13     }
14
15     /*
16      * @Override public void methodA(int x) {
17      * System.out.println("children appearance"); }
18      */
19
20     void suppressWarnigsDemo() {
21         List list= new ArrayList();
22         list.add("a1");
23         list.add("a2");
24         list.add("a3");
25         list.add("a4");
26         list.forEach(System.out::println);
27     }
28
29     public static void main(String[] args) {
30         AnnotationDemo newObject = new AnnotationDemo();
31         newObject.suppressWarnigsDemo();
32     }
33 }
```

Complie print warning, but complier still success




Add annotation @SuppressWarning to complie pass not appearance warning

Complie Successful



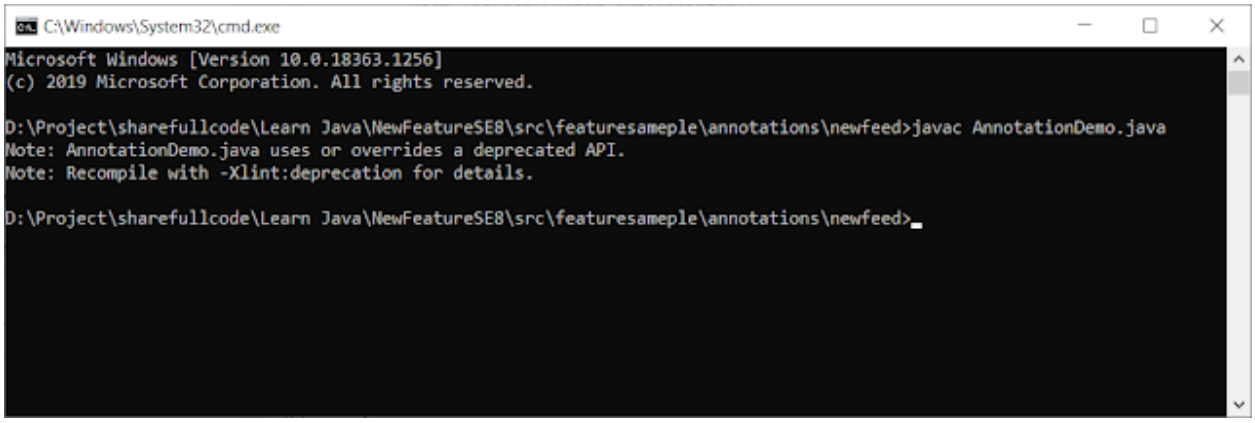

## @Deprecated

Annotation marks that this method, field deprecated so complier print warning. in the Future method, filed not use so that now you do not use it.

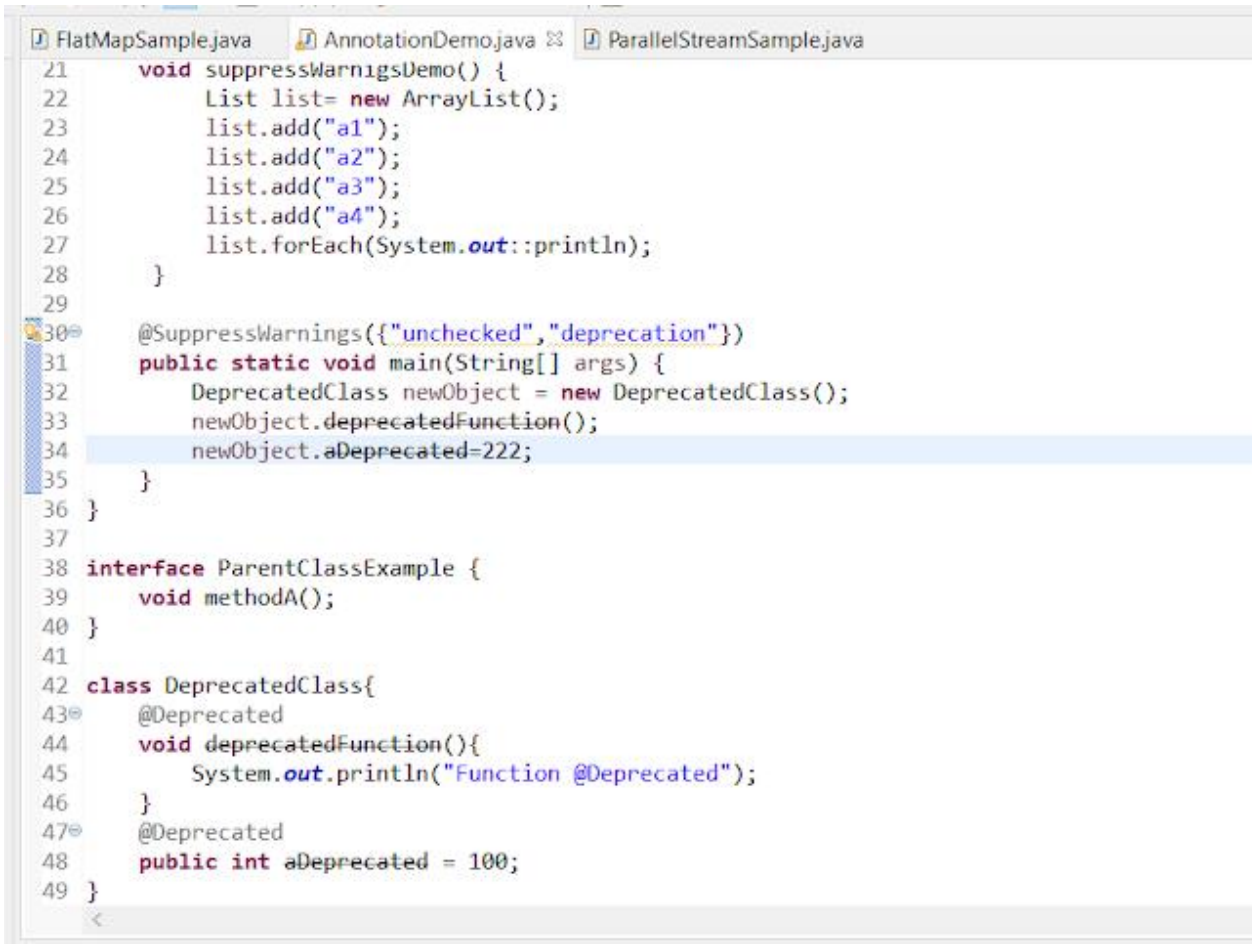
FlatMapSample.java | AnnotationDemo.java | ParallelStreamSample.java

```
20    @SuppressWarnings({ "unchecked", "rawtypes"})
21    void suppressWarnigsDemo() {
22         List list= new ArrayList();
23         list.add("a1");
24         list.add("a2");
25         list.add("a3");
26         list.add("a4");
27         list.forEach(System.out::println);
28     }
29
30    public static void main(String[] args) {
31        DeprecatedClass newObject = new DeprecatedClass();
32        newObject.deprecatedFunction();
33        newObject.aDeprecated=222;
34    }
35 }
36
37 interface ParentClassExample {
38    void methodA();
39 }
40
41 class DeprecatedClass{
42    @Deprecated
43    void deprecatedFunction(){
44        System.out.println("Function @Deprecated");
45    }
46    @Deprecated
47    public int aDeprecated = 100;
48 }
```

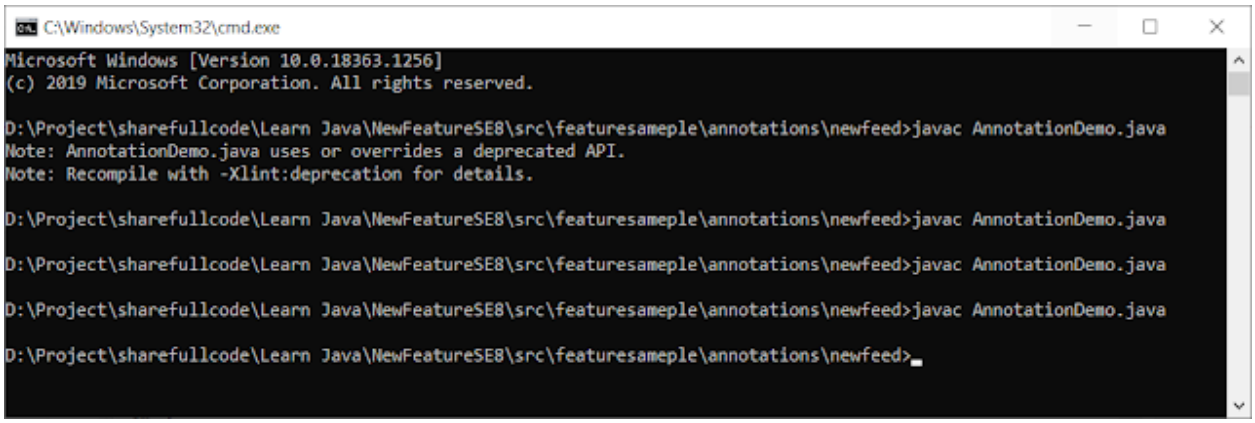
Complie print waring




To fix not appearance warning to add @SuppressWaring in functions are using Deprecate function. But it waring in source in Eclipse IDE it said "remove @SuppressWarning" it not importain , but the best not use Deprecated for your source.



Complie not print warning




# Use Library The Checker Framwork

## Introduce :

https://checkerframework.org Last Update (1 Dec 2020)

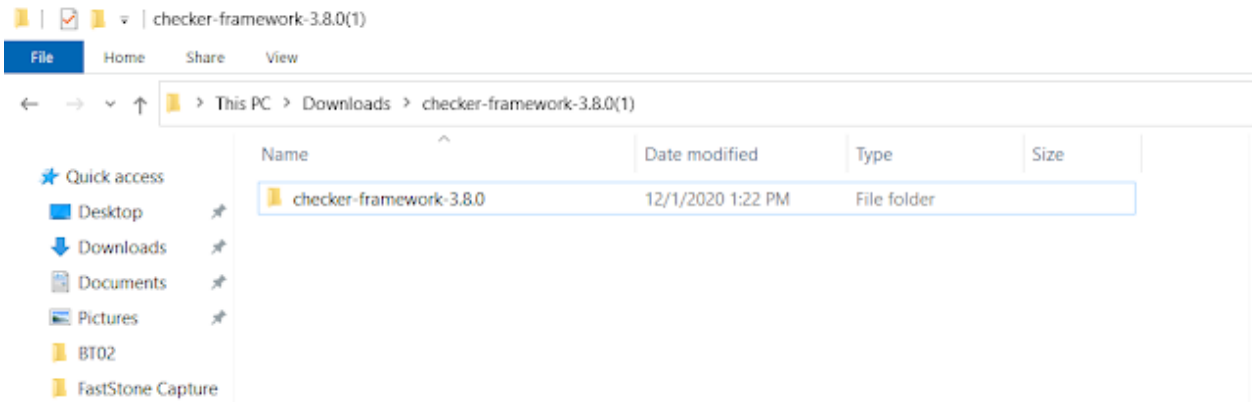
Features : Check null exception, unintended side effects, SQL ịnections, concurrent error, mistaken equality tests.

Summary : The checker framwork help programmer detect and prevent errors in their java programs.

Requiment: JDK 8, JDK 11.

## Install

Download file zip : https://checkerframework.org/checker-framework-3.8.0.zip
Unzip it to create a checker-framwork-3.8.9 directory

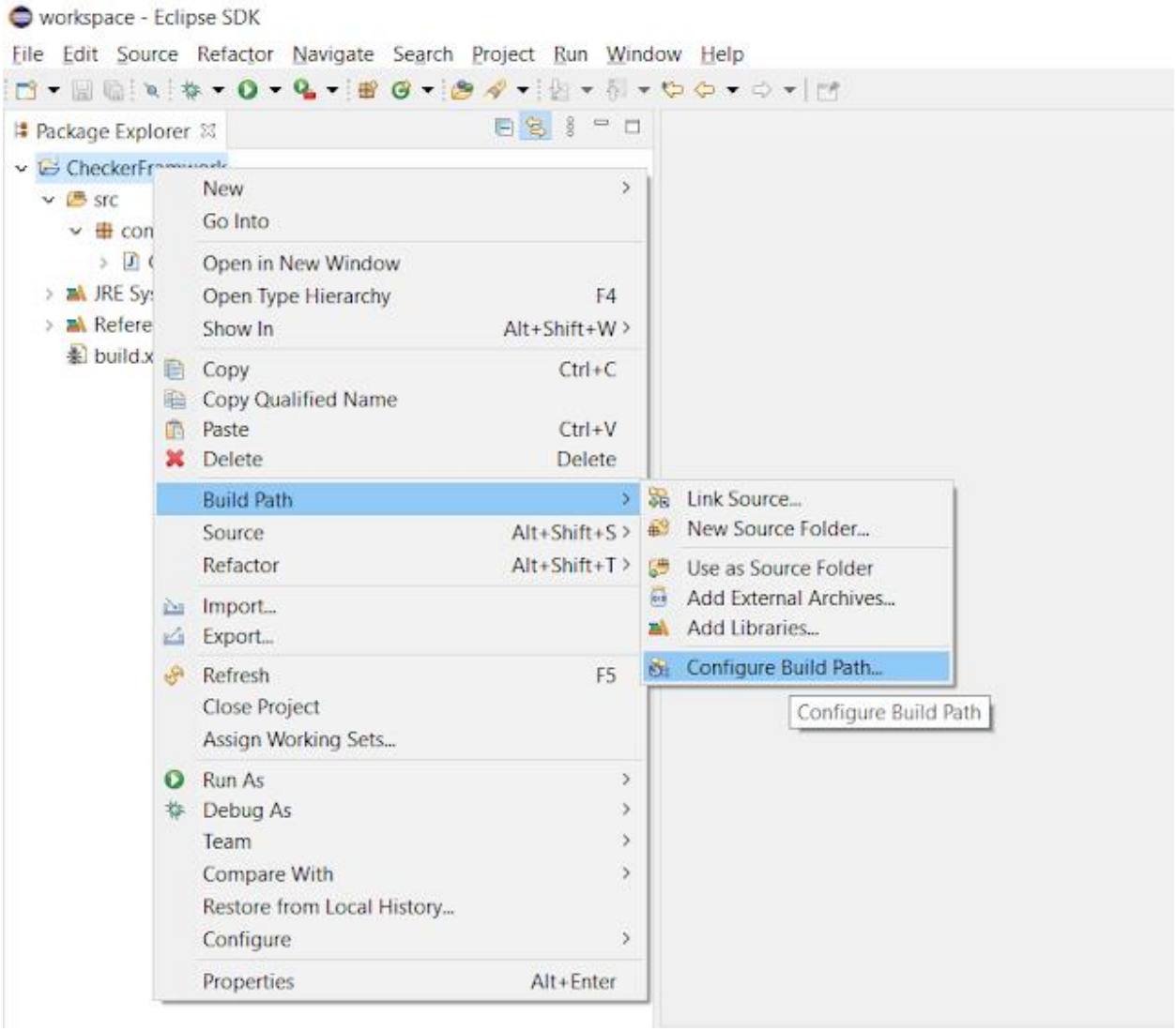


## Config your IDE

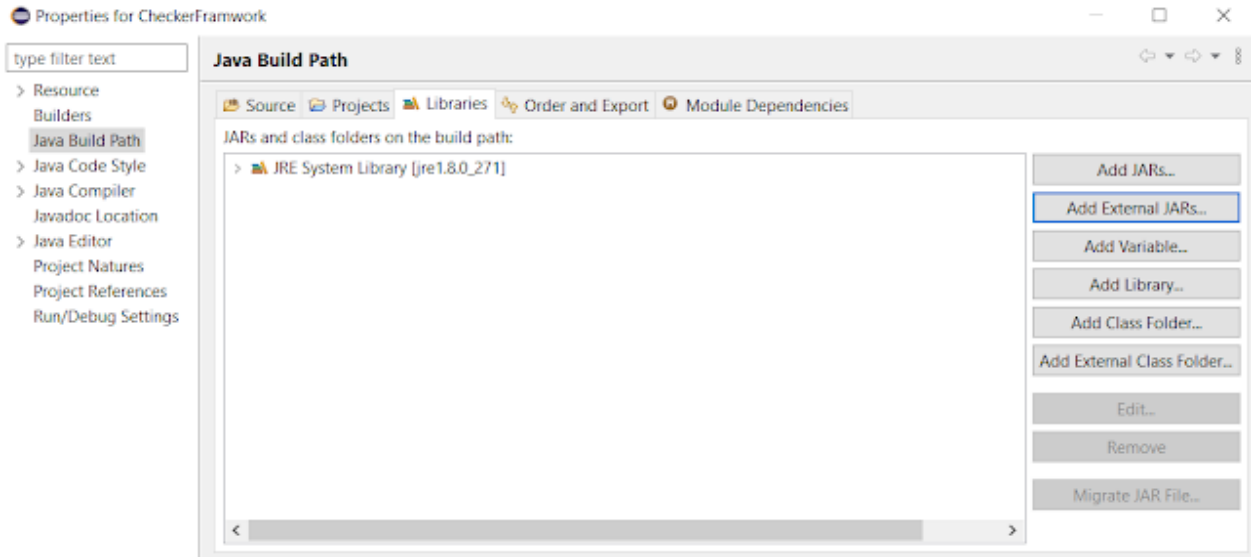
Build System, or command shell to include the Checker Framawork on the classpath. In this use Eclipse if you other you can ref

https://checkerframework.org/manual/#external-tools

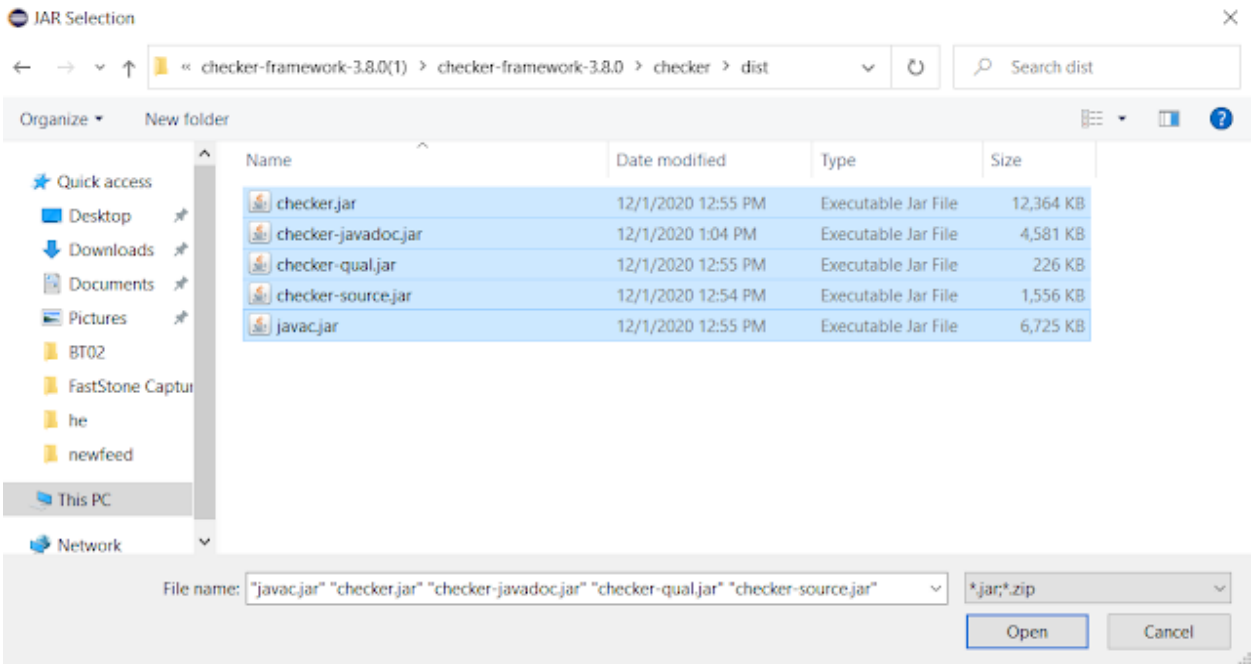
Add buildPath in project



Choice Add Extenal JARs that add File JAR




Ctrl + A and Open all



## File Complied

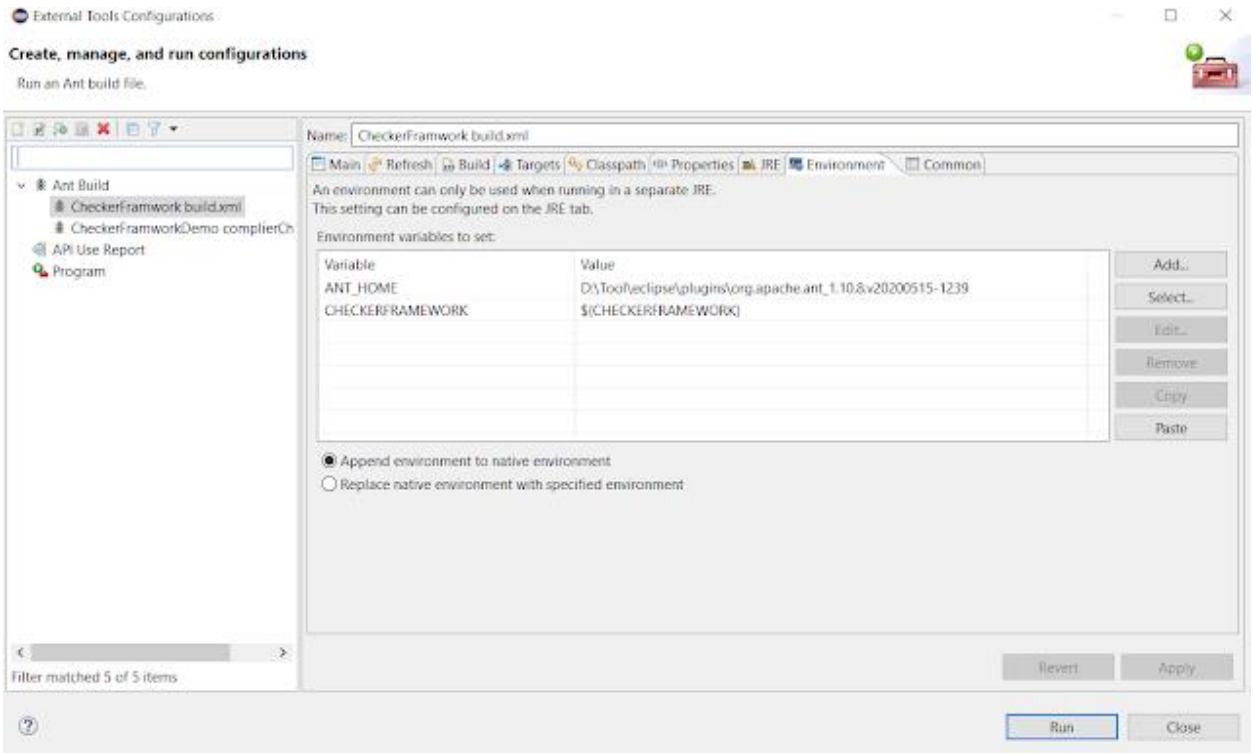
Config file Ant That source is complied here i use Ant to complie . I create file build.xml and run it by Ant

CheckerDemoBasic.java | build.xml

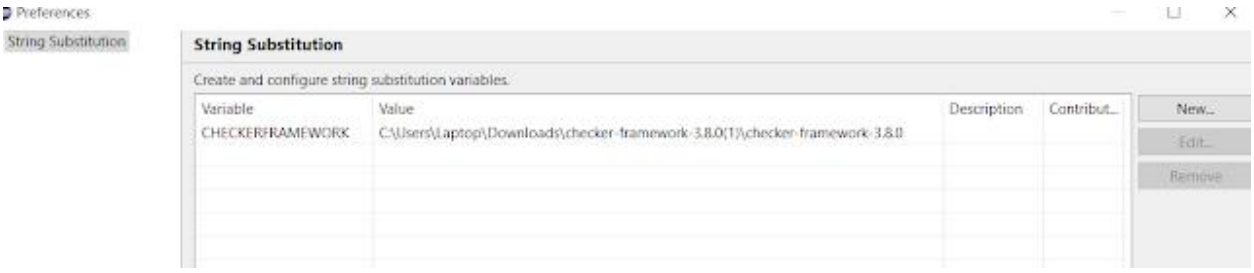
```
<?xml version="1.0" encoding="UTF-8"?>
<project name="CheckerFramwork" default="CheckerFramwork" basedir=".">

    <!-- 1.0 config path enviroment for javac in checkerframwork -->
    <property environment="env"/>
    <property name="checkerframework" value="${env.CHECKERFRAMEWORK}" />
    <condition property="cfJavac" value="javac.bat" else="javac">
        <os family="windows" />
    </condition>

    <!-- 2.0 pre set defination of tag -->
    <presetdef name="jsr308.javac">
        <javac fork="yes" executable="${checkerframework}\checker\bin\${cfJavac}" >
            <compilerarg value="-version"/>
            <compilerarg value="-implicit:class"/>
        </javac>
    </presetdef>

    <!-- 3.0 defaut run with ant run clean then run check-nullness -->
    <target name="CheckerFramwork" depends="clean, check-nullness">
          <echo>${checkerframework}</echo>
    </target>

    <target name="clean">
        <delete dir="build"/>
    </target>

    <!-- 4.0 compile check-nullness -->
    <target name="check-nullness"
              description="Check for null pointer dereferences"
              depends="clean">
        <!-- use jsr308.javac instead of javac -->
        <jsr308.javac srcdir =".">
            <compilerarg line="-processor org.checkerframework.checker.nullness.NullnessChecker"/>
            <!-- optional, to not check uses of library methods:
            <compilerarg value="-AskipUses=^(java\.awt\.|javax\.swing\.)"/>
          -->
            <compilerarg line="-Xmaxerrs 10000"/>
        </jsr308.javac>
    </target>
</project>
```

1.0 Set enviroment for variable CHECKERFRAMEWORK in Ant config

Choice file build.xml open External Tools Configurations

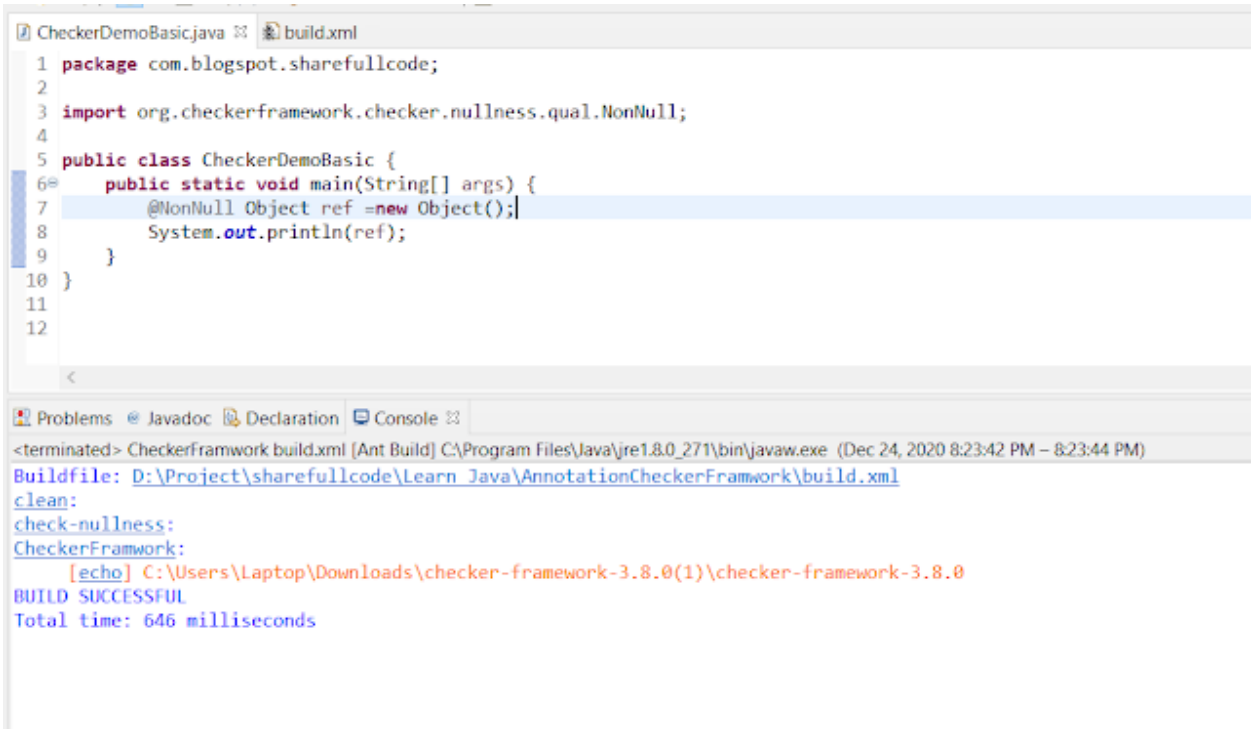



New variable CHECKERFRAMEWORK with path fot it directory checker-framwork was unzip before.



## Test Source

Choice build.xml run with ant and complie successful




But change object = null complie error

**This section is non commercial mainly sharing and advance knowlage for java.This tutorials has referenced document from the list below if you has complain for license, i will remove all from internet. Thank you all everything.**

https://www.javatpoint.com/java-annotation
https://www.geeksforgeeks.org/annotations-in-java/