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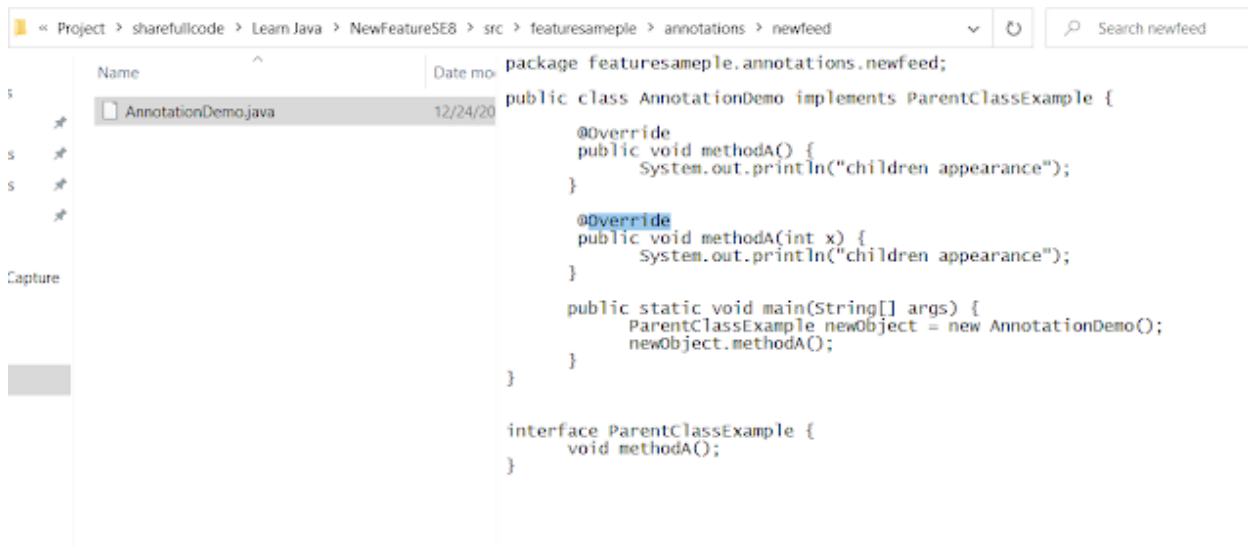
Annotation start with "@", annotation not change action of a compiled program.

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.


```
package featuresameple.annotations.newfeed;

public class AnnotationDemo implements ParentClassExample {

        @Override
        public void methodA() {
                System.out.println("children appearance");
        }

        @Override
        public void methodA(int x) {
                System.out.println("children appearance");
        }

        public static void main(String[] args) {
                ParentClassExample newObject = new AnnotationDemo();
                newObject.methodA();
        }
}

interface ParentClassExample {
        void methodA();
}
```

Error during complie Javac,

```
C:\Windows\System32\cmd.exe                                                    —    □    X

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>javac AnnotationDemo.java

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>javac AnnotationDemo.java
AnnotationDemo.java:10: error: method does not override or implement a method from a supertype
        @Override
        ^
1 error

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>
```

@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
   7
   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

```
C:\Windows\System32\cmd.exe                                              —    □    ×

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>javac AnnotationDemo.java
Note: AnnotationDemo.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>_
```

Add annotation @SuppressWarning to complie pass not appearance warning

```
 FlatMapSample.java      AnnotationDemo.java ⊠    ParallelStreamSample.java
  8
  9⊖        @Override
⊿10         public void methodA() {
 11             System.out.println("children appearance");
 12         }
 13
 14⊖        /*
 15          * @Override public void methodA(int x) {
 16          * System.out.println("children appearance"); }
 17          */
 18
 19
 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             AnnotationDemo newObject = new AnnotationDemo();
 32             newObject.suppressWarnigsDemo();
 33         }
 34 }
```

Complie Successful

```
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>javac AnnotationDemo.java

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>_
```

@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

```
C:\Windows\System32\cmd.exe                                          —    □    ✕

Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>javac AnnotationDemo.java
Note: AnnotationDemo.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

D:\Project\sharefullcode\Learn Java\NewFeatureSE8\src\featuresameple\annotations\newfeed>_
```

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 importian , but the best not use Deprecated for your source .


```
 FlatMapSample.java     AnnotationDemo.java ☒  ParallelStreamSample.java
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⊝    @SuppressWarnings({"unchecked","deprecation"})
31        public static void main(String[] args) {
32            DeprecatedClass newObject = new DeprecatedClass();
33            newObject.deprecatedFunction();
34            newObject.aDeprecated=222;
35        }
36 }
37
38 interface ParentClassExample {
39        void methodA();
40 }
41
42 class DeprecatedClass{
43⊝        @Deprecated
44        void deprecatedFunction(){
45            System.out.println("Function @Deprecated");
46        }
47⊝        @Deprecated
48        public int aDeprecated = 100;
49 }
```

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 įnections, concurrent error, mistaken equality tests.
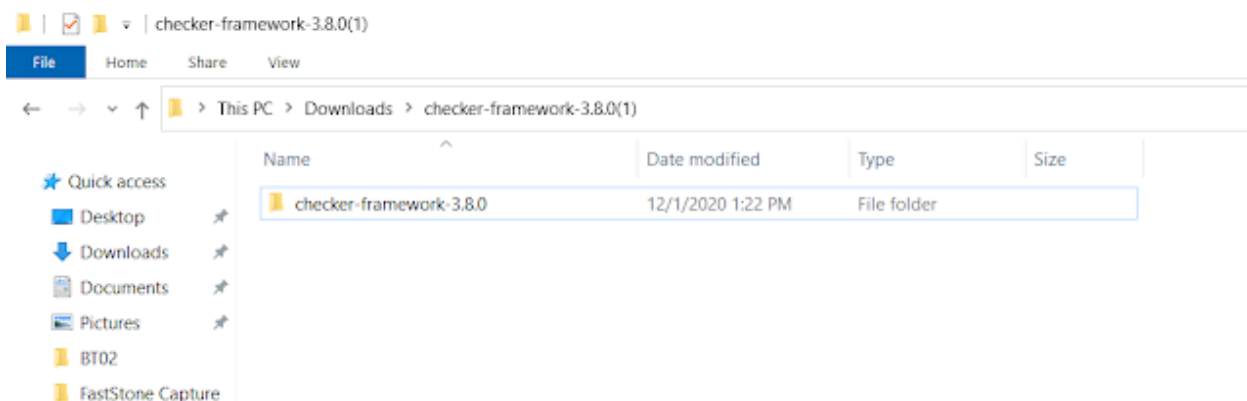
Summary : The checker framwork help programmer detect and prevent errors in their java programs.

Requiment: JDK 8, JDK 11.

Installation

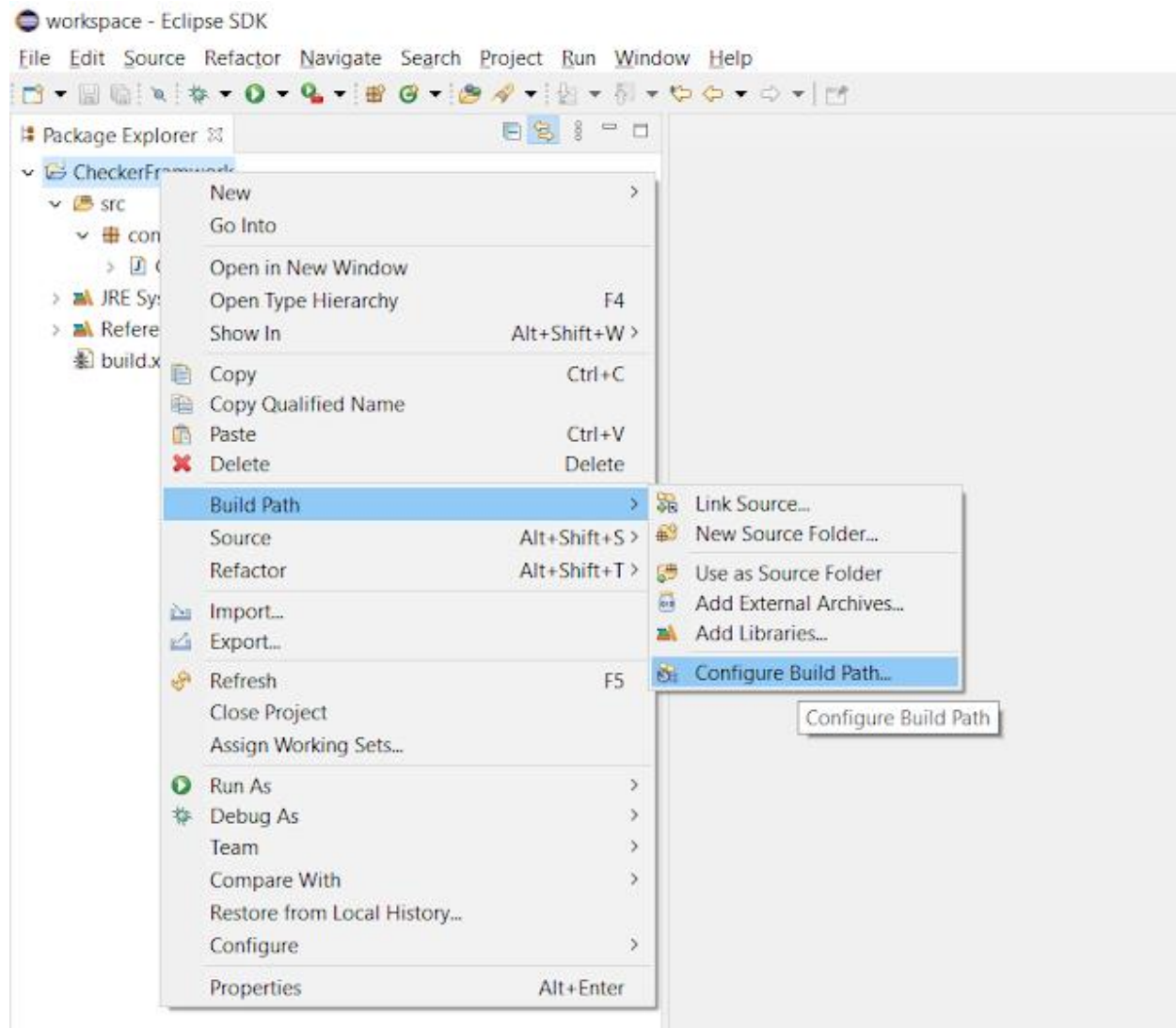1 Download file zip : https://checkerframework.org/checker-framework-3.8.0.zip
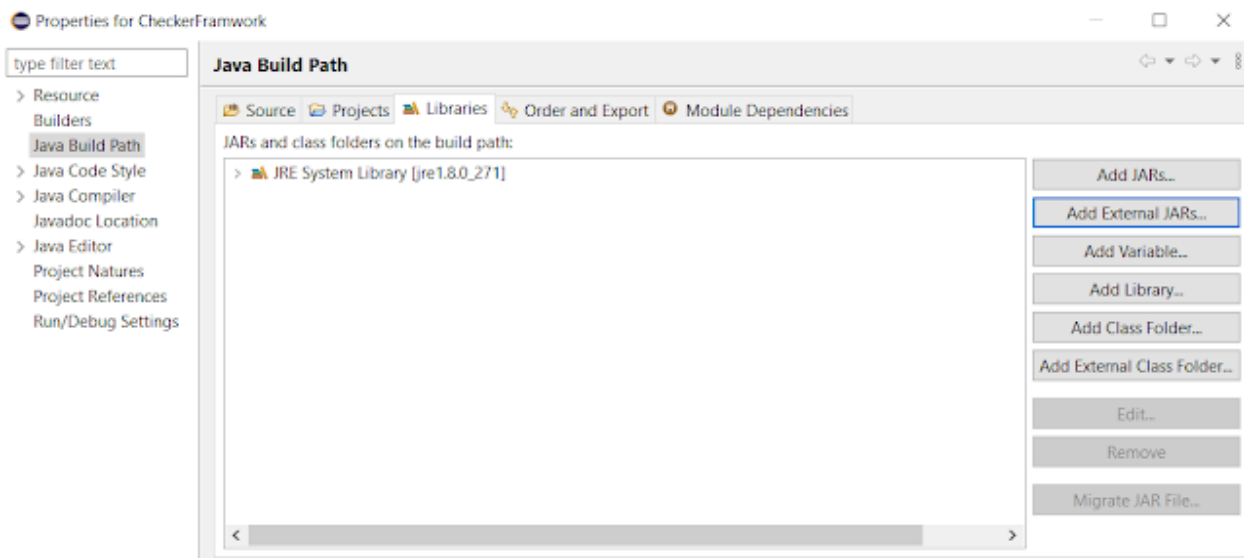
2 Unzip it to create a checker-framwork-3.8.9 directory

3 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
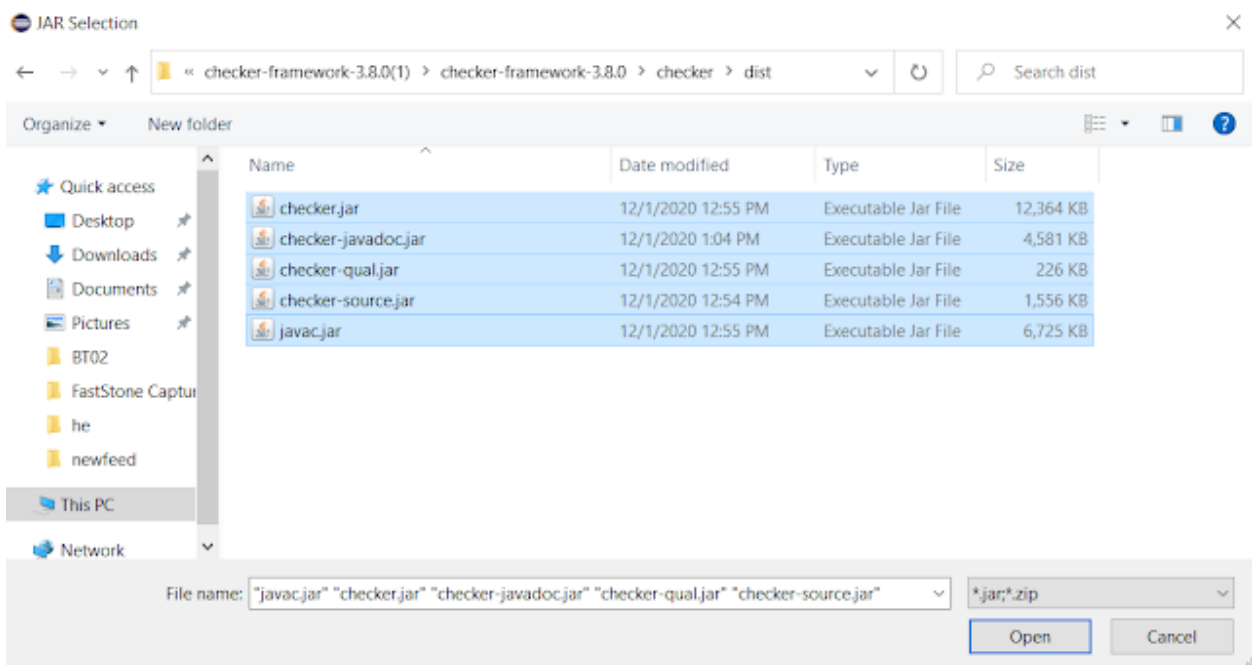
https://checkerframework.org/manual/#external-tools

Add buildPath in project

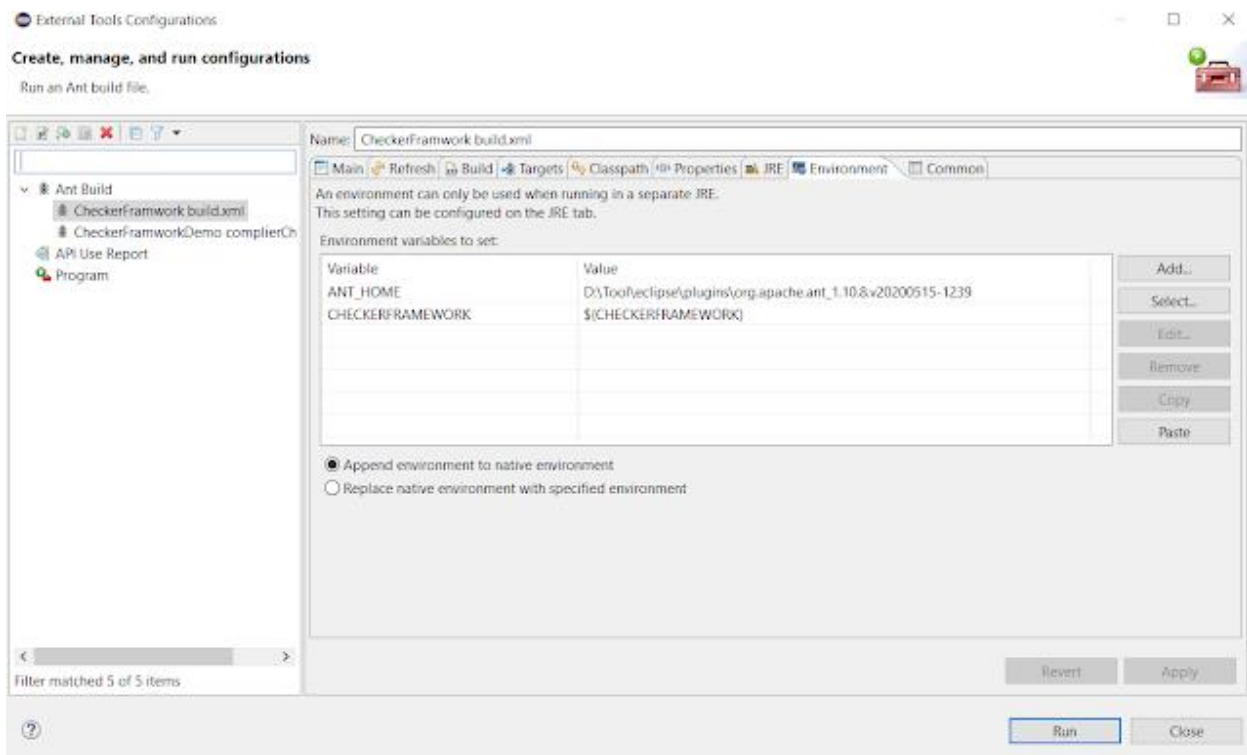Choice Add Extenal JARs that add File JAR

Ctrl + A and Open all

Config file Ant That source is complied here i use Ant to complie . I create file build.xml and run it by Ant
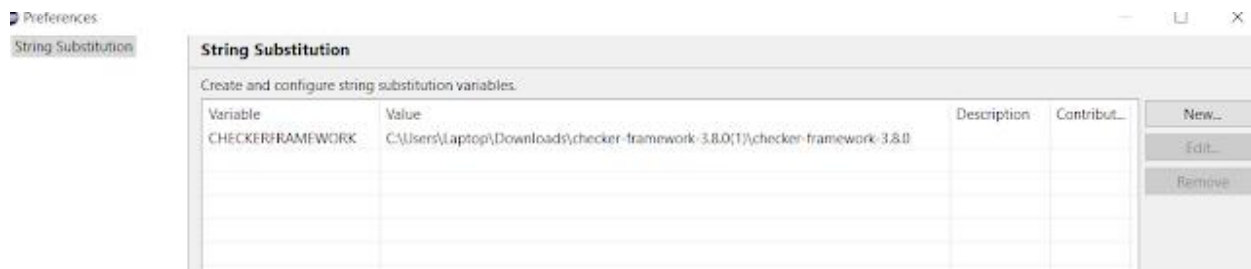
```xml
  □ CheckerDemoBasic.java ⊠  ⓐ build.xml ⊠
   1  <?xml version="1.0" encoding="UTF-8"?>
   2⊖ <project name="CheckerFramwork" default="CheckerFramwork" basedir=".">
   3
   4      <!-- 1.0 config path enviroment for javac in checkerframwork -->
   5      <property environment="env"/>
   6      <property name="checkerframework" value="${env.CHECKERFRAMEWORK}" />
   7⊖     <condition property="cfJavac" value="javac.bat" else="javac">
   8          <os family="windows" />
   9      </condition>
  10
  11      <!-- 2.0 pre set defination of tag -->
  12⊖     <presetdef name="jsr308.javac">
  13⊖         <javac fork="yes" executable="${checkerframework}\checker\bin\${cfJavac}" >
  14              <compilerarg value="-version"/>
  15              <compilerarg value="-implicit:class"/>
  16          </javac>
  17      </presetdef>
  18
  19      <!-- 3.0 defaut run with ant run clean then run check-nullness -->
  20⊖     <target name="CheckerFramwork" depends="clean, check-nullness">
  21              <echo>${checkerframework}</echo>
  22      </target>
  23
  24⊖     <target name="clean">
  25          <delete dir="build"/>
  26      </target>
  27
  28      <!-- 4.0 compile check-nullness -->
  29⊖     <target name="check-nullness"
  30              description="Check for null pointer dereferences"
  31              depends="clean">
  32          <!-- use jsr308.javac instead of javac -->
  33⊖         <jsr308.javac srcdir =".">
  34              <compilerarg line="-processor org.checkerframework.checker.nullness.NullnessChecker"/>
  35⊖             <!-- optional, to not check uses of library methods:
  36              <compilerarg value="-AskipUses=^(java\.awt\.|javax\.swing\.)"/>
  37          -->
  38              <compilerarg line="-Xmaxerrs 10000"/>
  39          </jsr308.javac>
  40      </target>
  41 </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

```
 1  package com.blogspot.sharefullcode;
 2
 3  import org.checkerframework.checker.nullness.qual.NonNull;
 4
 5  public class CheckerDemoBasic {
 6⊖     public static void main(String[] args) {
 7          @NonNull Object ref =new Object();
 8          System.out.println(ref);
 9      }
10  }
11
12
```

🔲 Problems ● Javadoc 🔍 Declaration 🖥 Console ⊠

<terminated> CheckerFramwork build.xml [Ant Build] C:\Program Files\Java\jre1.8.0_271\bin\javaw.exe (Dec 24, 2020 8:23:42 PM – 8:23:44 PM)

```
Buildfile: D:\Project\sharefullcode\Learn Java\AnnotationCheckerFramwork\build.xml
clean:
check-nullness:
CheckerFramwork:
    [echo] C:\Users\Laptop\Downloads\checker-framework-3.8.0(1)\checker-framework-3.8.0
BUILD SUCCESSFUL
Total time: 646 milliseconds
```

But change object = null complie error

```
 1  package com.blogspot.sharefullcode;
 2
 3  import org.checkerframework.checker.nullness.qual.NonNull;
 4
 5
 6
 7  public class CheckerDemoBasic {
 8⊖     public static void main(String[] args) {
 9          @NonNull Object ref =new Object();
10          System.out.println(ref);
11          ref=null;
12      }
13  }
```

🔲 Problems ● Javadoc 🔍 Declaration 🖥 Console ⊠

<terminated> CheckerFramwork build.xml [Ant Build] C:\Program Files\Java\jdk1.8.0_271\bin\javaw.exe (Dec 24, 2020 8:50:43 PM – 8:50:50 PM)

```
Buildfile: D:\Project\sharefullcode\Learn Java\AnnotationCheckerFramwork\build.xml
clean:
check-nullness:
[jsr308.javac] Compiling 1 source file
[jsr308.javac] javac (version info not available)
[jsr308.javac] D:\Project\sharefullcode\Learn Java\AnnotationCheckerFramwork\src\com\blogspot\sharefullcode\CheckerDemoBasic.java:11: error: [assignment.type.incompatibl
[jsr308.javac]            ref=null;
[jsr308.javac]               ^
[jsr308.javac]    found   : null
[jsr308.javac]    required: @UnknownInitialization @NonNull Object
[jsr308.javac] 1 error

BUILD FAILED
D:\Project\sharefullcode\Learn Java\AnnotationCheckerFramwork\build.xml:34: Compile failed; see the compiler error output for details.

Total time: 5 seconds
```

https://www.javatpoint.com/java-annotation

https://www.geeksforgeeks.org/annotations-in-java/