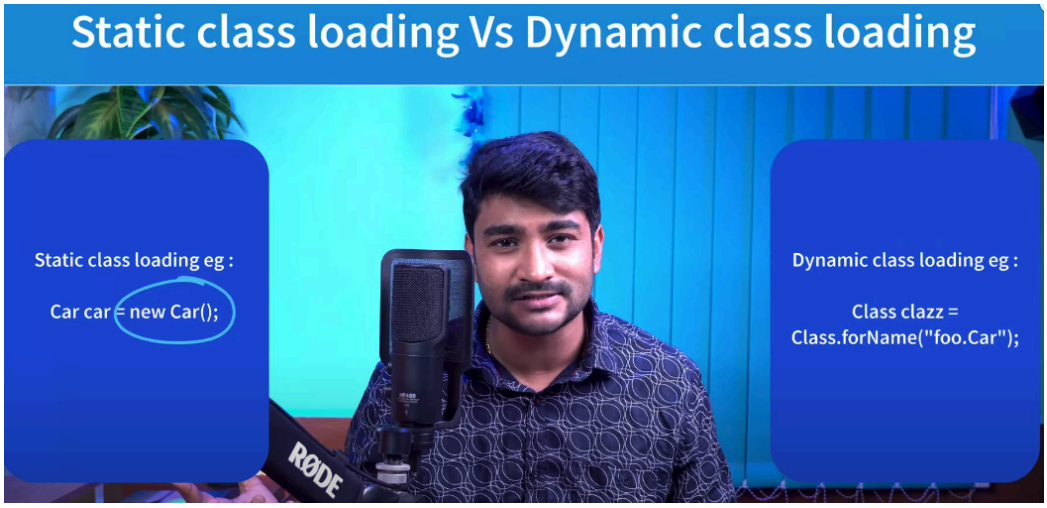
5 JAVA ERRORS you should learn today (NoClassDefFoundError) | JVM architecture - ClassLoader



NoClassDefFoundError is main villain. The other errors his right hands

Video Contents :

Static class Loading

Dynamic class loading

JVM architecture internals

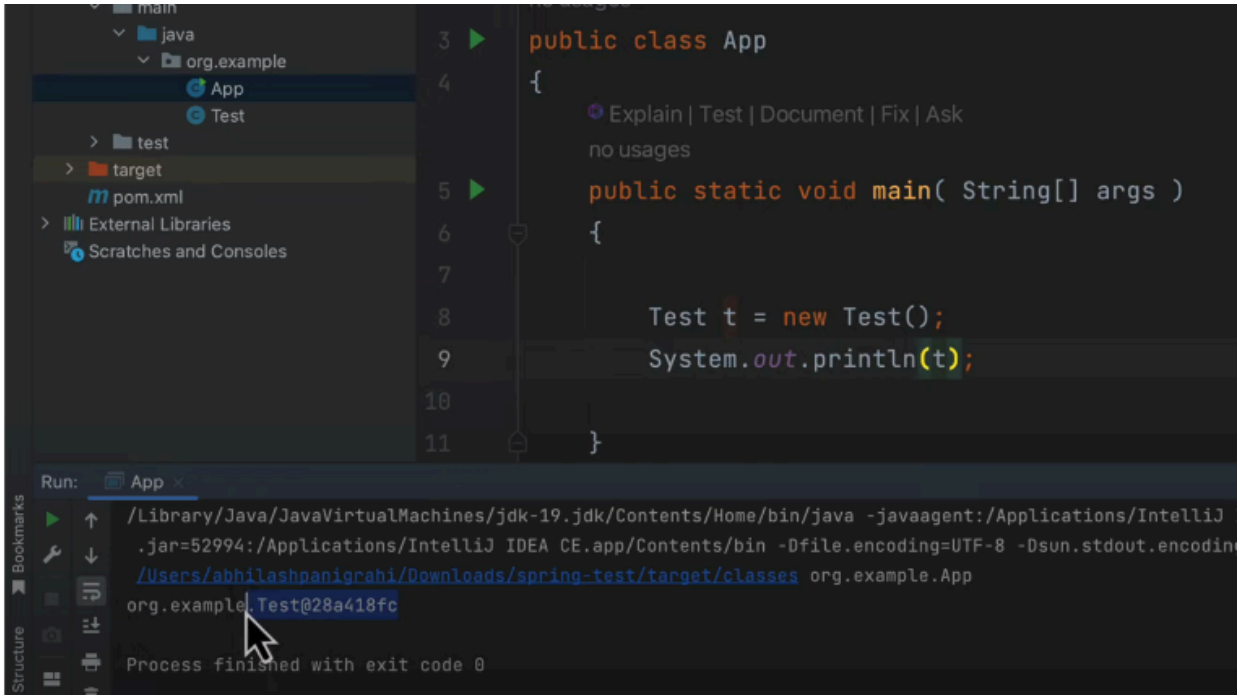
Class loader subsystems

NoClassDefFoundError

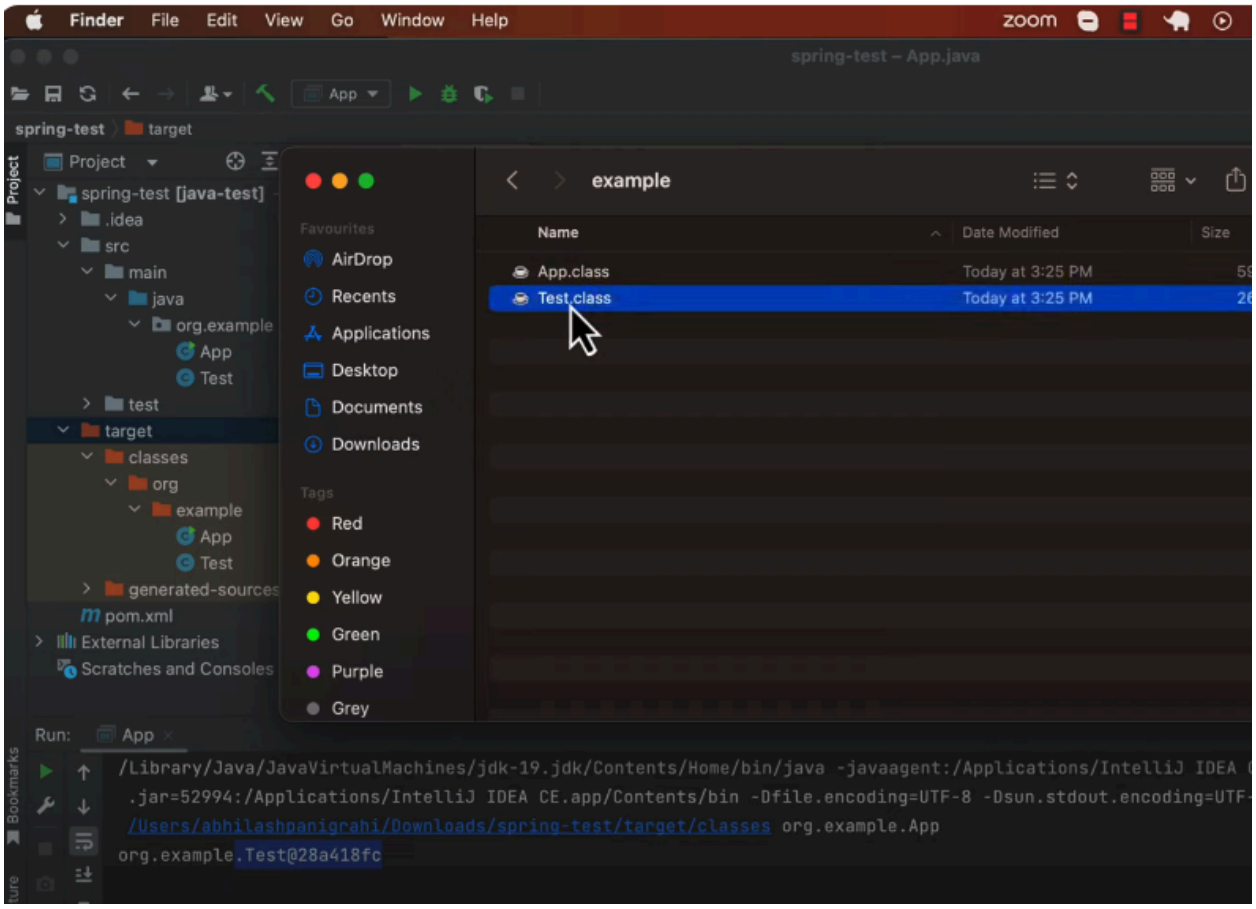
ExceptionInitializerError

Real-Time problem solving

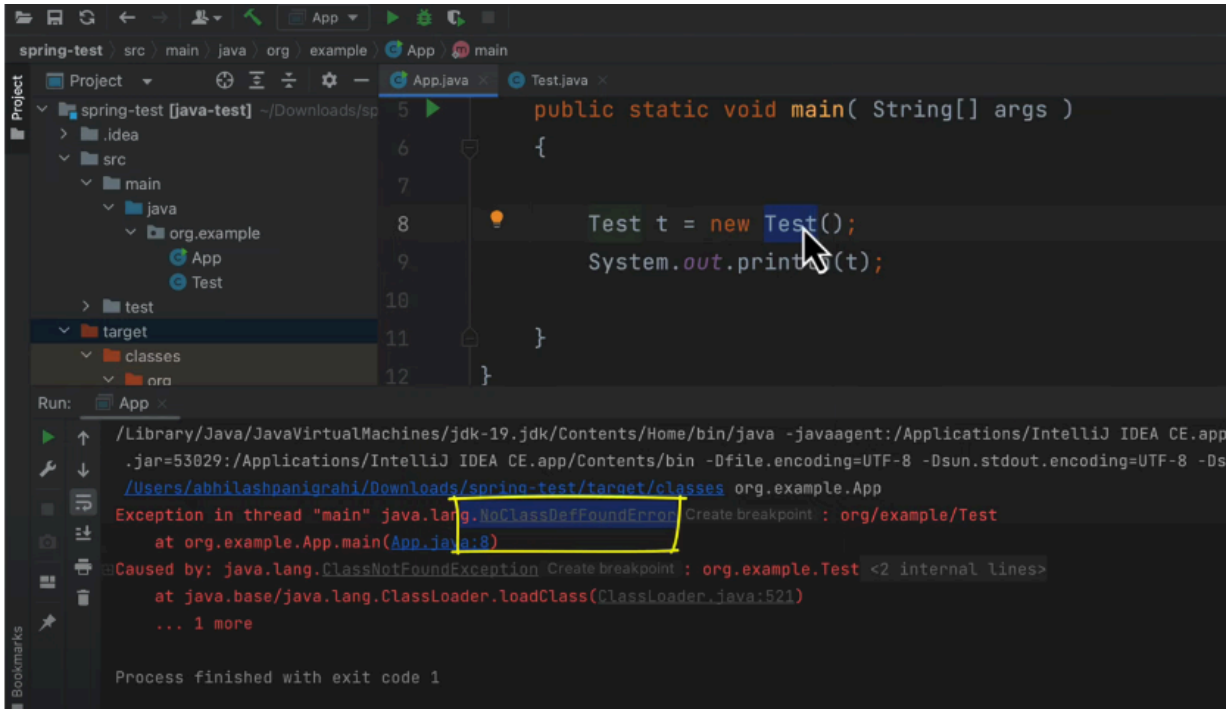
Just print test class object



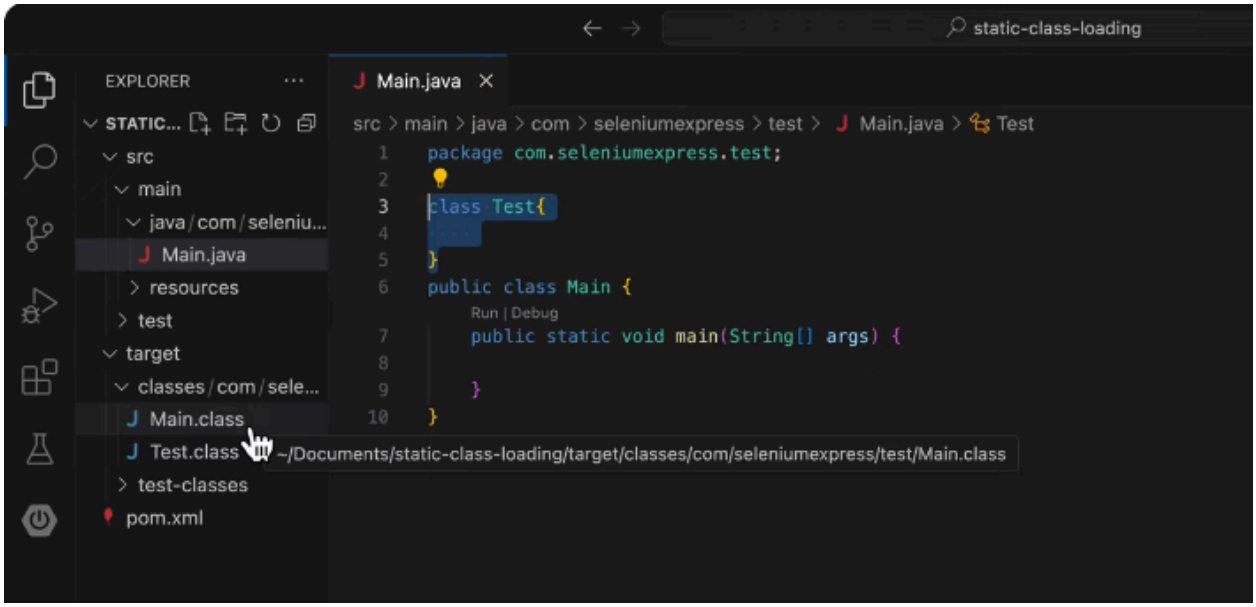
Just delete .class file from target folder and run



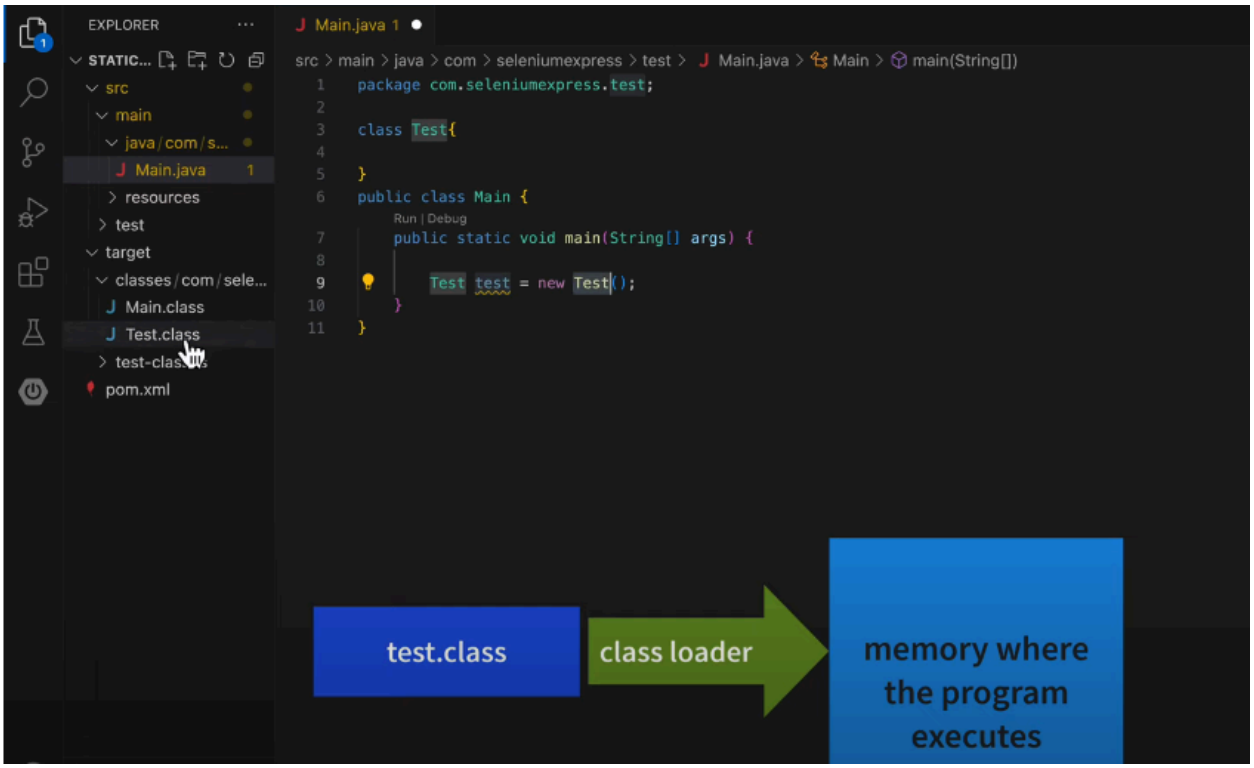
Now run the same code again. We get NoClassDefFoundError. If it is your understanding then you are wrong



The Foundation
Static Class Loading
[Inheritance, Association & errors]

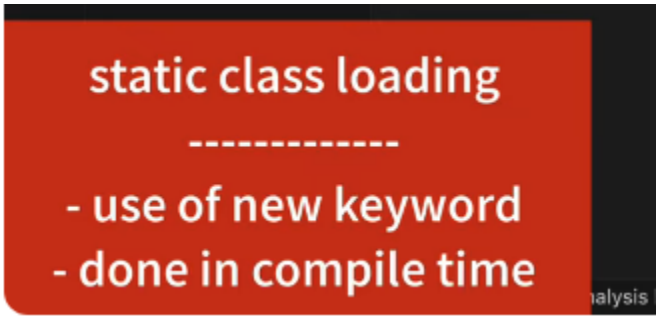


We have got two class files which contains bytecode and will load into memory with help of java class loader.

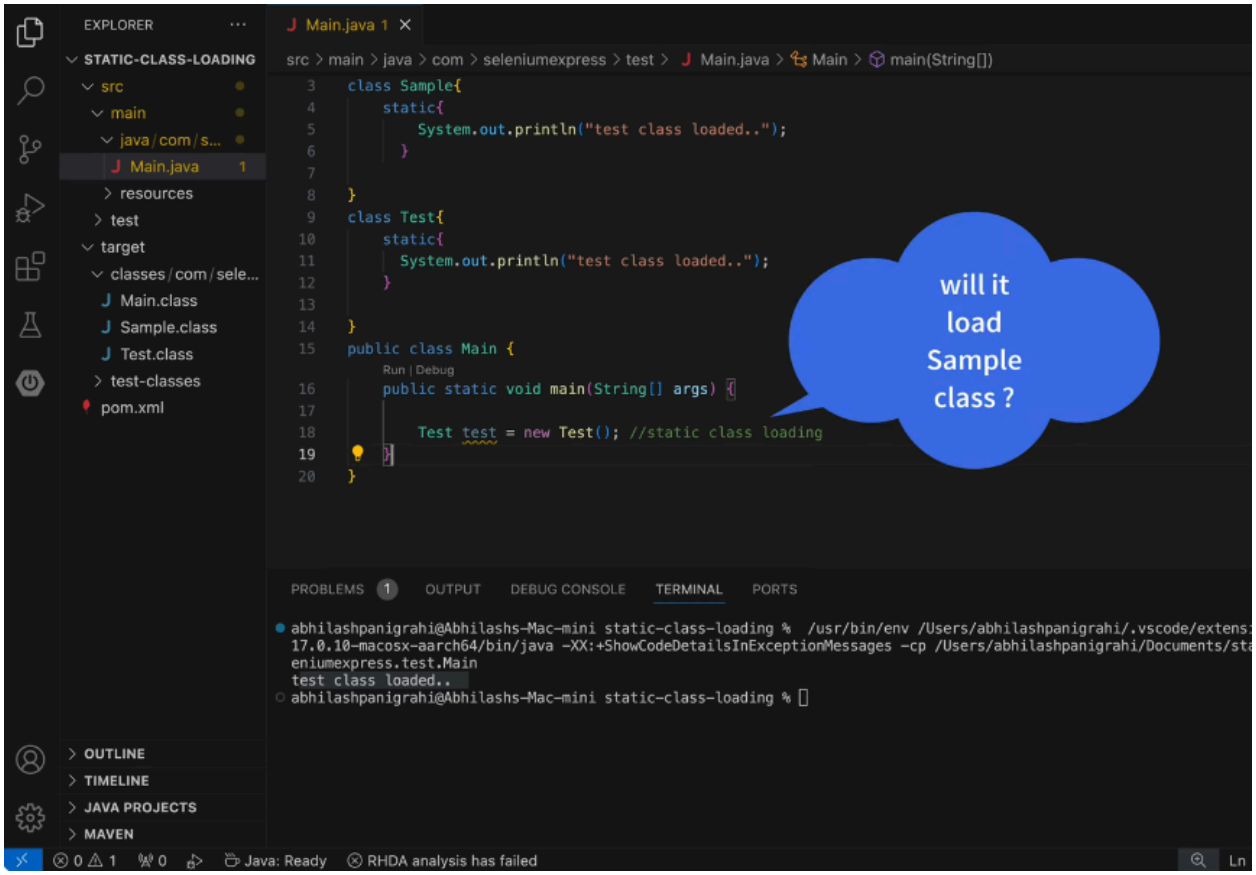


```
5   }
6   public class Main {
        Run | Debug
7       public static void main(String[] args) {
8
9           Test test = new Test(); //static class loading
10      }
11  }
```

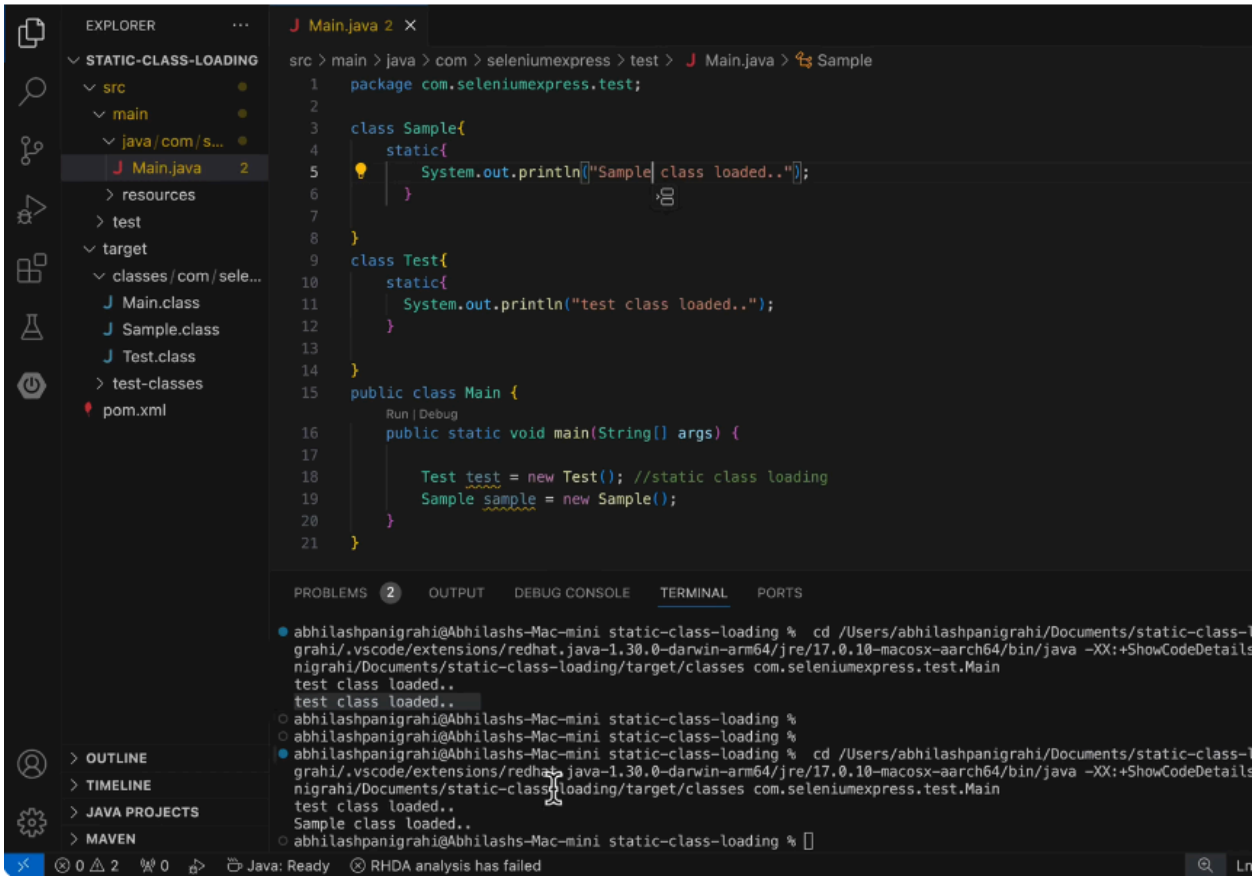
Test class will be loaded into memory when line 9 executes. From next time onward wen we refer the object it will be very fast.

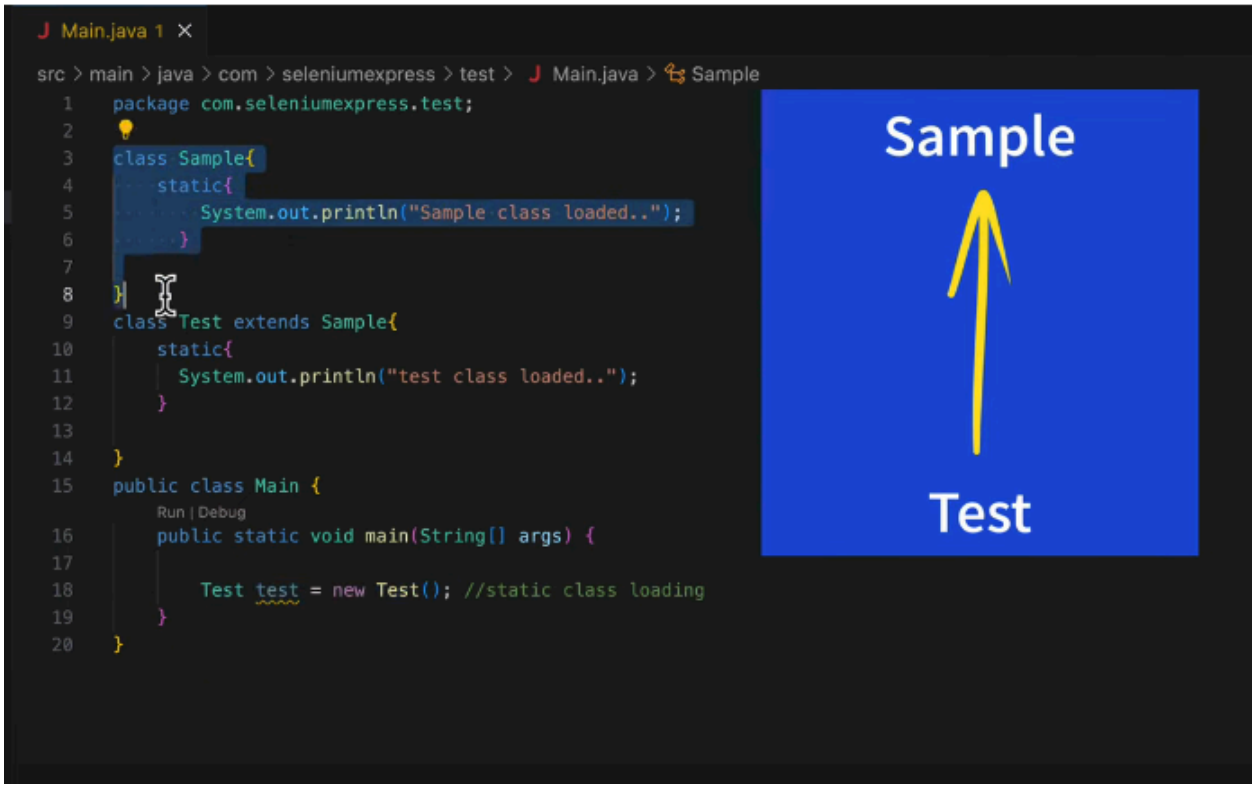


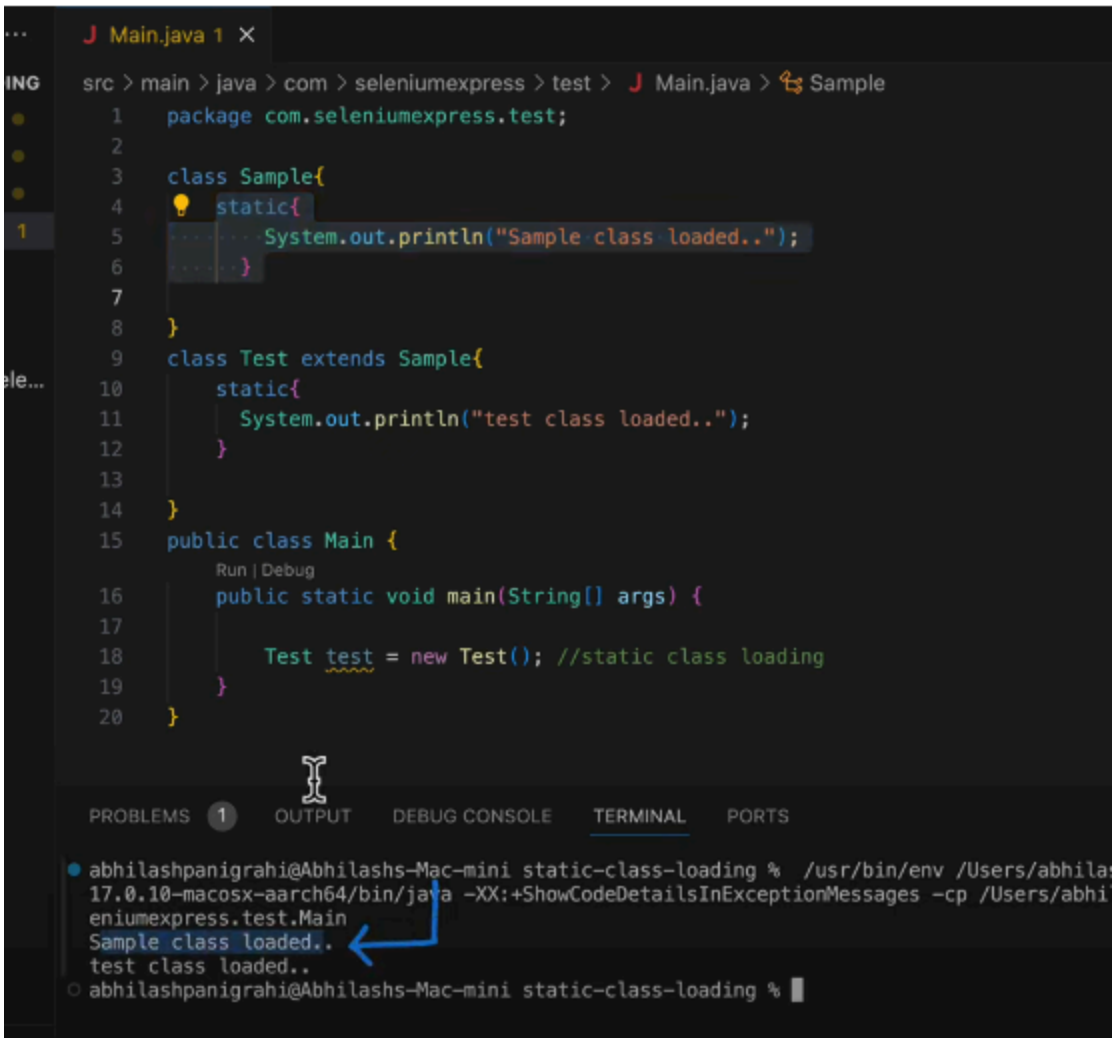
Sample class is not loaded even though we have a static block in it.



After adding a sample class into the main method then only it rams. Refer below screenshot

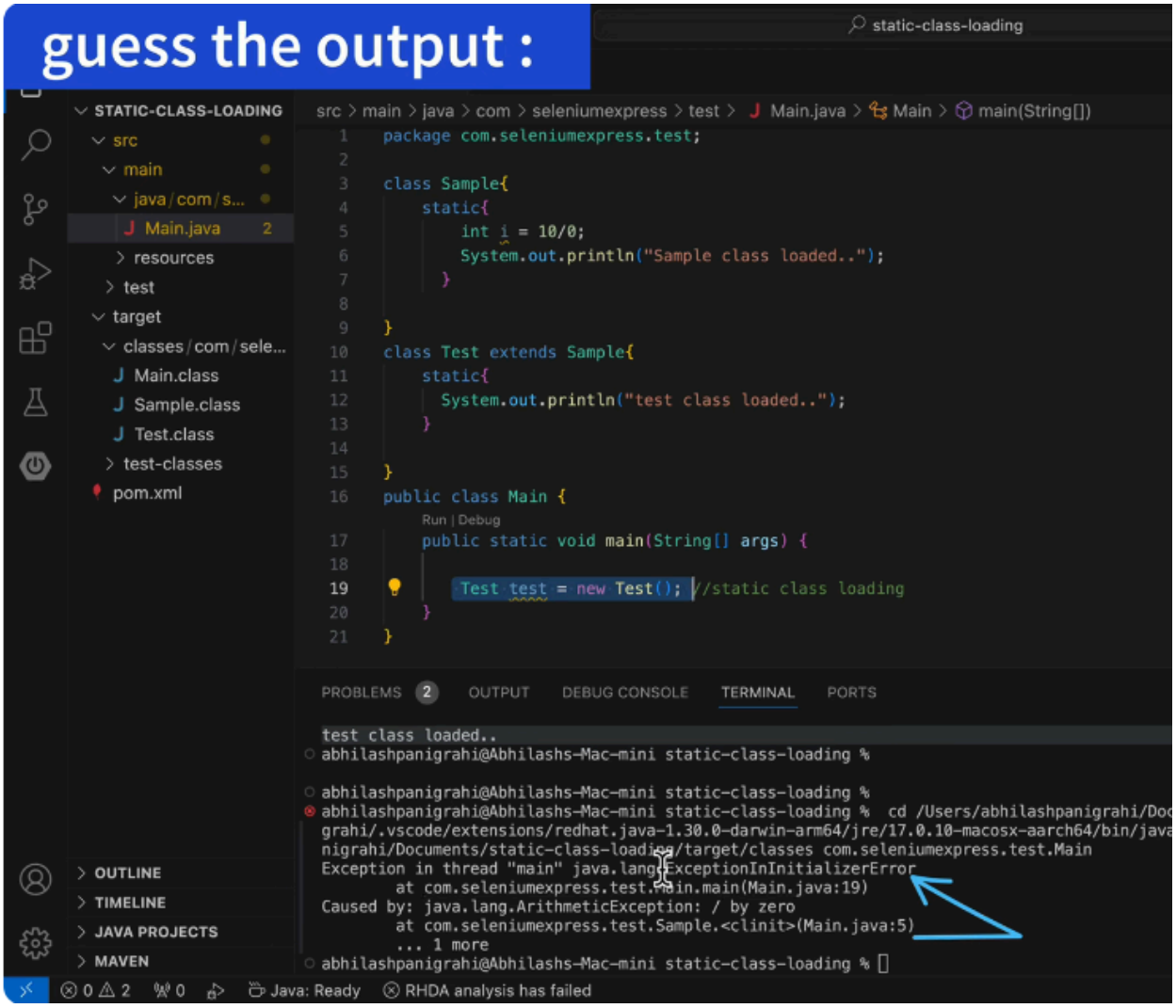

EXPLORER
STATIC-CLASS-LOADING
src
main
java/com/s...
Main.java 2
resources
test
target
classes/com/sele...
Main.class
Sample.class
Test.class
test-classes
pom.xml
OUTLINE
TIMELINE
JAVA PROJECTS
MAVEN
Main.java 2
src > main > java > com > seleniumexpress > test > Main.java > Sample
1 package com.seleniumexpress.test;
3 class Sample{
4 static{
5 System.out.println("Sample class loaded..");
6 }
8 }
9 class Test{
10 static{
11 System.out.println("test class loaded..");
12 }
14 }
15 public class Main {
Run | Debug
16 public static void main(String[] args) {
18 Test test = new Test(); //static class loading
19 Sample sample = new Sample();
20 }
21 }
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
abhilashpanigrahi@Abhilashs-Mac-mini static-class-loading % cd /Users/abhilashpanigrahi/Documents/static-class-
grahi/.vscode/extensions/redhat.java-1.30.0-darwin-arm64/jre/17.0.10-macosx-aarch64/bin/java -XX:+ShowCodeDetails
nigrahi/Documents/static-class-loading/target/classes com.seleniumexpress.test.Main
test class loaded..
test class loaded..
abhilashpanigrahi@Abhilashs-Mac-mini static-class-loading %
abhilashpanigrahi@Abhilashs-Mac-mini static-class-loading %
abhilashpanigrahi@Abhilashs-Mac-mini static-class-loading % cd /Users/abhilashpanigrahi/Documents/static-class-
grahi/.vscode/extensions/redhat.java-1.30.0-darwin-arm64/jre/17.0.10-macosx-aarch64/bin/java -XX:+ShowCodeDetails
nigrahi/Documents/static-class-loading/target/classes com.seleniumexpress.test.Main
test class loaded..
Sample class loaded..
abhilashpanigrahi@Abhilashs-Mac-mini static-class-loading %
0 2 0
Java: Ready
RHDA analysis has failed


Main.java 1
src > main > java > com > seleniumexpress > test > Main.java > Sample
1 package com.seleniumexpress.test;
2
3 class Sample{
4 static{
5 System.out.println("Sample class loaded..");
6 }
7
8 }
9 class Test extends Sample{
10 static{
11 System.out.println("test class loaded..");
12 }
13
14 }
15 public class Main {
Run | Debug
16 public static void main(String[] args) {
17
18 Test test = new Test(); //static class loading
19 }
20 }
Sample
Test

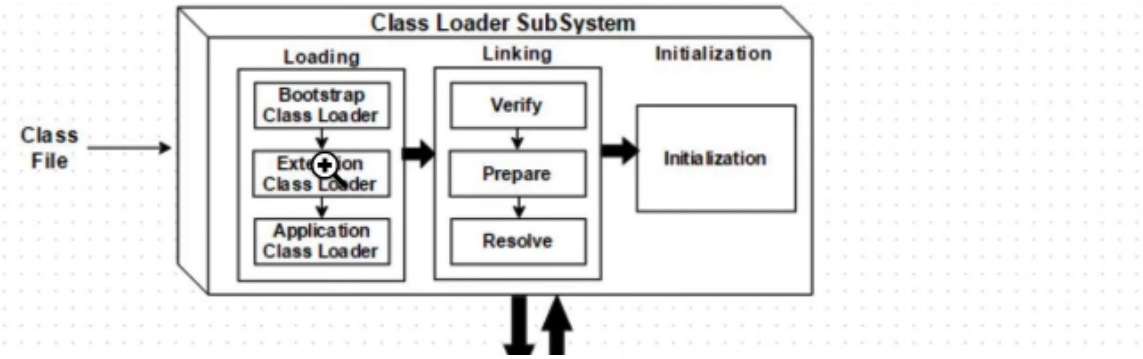


Whenever we have exception in super class static block we will get ExceptionInitializerError In line number 5 contains error.

ExceptionInitializerError error will come whenever static block fails.



How ClassLoader works ?
(JVM Internals)



ExceptionInitializerError error happens in the initialization phase in JVM.

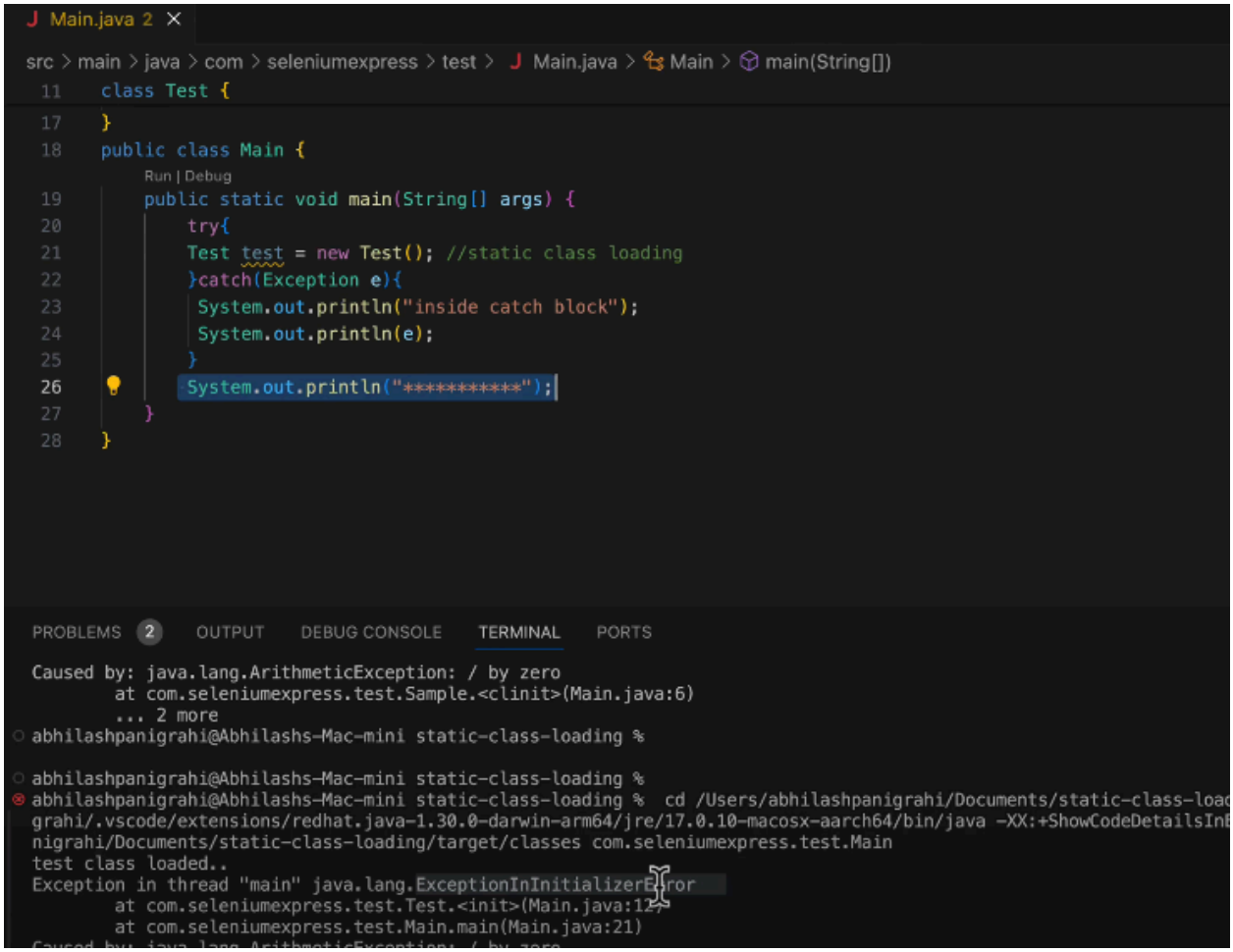
Loading stage : The application class loader will load user defined class files and BootStrap and Extension class loaders load system class and builtin class files.

In the linking stage: memory will be allocated.

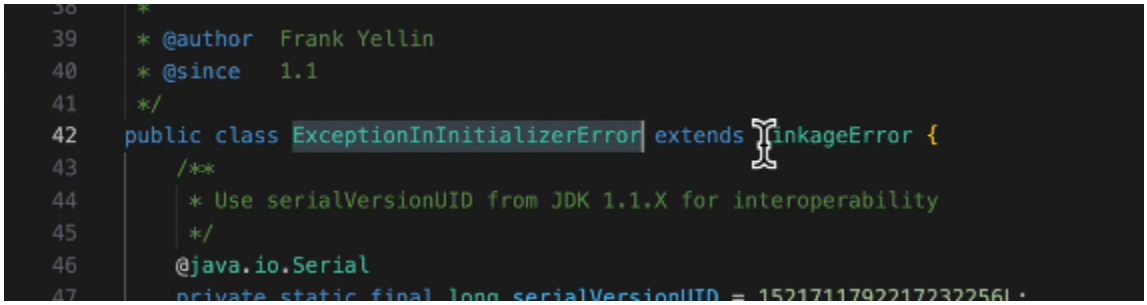
Initialization stage : static blocks and other object initialization will happen

Now create has a relationship between test and sample class as shown below.

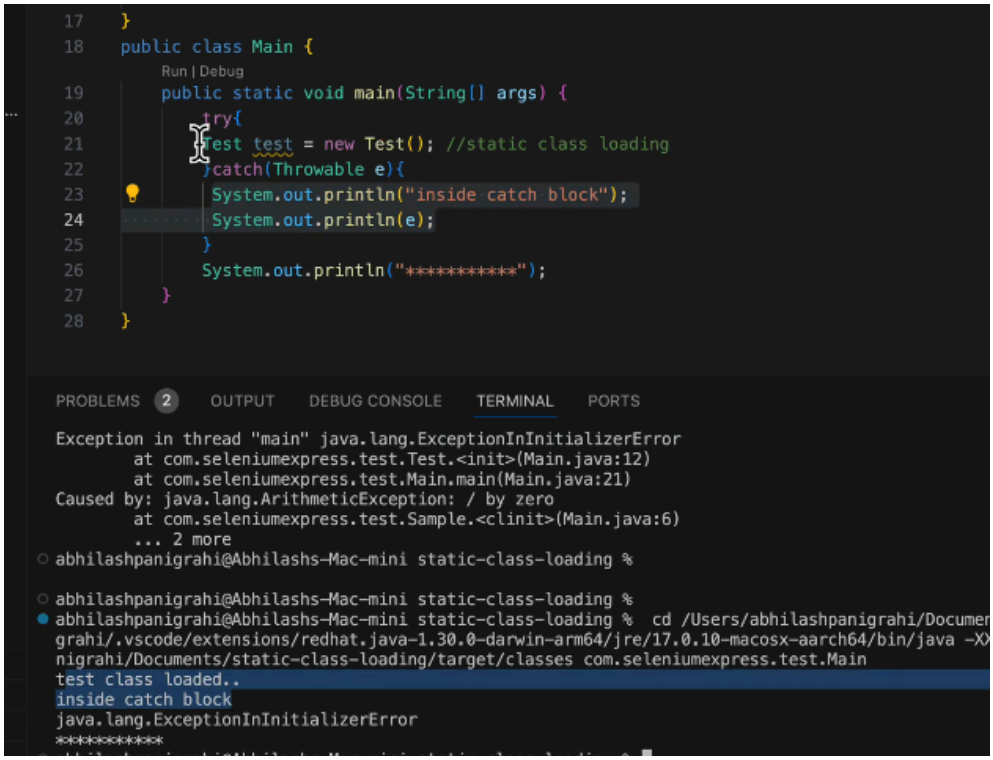
Refer below console where catch block is not printed. This is because we have runtime exception.



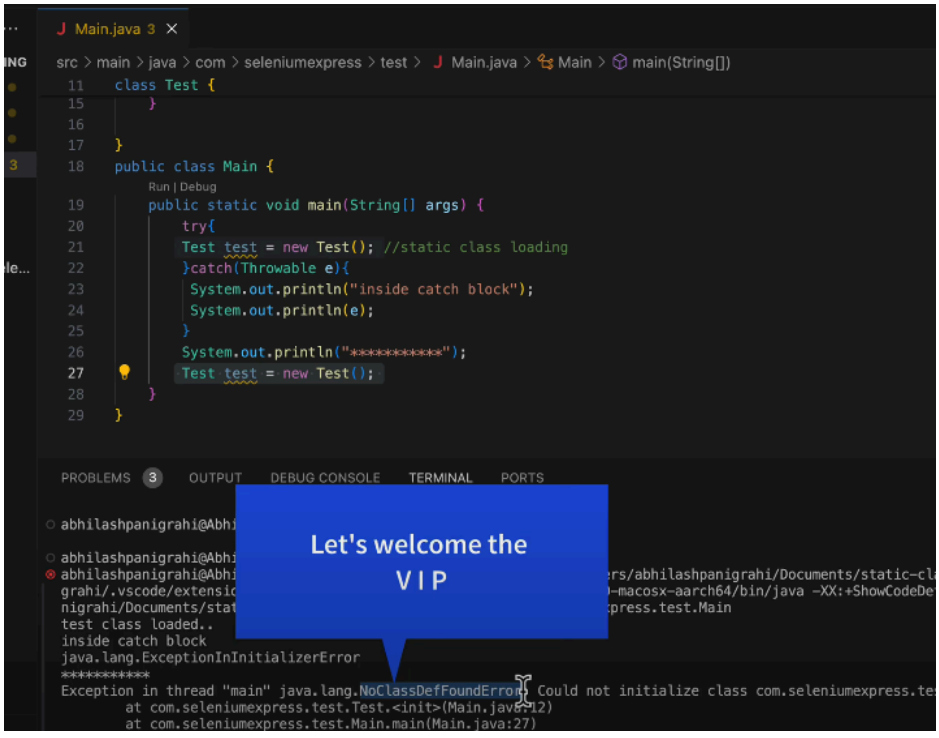
An error that happen in linking phase then it will come under linkageException



After changing throwable it is printed catch block.



Now create one more test object after catch block. We will get NoClassDefFoundError.



When it was previously tried we got an exception. So JVM imagine now also we will get this error so through NoClassDefFoundError

If you change .class file (delete some characters) we will get ClassFormatError

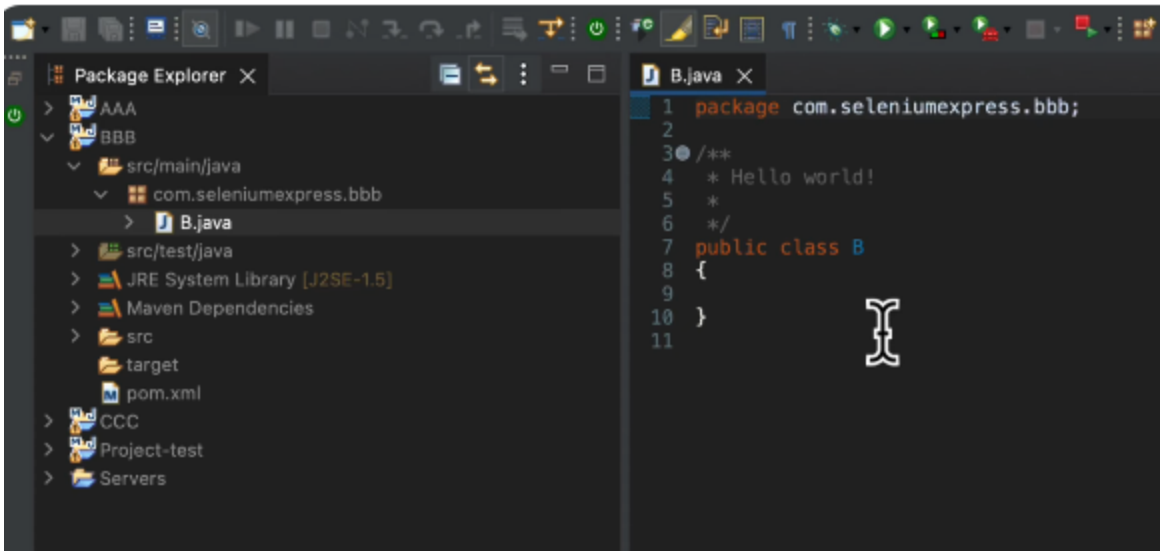
```
grahi/.vscode/extensions/redhat.java-1.30.0-darwin-arm64/jre/17.0.10-macosx-aarch64/bin/java -XX:+Show
nigrahi/Documents/static-class-loading/target/classes com.seleniumexpress.test.Main
inside catch block
java.lang.ClassFormatError: Illegal exception table range in class file com/seleniumexpress/test/Test
**********
```

It is linked to LinkageError

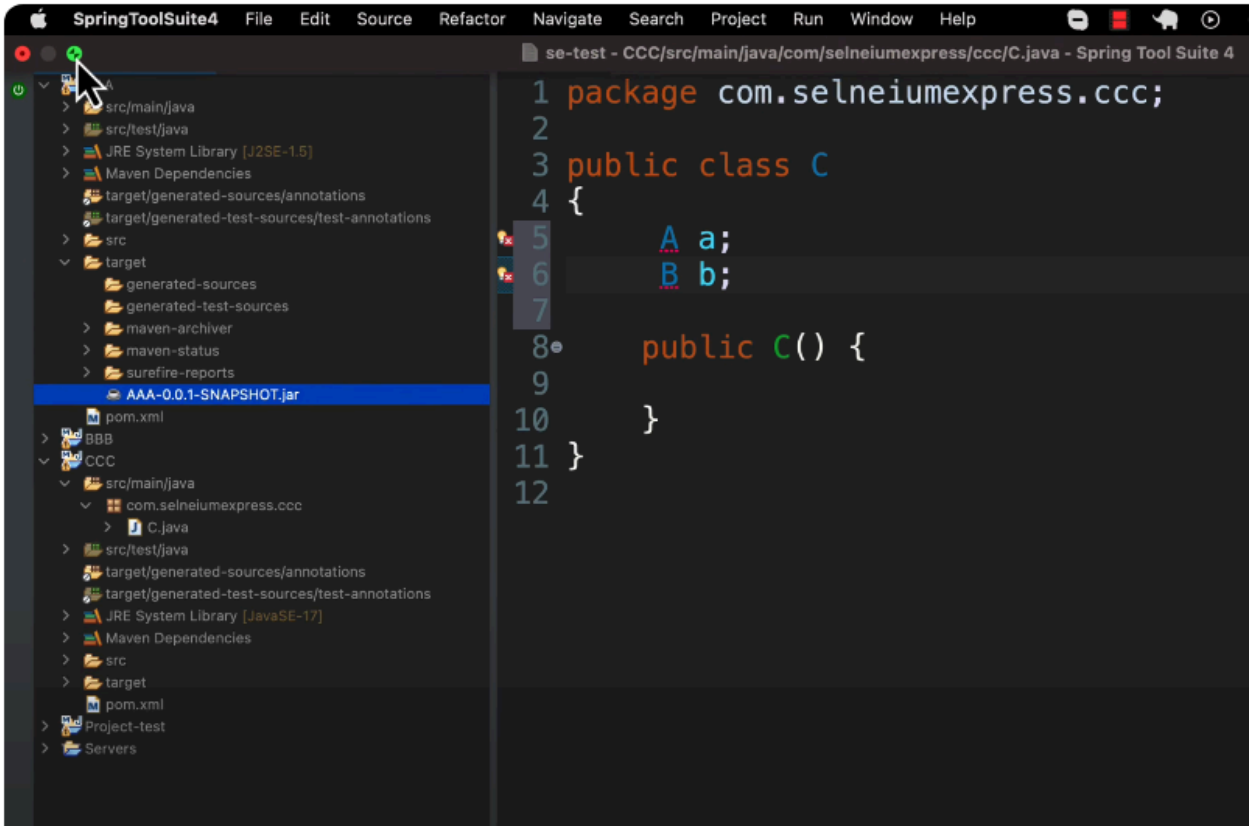
```
*/
public class ClassFormatError extends LinkageError {
    @java.io.Serial
    private static final long serialVersionUID = -8420114879011949195L;
```

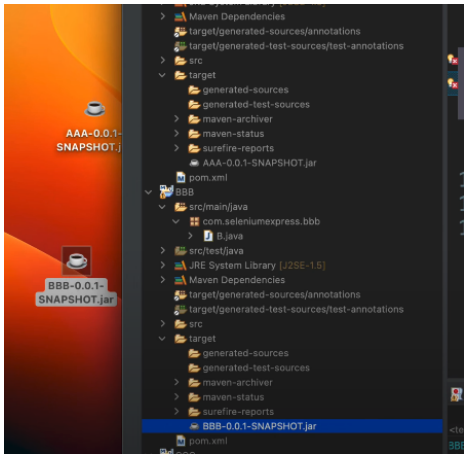
The below concept is missing dependencies

We have three projects with single class like below

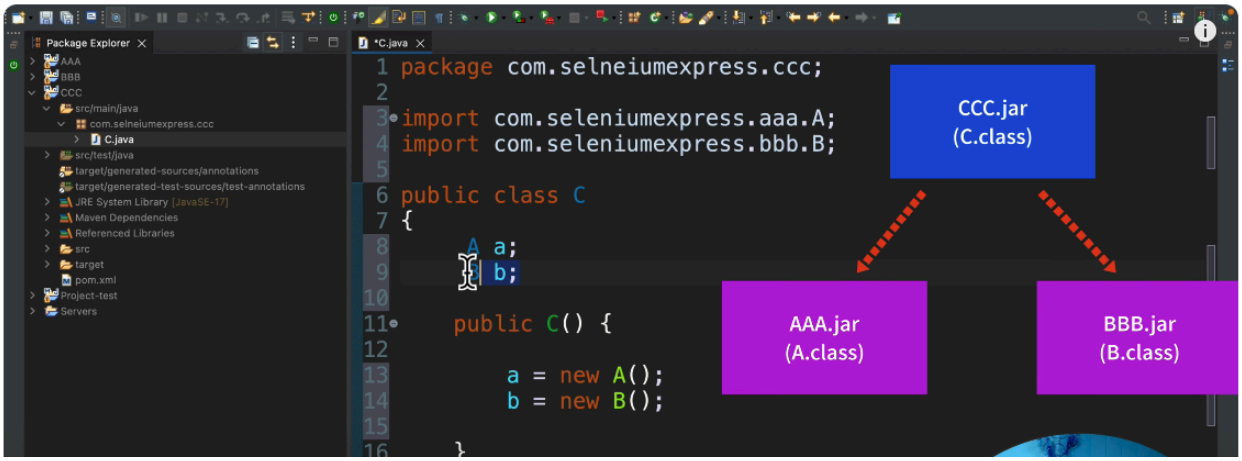


Build jars and include in project CCC

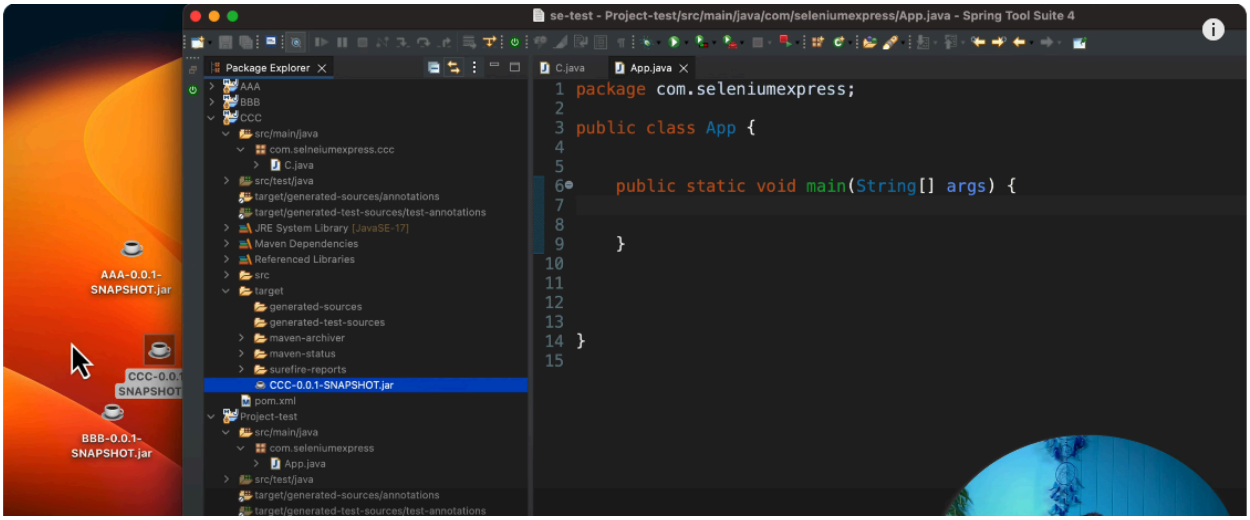




Include them as external jar file in classpath

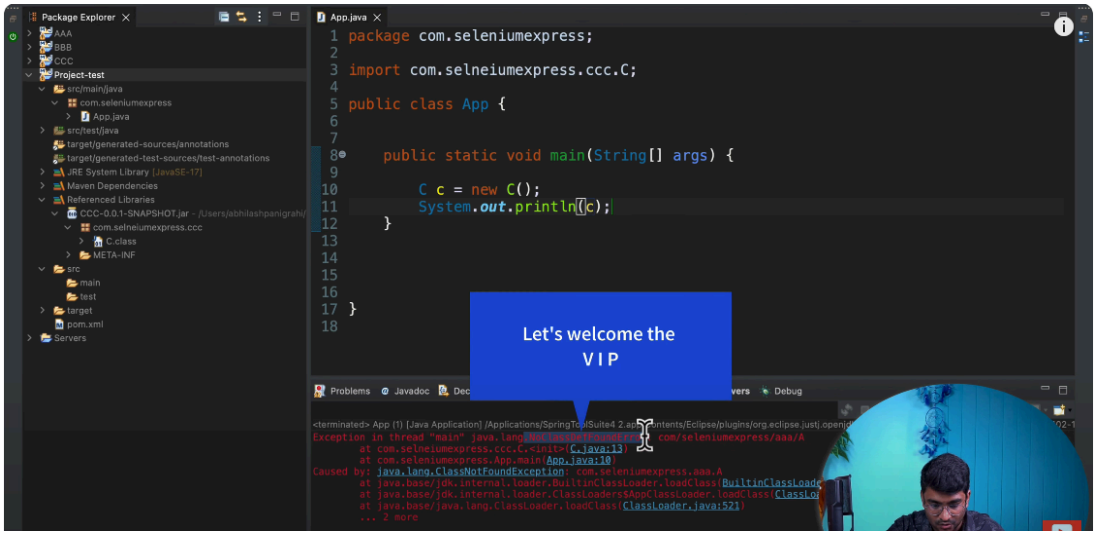


Now create jar file project ccc

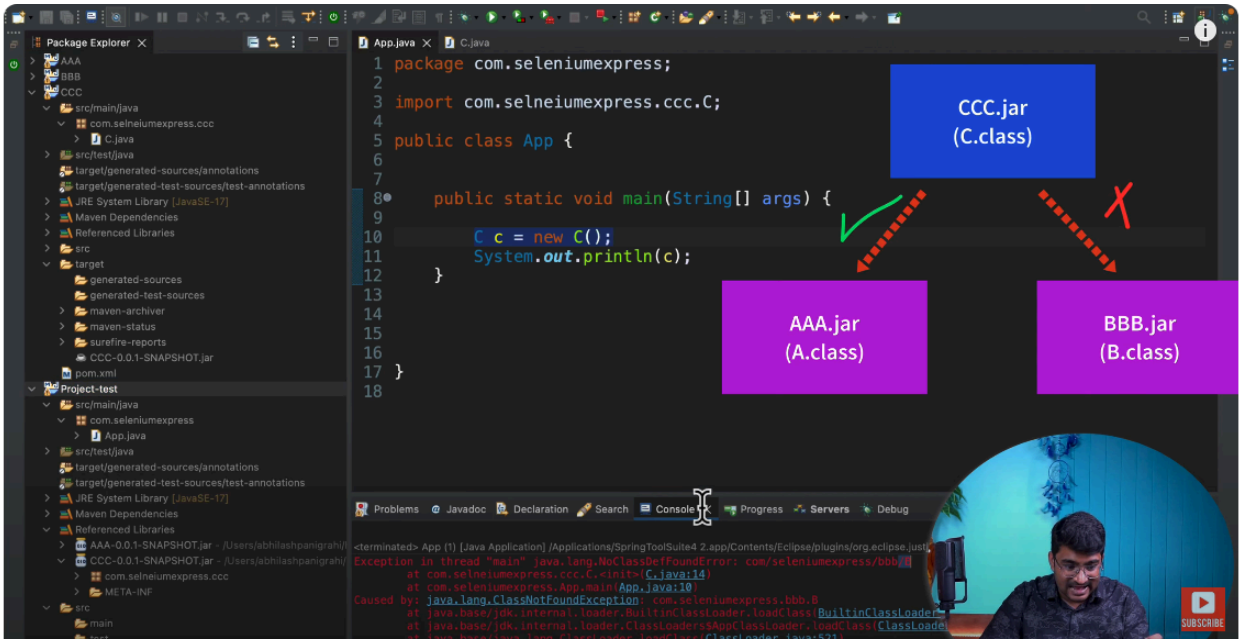


Add it in new project

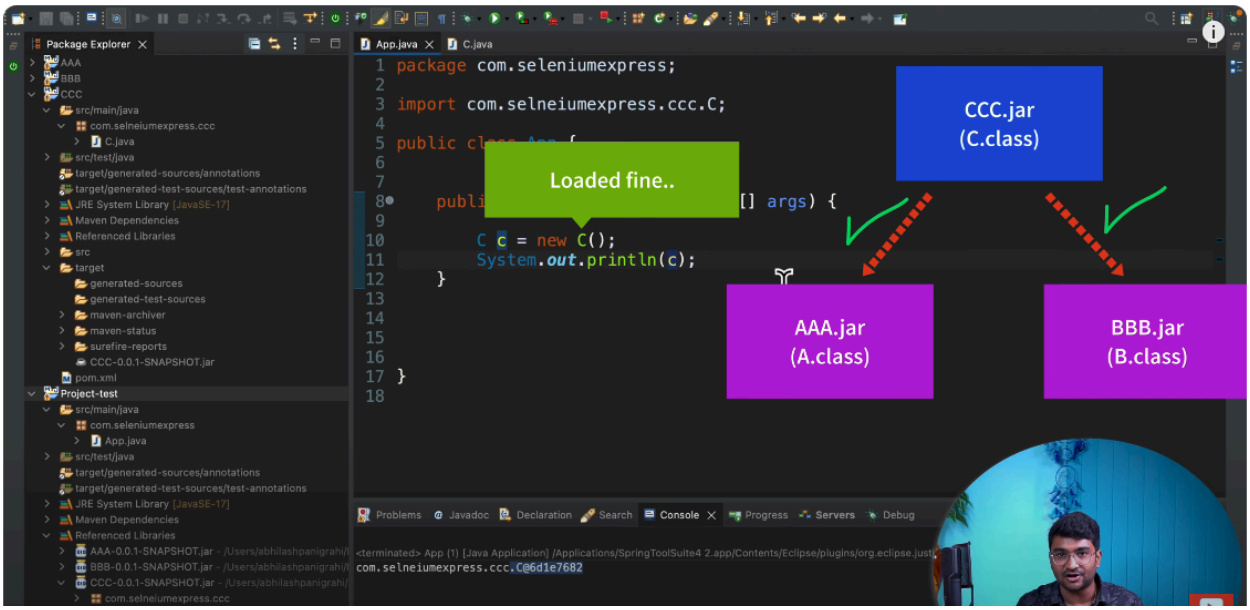
Now we got class not found exception



Now add project AAA.jar as it is showing noClassDefFoundError for A and now we got same error for B as well.



After adding bbb.jar it ran well.

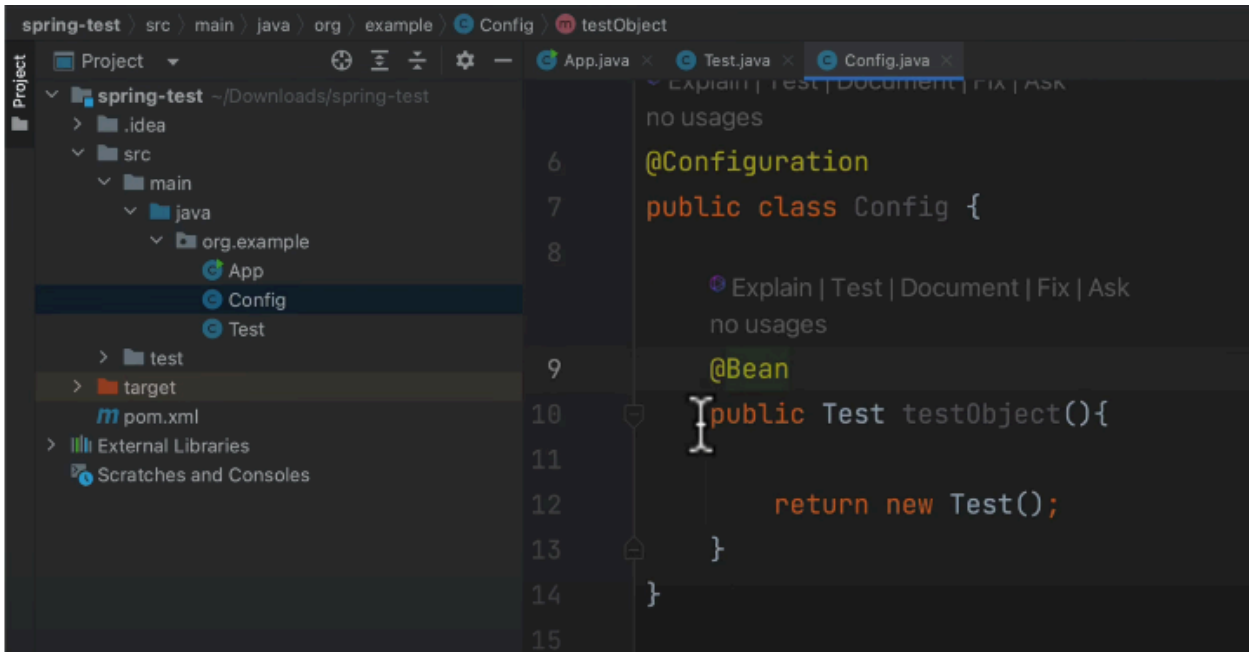


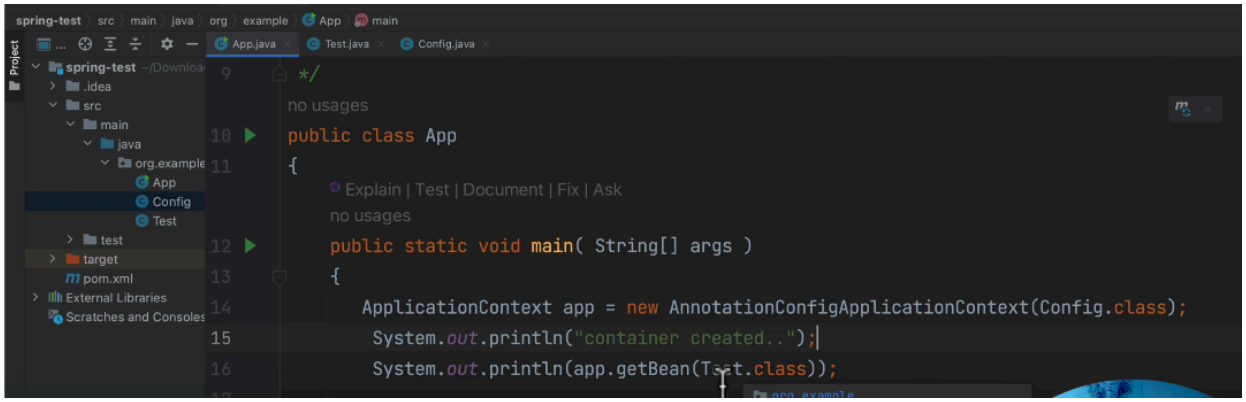
Advanced Framework Issues (demo using spring framework)

The below concept is also called - the jar conflict

Most occurrences in our day to day. Wasting lot of time without knowing something wrong with our dependencies

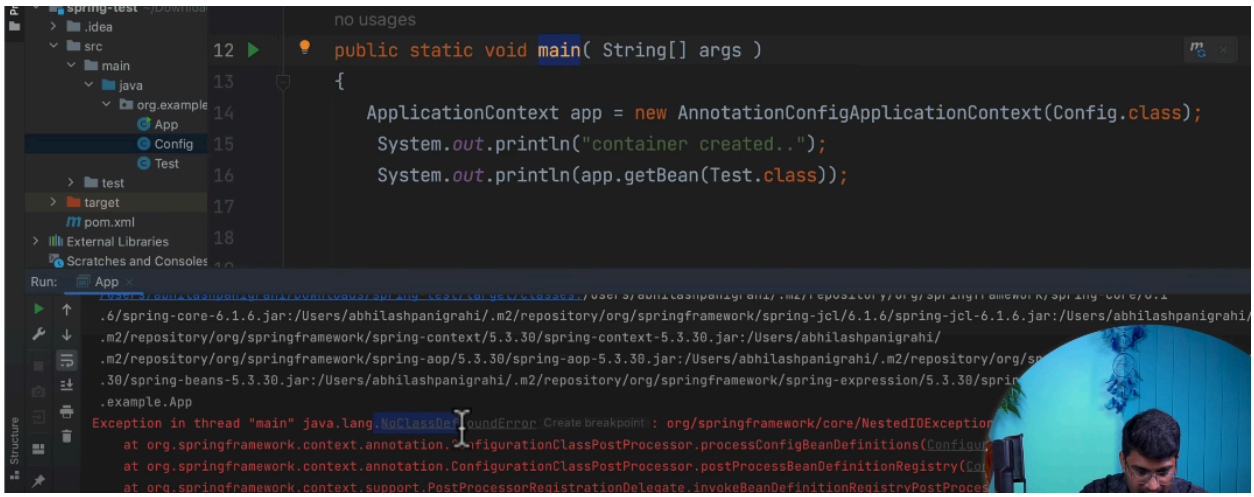
Assume that we have below spring boot application





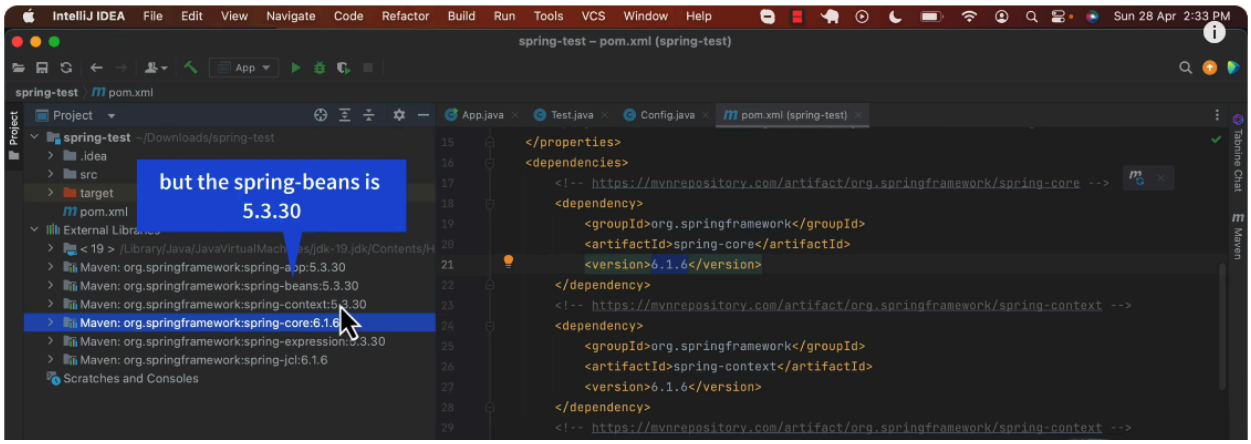
All we want to print the bean

When we run we get NoClassDefFoundError

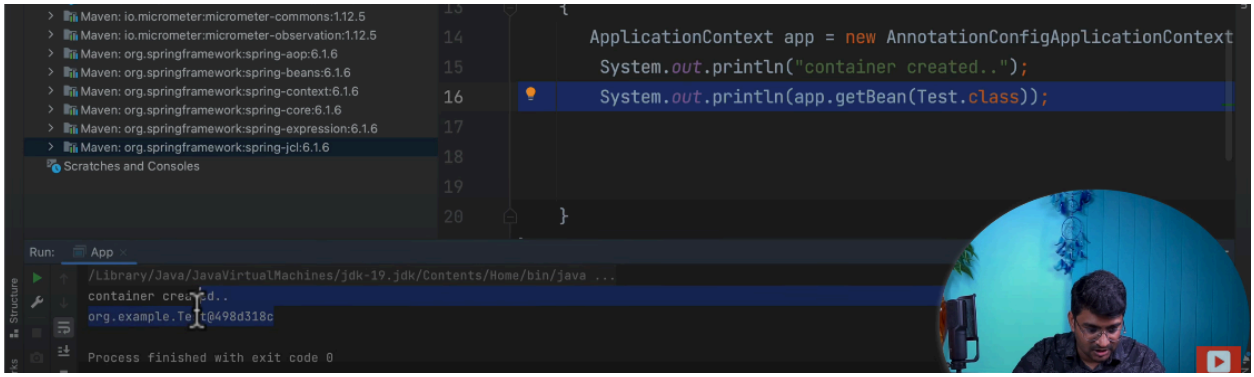


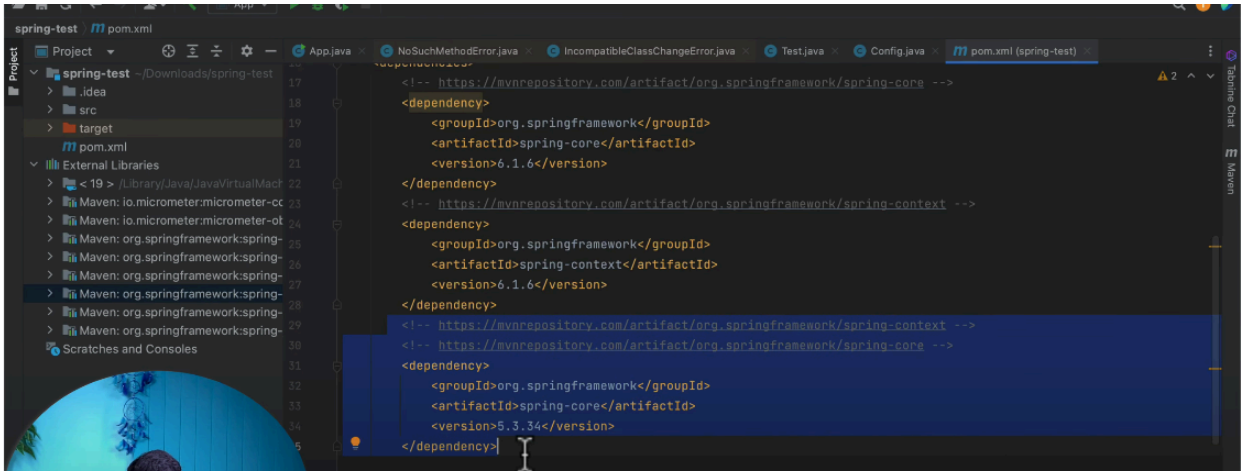
Caused by ClassNotFoundError

Why are we getting this error when the code is clean?
Because it has some mismatched versions of jar. Someone did some changes build.gradle / pom.xml

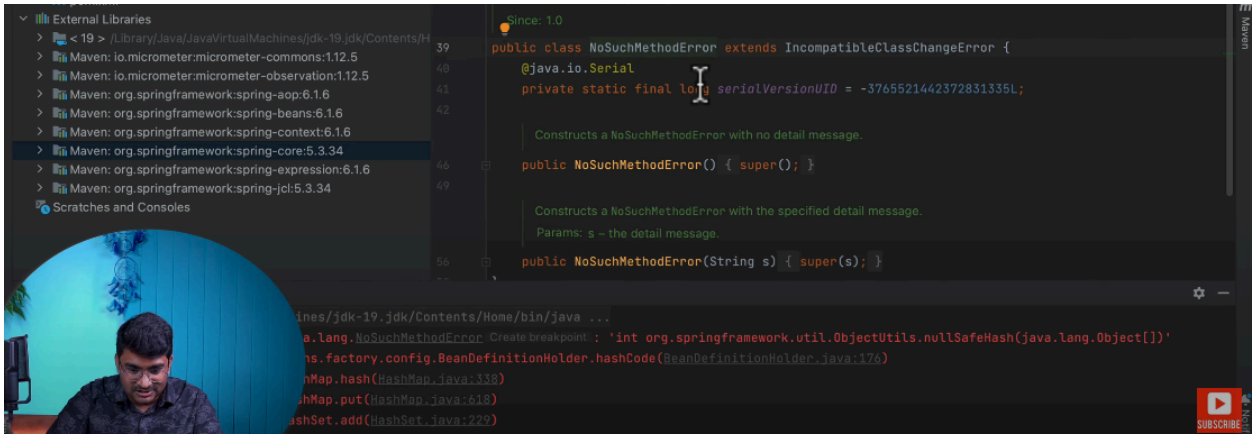


Clean and rebuild the project and run the project you will get our print statement

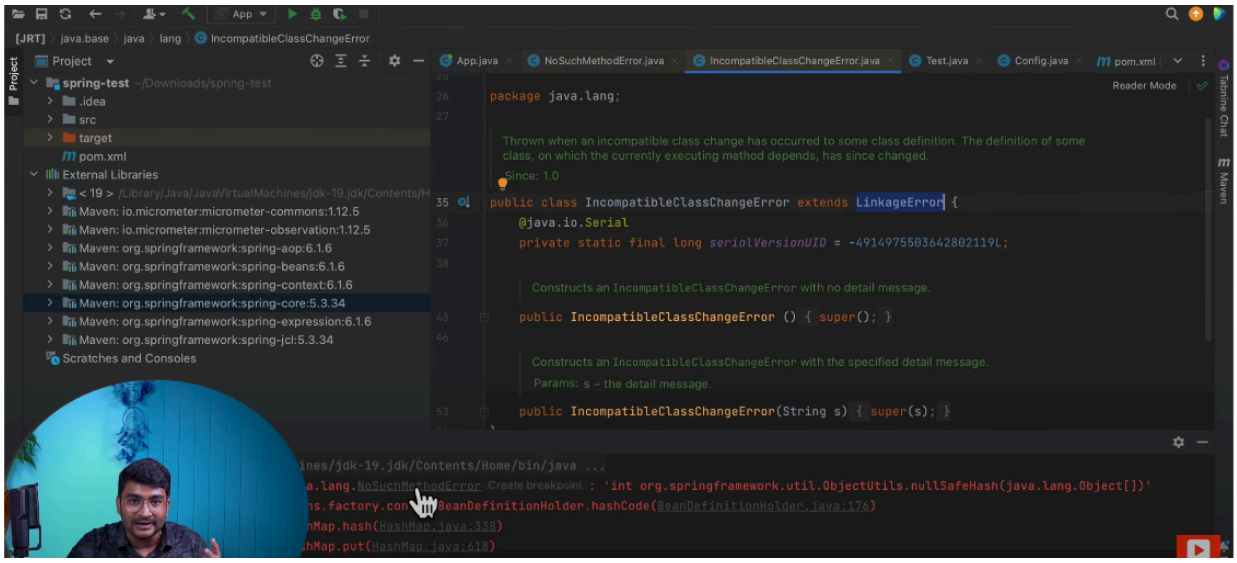




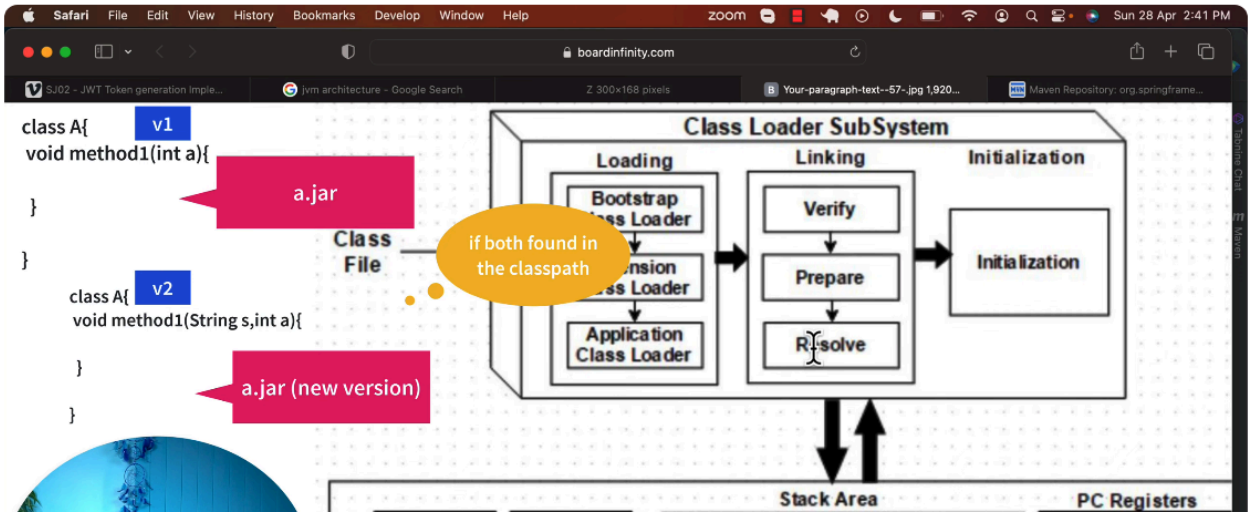
When we have different versions of spring jars we will get errors like NoSuchMethodError.

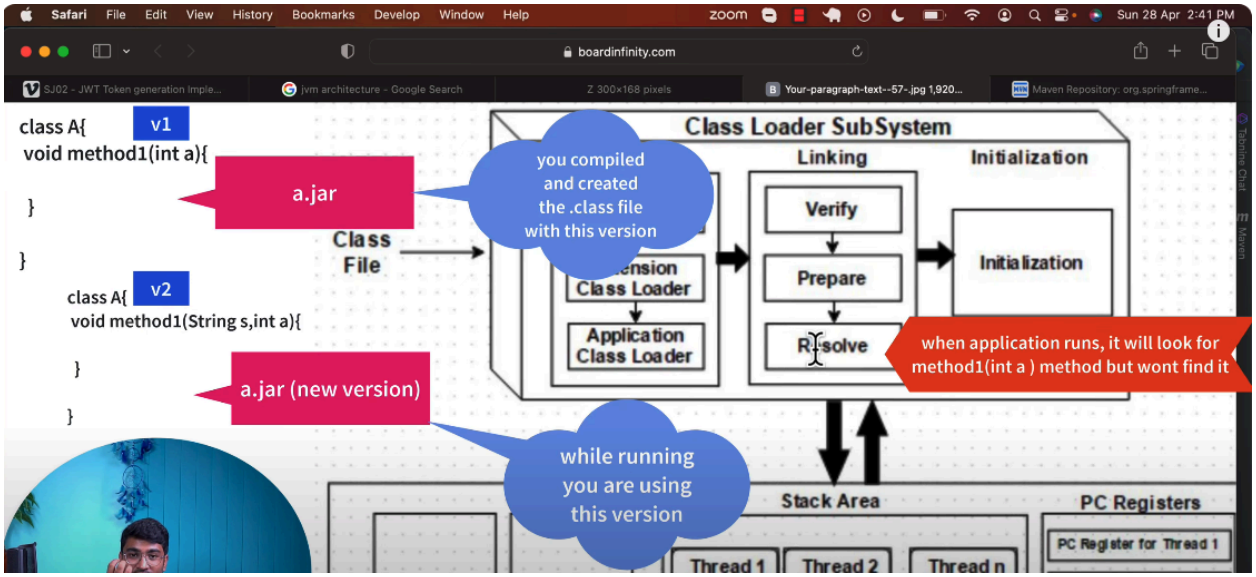


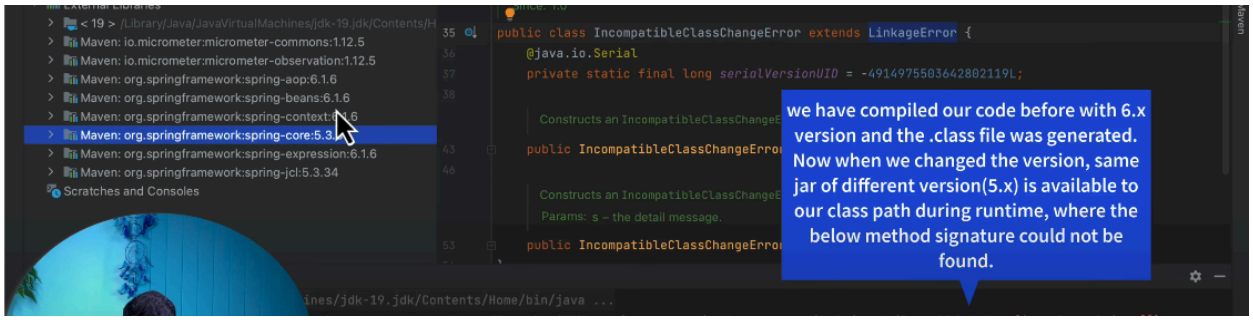
NoSuchMethodError will come as part of Linkage Error.

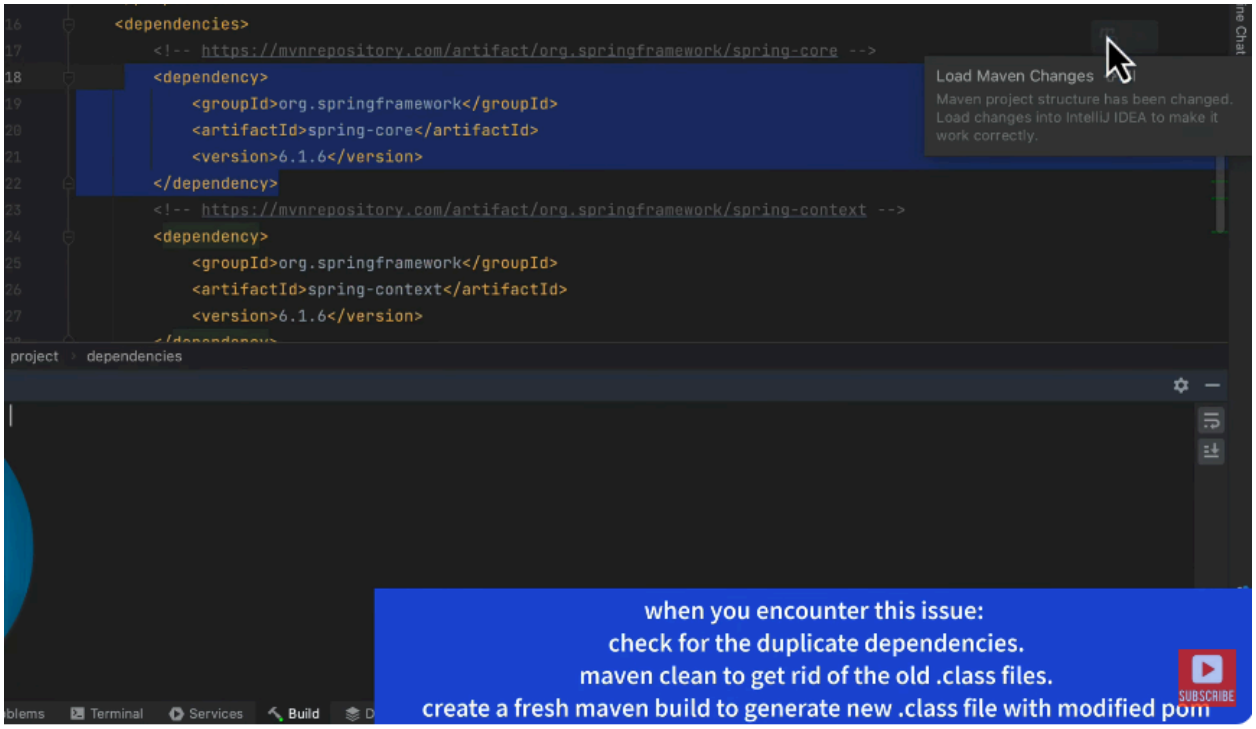


Because we have two versions
During compile time it take one version and During Runtime it takes others version
Methods argument varies so in JVM during it is not able to Resolve in finding correct method


boardinfinity.com
class A{
void method1(int a){
}
}
v1
a.jar
Class File
if both found in the classpath
class A{
void method1(String s,int a){
}
}
v2
a.jar (new version)
Class Loader SubSystem
Loading
Linking
Initialization
Bootstrap Class Loader
Application Class Loader
Verify
Prepare
Resolve
Initialization
Stack Area
PC Registers


boardinfinity.com
class A{
void method1(int a){
}
}
v1
a.jar
you compiled and created the .class file with this version
Class File
class A{
void method1(String s,int a){
}
}
v2
a.jar (new version)
while running you are using this version
Class Loader SubSystem
Linking
Initialization
Application Class Loader
Verify
Prepare
Initialization
when application runs, it will look for method1(int a) method but wont find it
Stack Area
PC Registers
PC Register for Thread 1
Thread 1
Thread 2
Thread n


Maven: io.micrometer:micrometer-commons:1.12.5
Maven: io.micrometer:micrometer-observation:1.12.5
Maven: org.springframework:spring-aop:6.1.6
Maven: org.springframework:spring-beans:6.1.6
Maven: org.springframework:spring-core:5.3.34
Maven: org.springframework:spring-expression:6.1.6
Maven: org.springframework:spring-jcl:5.3.34
Scratches and Consoles
public class IncompatibleClassChangeError extends LinkageError {
@java.io.Serial
private static final long serialVersionUID = -4914975503642802119L;
we have compiled our code before with 6.x version and the .class file was generated. Now when we changed the version, same jar of different version(5.x) is available to our class path during runtime, where the below method signature could not be found.

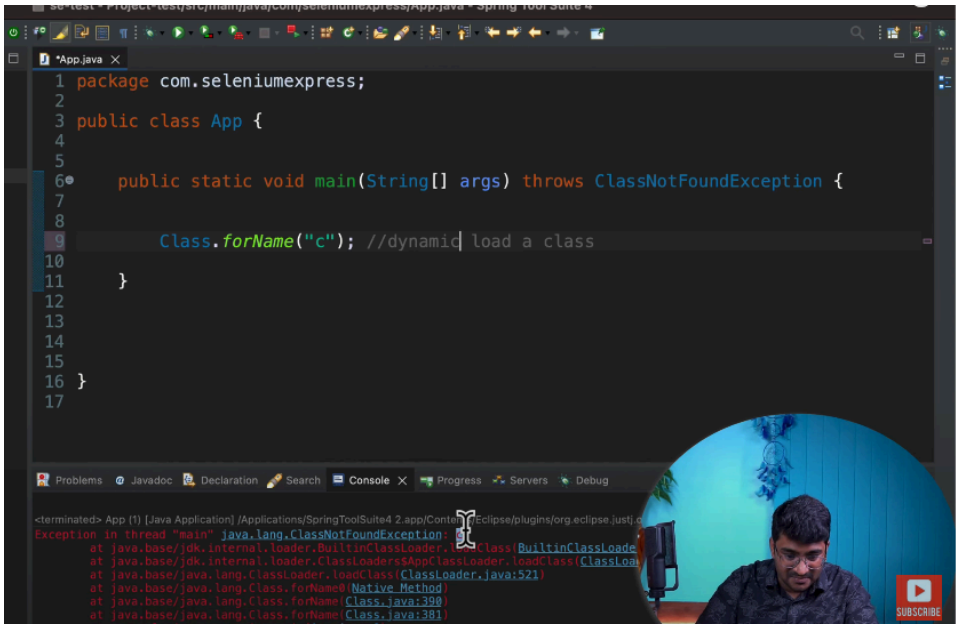


When ever we want to use same class object again and again
Static class loading happens during compile time. Whatever we discussed so far is static class loading.

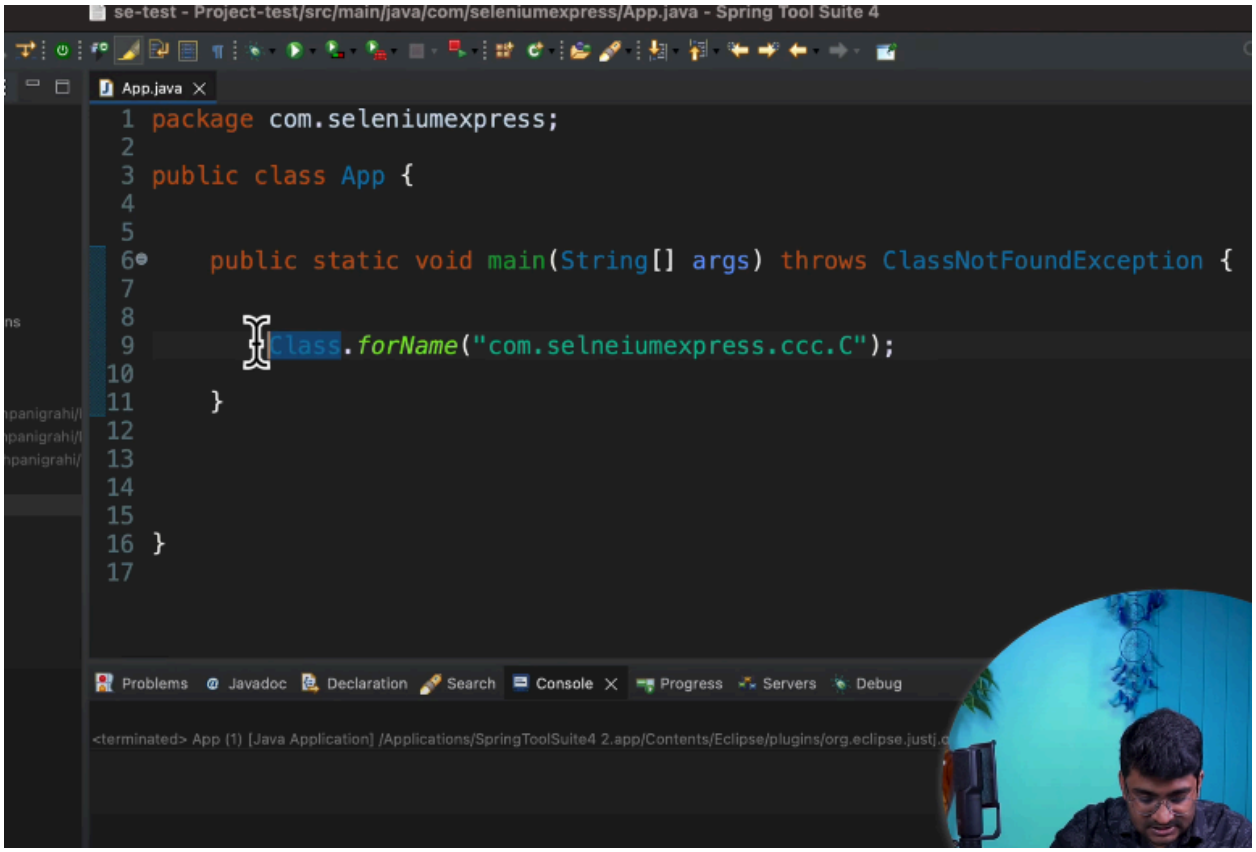
Dynamic class loading happens due to a particular method

ClassNotFoundException happens during Dynamic class loading
ClassDefNotFoundException happens during Static class loading

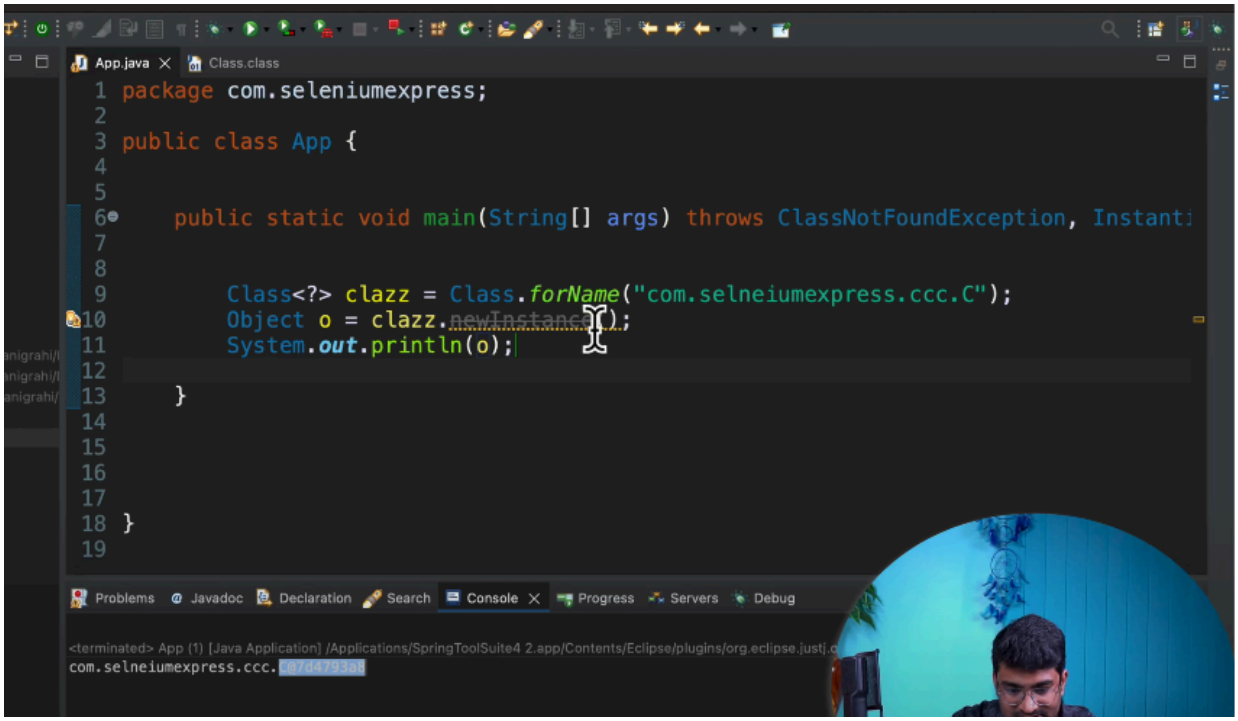
ClassNotFoundException throws ClassNotFoundException so we have to add to method signature.



In order to avoid above exception we have to load complete path.

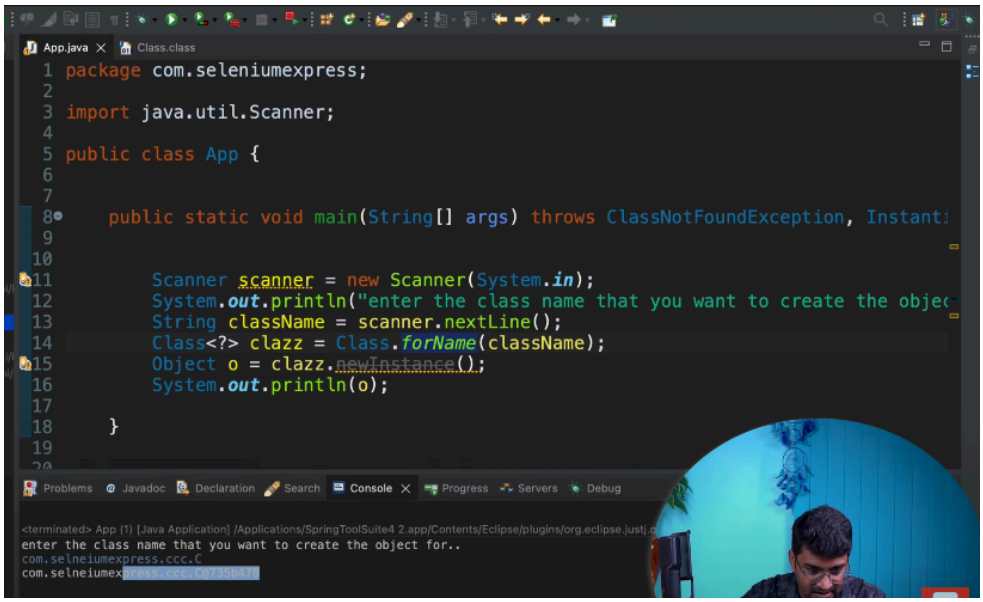


Another way of dynamic loading using class Class.

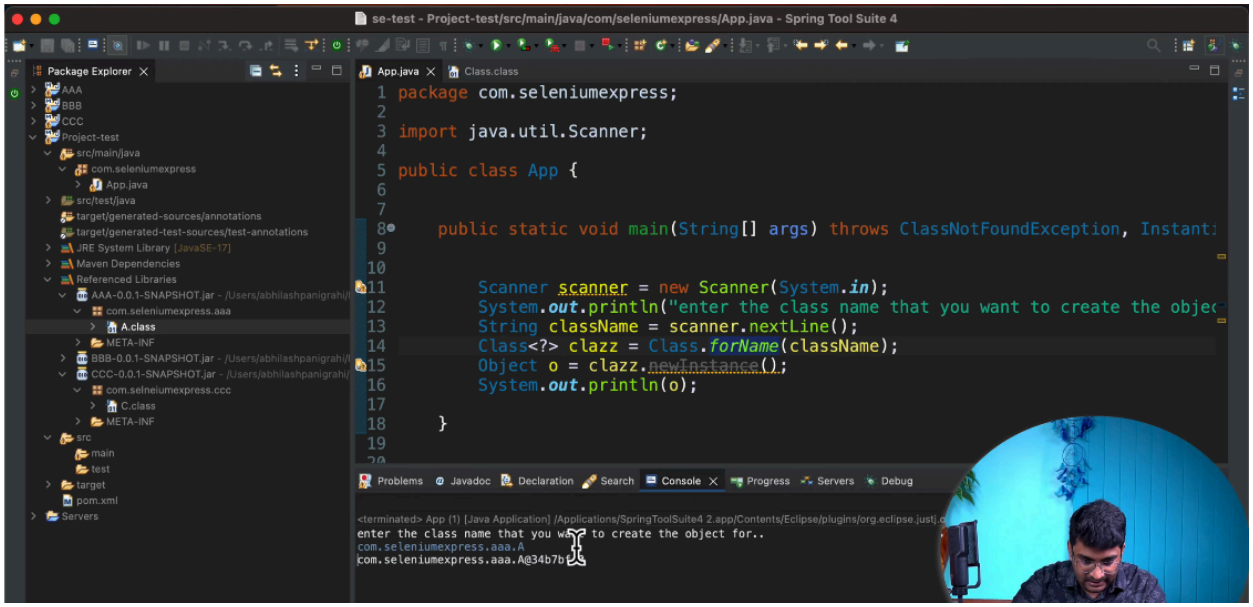


When it is useful
When we have a lot of classes. And we want to create object dynamically

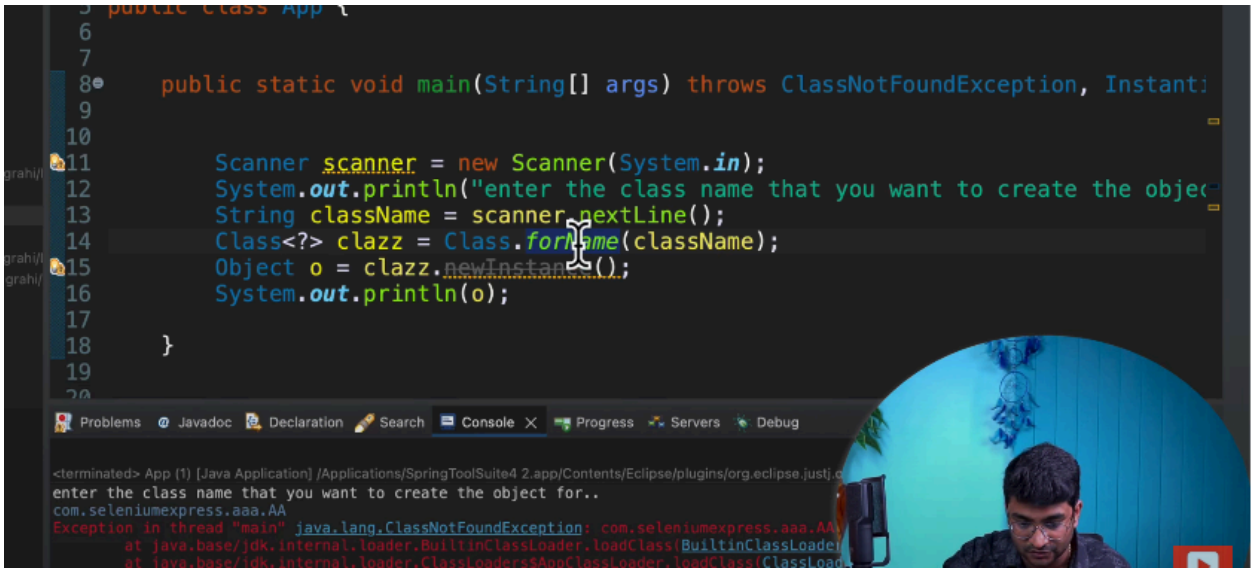
Create C class object.



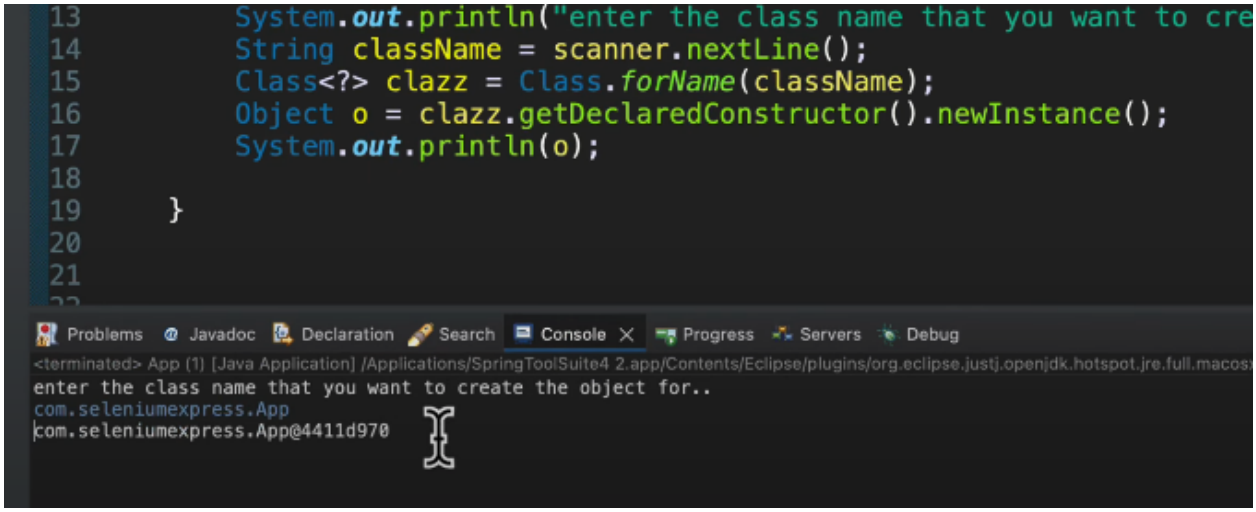
Without code change creating A class object dynamically



Whenever we enter wrong input - it will give ClassNotFoundException. (this exception comes only during dynamic loading of class)



The newInstance method is deprecated. The below one is a new way / method.



This Class.forName in real time example could be if want to download DB driver which can JDBC Driver/SQL Drive/Portugues Driver/noSQL driver/etc