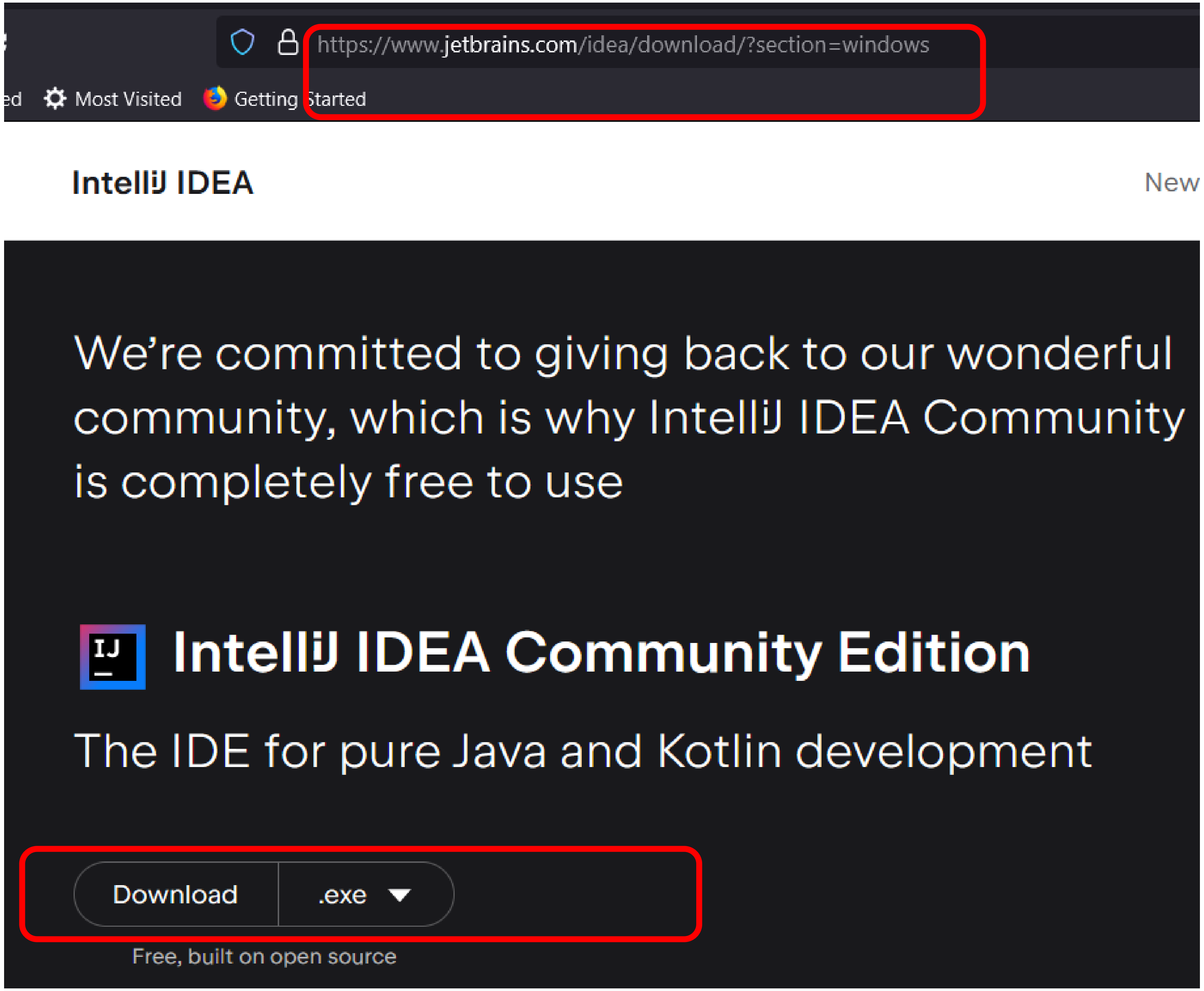
IntelliJ

Installation



https://www.jetbrains.com/idea/download/?section=windows

IntelliJ IDEA

We're committed to giving back to our wonderful community, which is why IntelliJ IDEA Community is completely free to use

IntelliJ IDEA Community Edition

The IDE for pure Java and Kotlin development

Free, built on open source

File structures

- Create an IntelliJ project -> contains all your projects (each project is called a module in IntelliJ).
- Then create an IntelliJ module for each of your project.
- You can create package, class, etc.

- Beware
  - Dragging files to IntelliJ Module folder will move files (not copy).
  - Try copy/paste instead if you don’t want to lose your work.

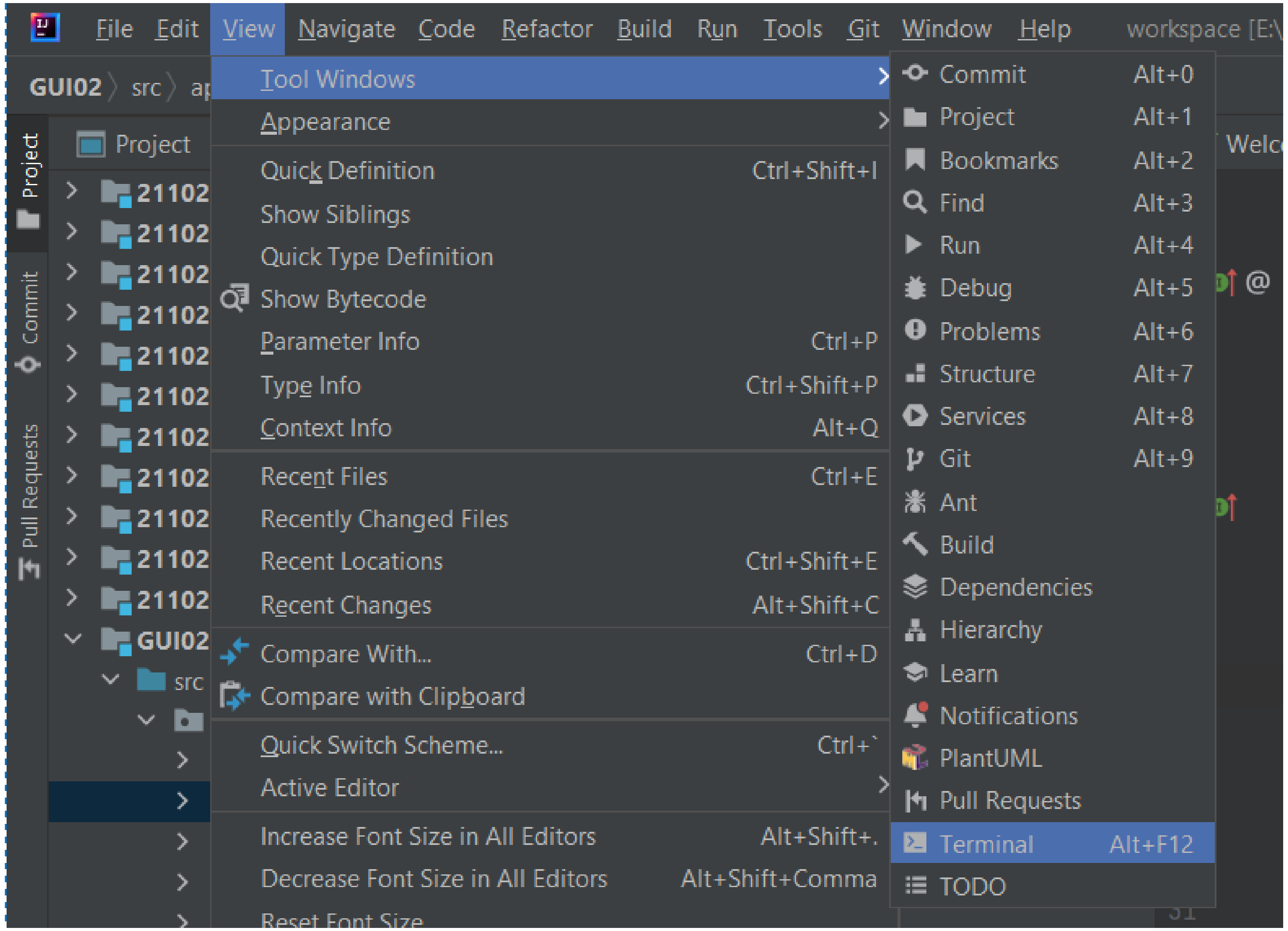
Hot Keys

- Code completion -> Enter
- Search anything -> Shift and then Shift
- Change Mode -> Choose “View -> Appearance” from the menu
  - There is a presentation mode!
- Zoom -> Shift + Alt + + or Shift + Alt + -
- Fix -> Alt + Enter , use F2/Shift + F2 to move to next/previous problem
- format code -> Ctrl + Alt + L
- Use sout to do System.out.println()

IntelliJ normally needs all files to compile in order to run a file in the same module

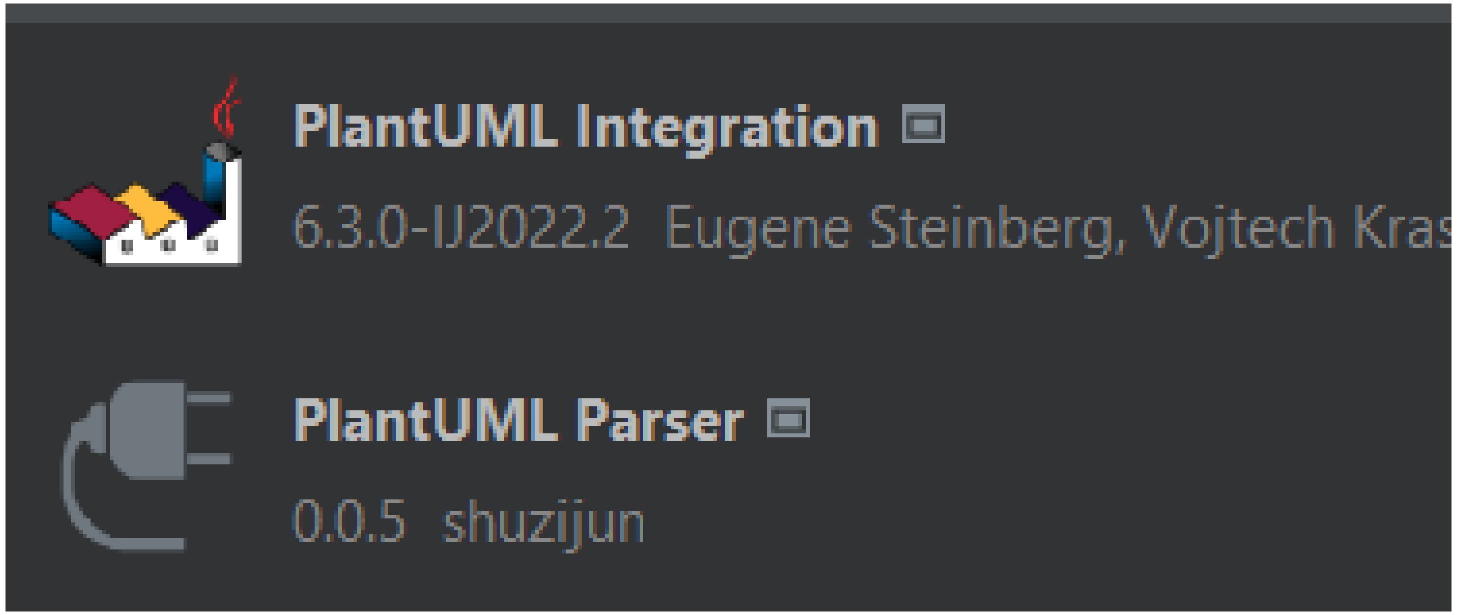
- Must make all files compiled.
- Eclipse compiler (compile all files while editing) is not available for Java version 19 or higher (sadly).
  - IntelliJ can compile only 1 file while editing (so you may not see errors on other dependent files).

How to open a Console window (if not open)



UML Generation

- Just install these plug-in:



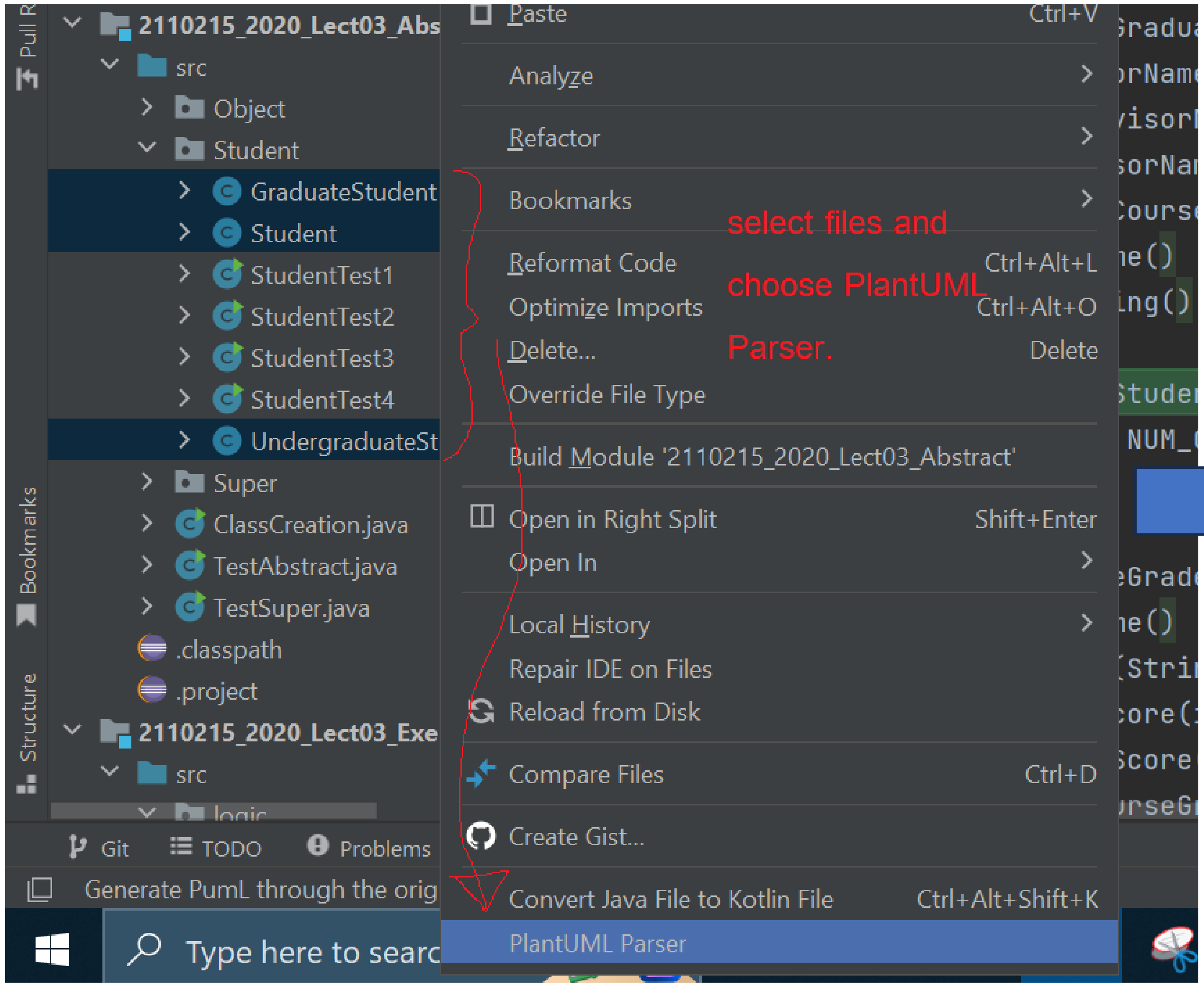
PlantUML Integration

6.3.0-IJ2022.2 Eugene Steinberg, Vojtech Kras

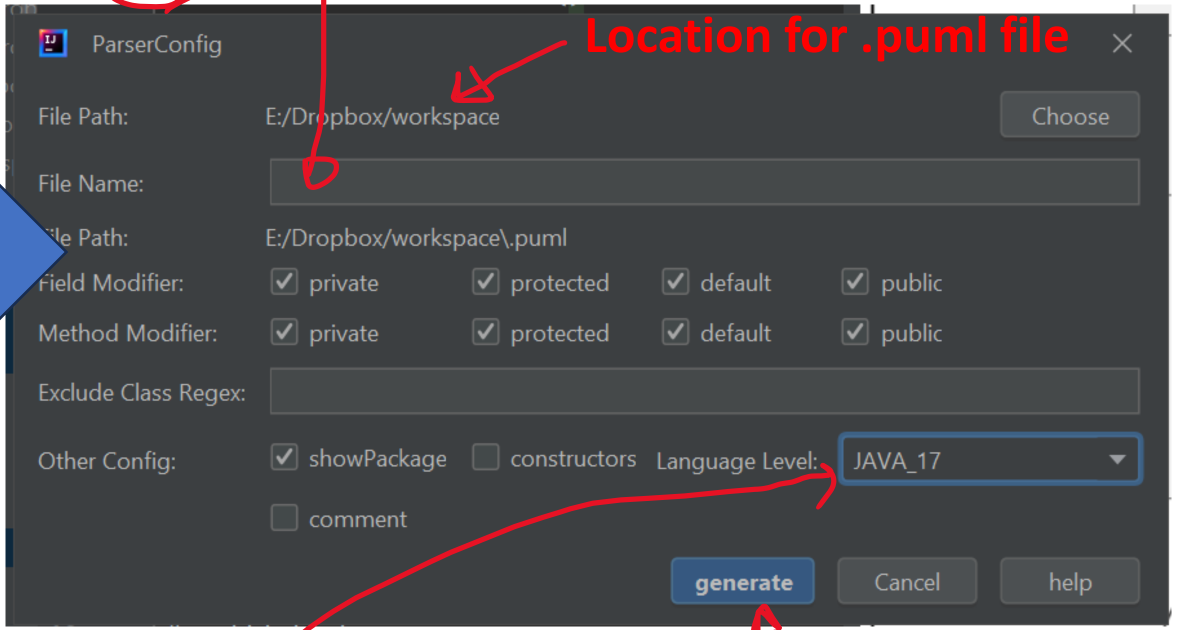
PlantUML Parser

0.0.5 shuzijun

When you want to generate UML:



select files and choose PlantUML Parser.



File Path: E:/Dropbox/workspace

File Name:

File Path: E:/Dropbox/workspace\.puml

Field Modifier: ✓ private ✓ protected ✓ default ✓ public

Method Modifier: ✓ private ✓ protected ✓ default ✓ public

Exclude Class Regex:

Other Config: ✓ showPackage ☐ constructors Language Level: JAVA_17

☐ comment

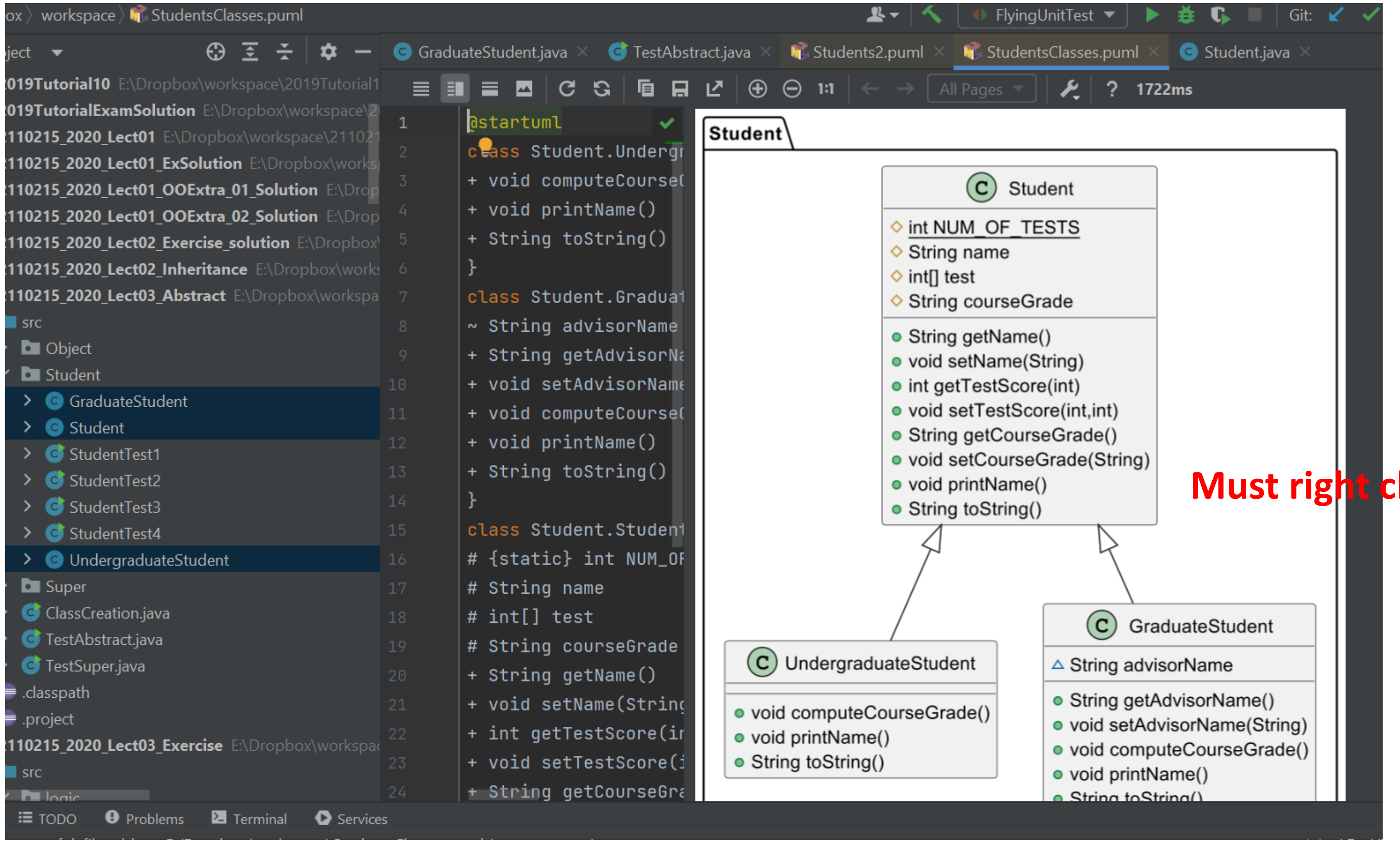
1 **File name for .puml file**

Location for .puml file

2 **Select Java version**

3

A UML Panel will emerge! Showing the diagram

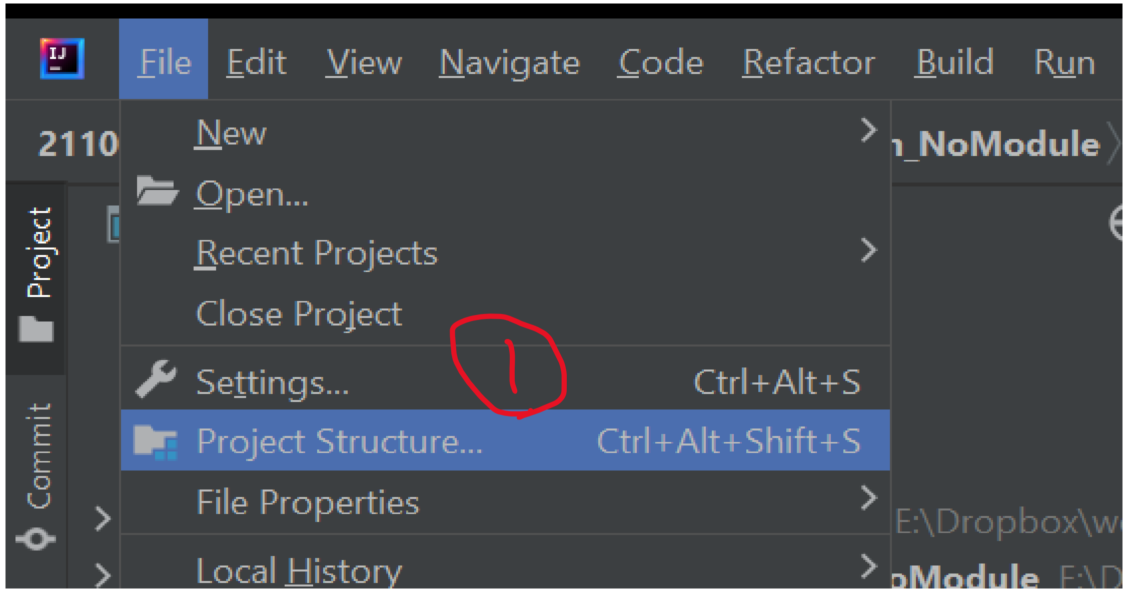


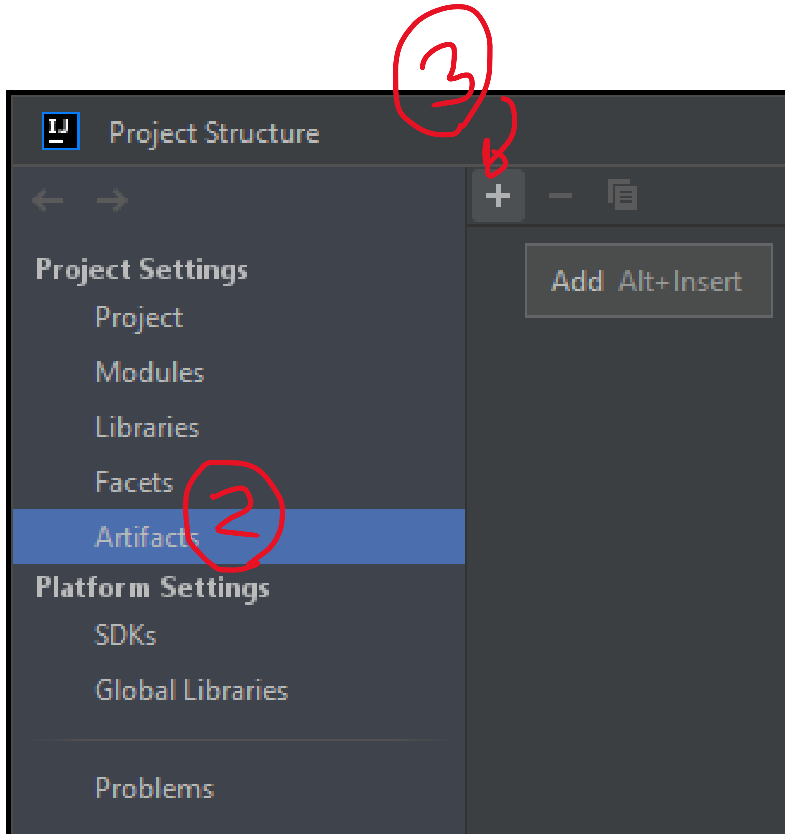
```
019Tutorial10 E:\Dropbox\workspace\2019Tutorial1
019TutorialExamSolution E:\Dropbox\workspace\2
110215_2020_Lect01 E:\Dropbox\workspace\21102
110215_2020_Lect01_ExSolution E:\Dropbox\works
110215_2020_Lect01_OOExtra_01_Solution E:\Drop
110215_2020_Lect01_OOExtra_02_Solution E:\Drop
110215_2020_Lect02_Exercise_solution E:\Dropbox
110215_2020_Lect02_Inheritance E:\Dropbox\work
110215_2020_Lect03_Abstract E:\Dropbox\workspa
  src
    Object
    Student
      GraduateStudent
      Student
      StudentTest1
      StudentTest2
      StudentTest3
      StudentTest4
      UndergraduateStudent
    Super
    ClassCreation.java
    TestAbstract.java
    TestSuper.java
  .classpath
  .project
110215_2020_Lect03_Exercise E:\Dropbox\workspa
  src
    logic
```

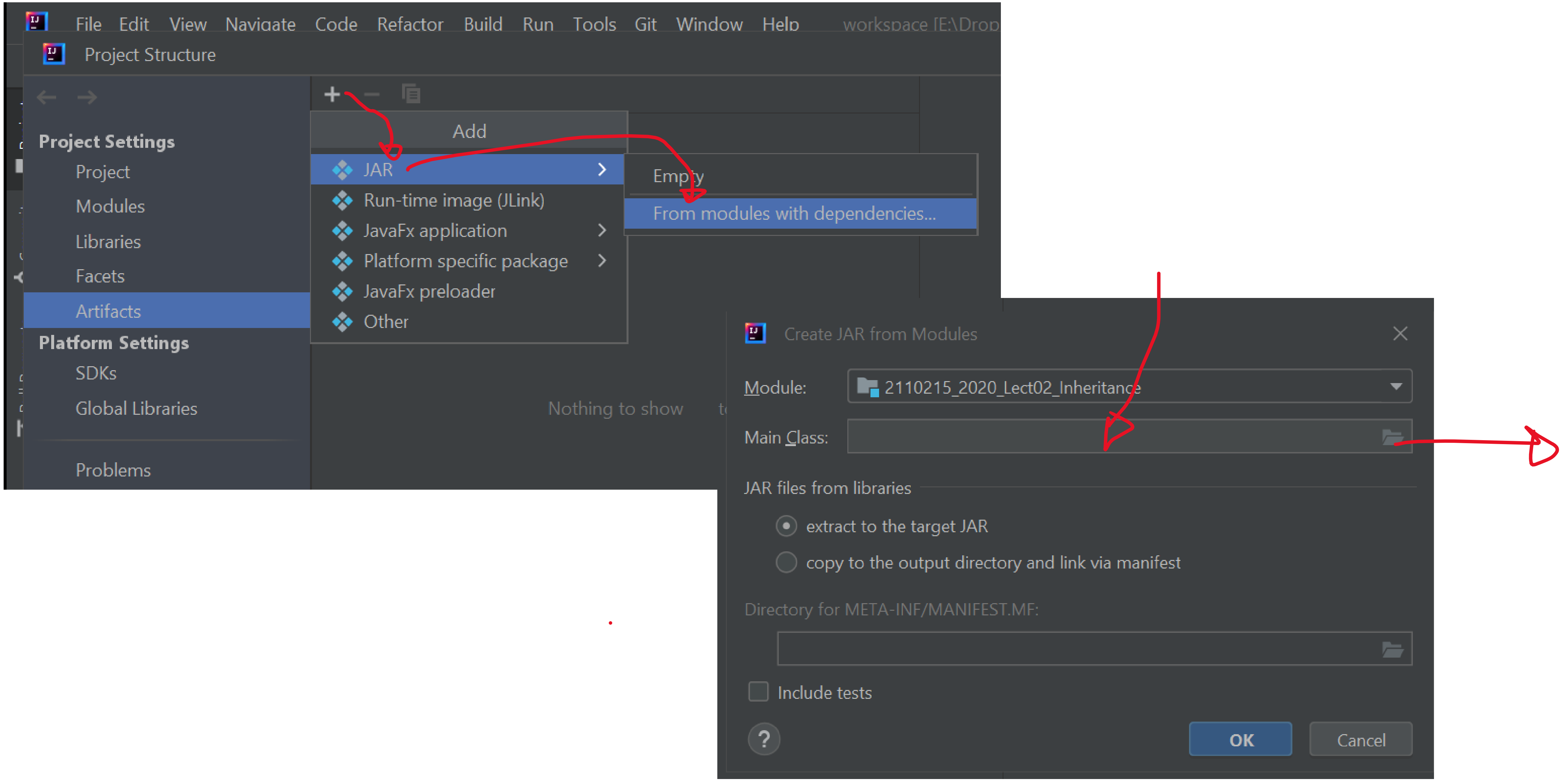
```
@startuml
class Student.Undergr
+ void computeCourse(
+ void printName()
+ String toString()
}
class Student.Graduat
~ String advisorName
+ String getAdvisorNa
+ void setAdvisorName
+ void computeCourse(
+ void printName()
+ String toString()
}
class Student.Student
# {static} int NUM_OF
# String name
# int[] test
# String courseGrade
+ String getName()
+ void setName(Strin
+ int getTestScore(in
+ void setTestScore(i
+ String getCourseGra
```

Must right click to save as picture.

How to export jar file (setting, the actual export is done later)







Module: 2110215_2020_Lect02_Inheritance

Main Class:

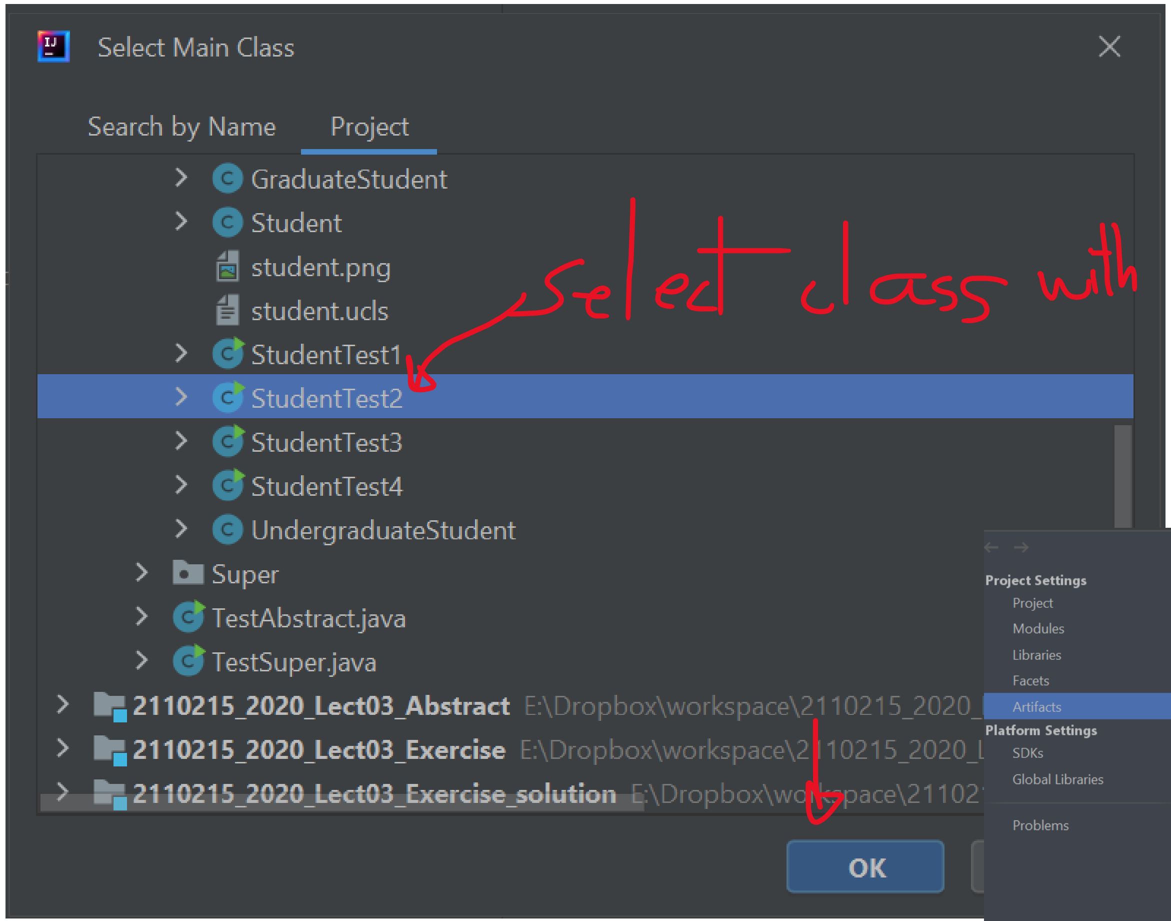
JAR files from libraries

[✓ extract to the target JAR]

copy to the output directory and link via manifest

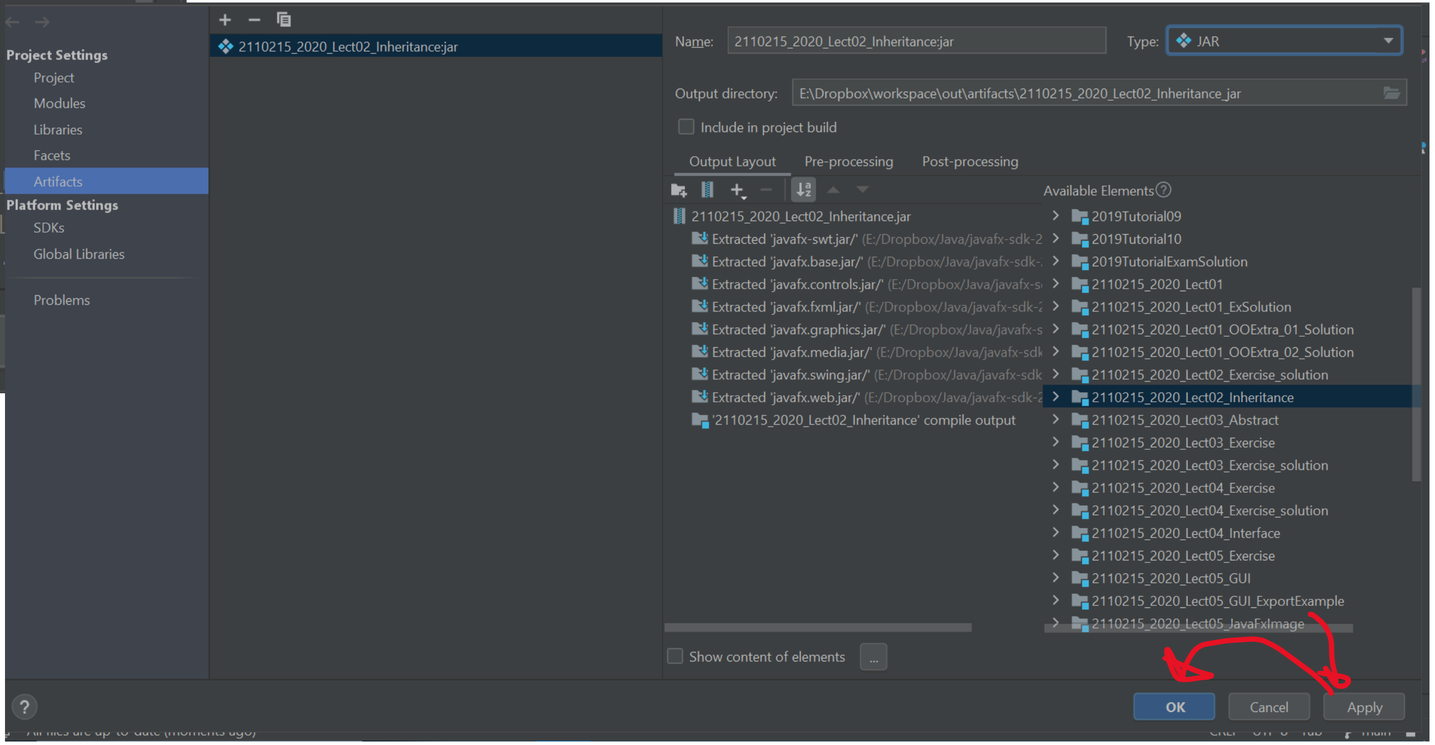
Directory for META-INF/MANIFEST.MF:

☐ Include tests



GraduateStudent
Student
student.png
student.ucls
StudentTest1
StudentTest2
StudentTest3
StudentTest4
UndergraduateStudent
Super
TestAbstract.java
TestSuper.java
2110215_2020_Lect03_Abstract E:\Dropbox\workspace\2110215_2020_
2110215_2020_Lect03_Exercise E:\Dropbox\workspace\2110215_2020_L
2110215_2020_Lect03_Exercise_solution E:\Dropbox\workspace\21102

select class with main



2110215_2020_Lect02_Inheritance:jar

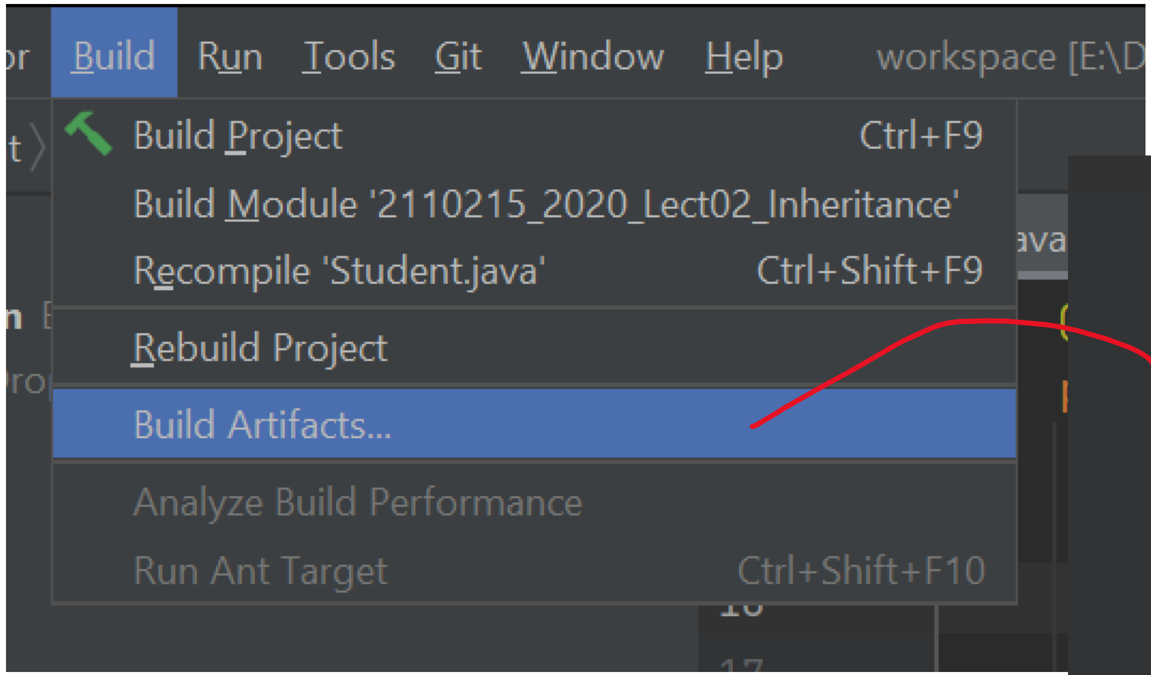
Name: 2110215_2020_Lect02_Inheritance:jar
Type: JAR
Output directory: E:\Dropbox\workspace\out\artifacts\2110215_2020_Lect02_Inheritance_jar
☐ Include in project build

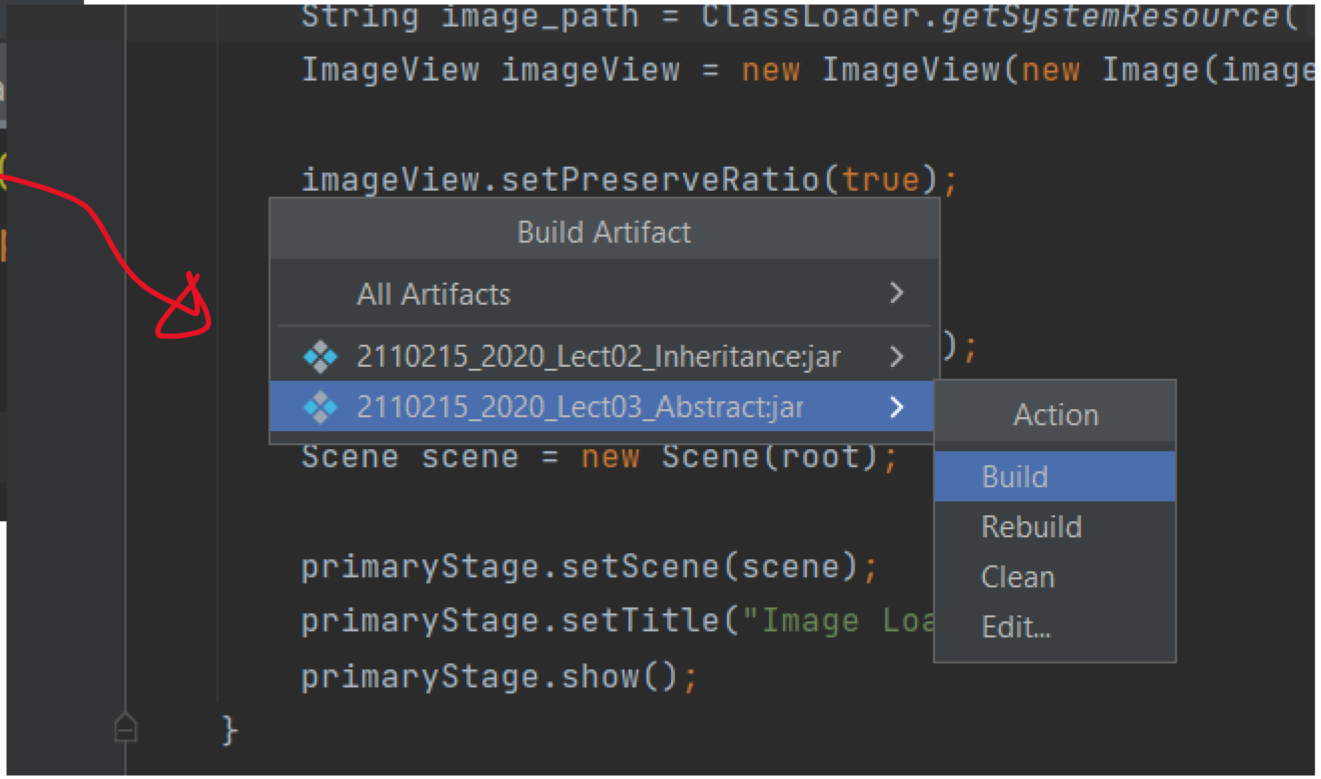
2110215_2020_Lect02_Inheritance.jar
Extracted 'javafx-swt.jar/' (E:/Dropbox/Java/javafx-sdk-2
Extracted 'javafx.base.jar/' (E:/Dropbox/Java/javafx-sdk-
Extracted 'javafx.controls.jar/' (E:/Dropbox/Java/javafx-s
Extracted 'javafx.fxml.jar/' (E:/Dropbox/Java/javafx-sdk-2
Extracted 'javafx.graphics.jar/' (E:/Dropbox/Java/javafx-s
Extracted 'javafx.media.jar/' (E:/Dropbox/Java/javafx-sdk
Extracted 'javafx.swing.jar/' (E:/Dropbox/Java/javafx-sdk
Extracted 'javafx.web.jar/' (E:/Dropbox/Java/javafx-sdk-2
'2110215_2020_Lect02_Inheritance' compile output

2019Tutorial09
2019Tutorial10
2019TutorialExamSolution
2110215_2020_Lect01
2110215_2020_Lect01_ExSolution
2110215_2020_Lect01_OOExtra_01_Solution
2110215_2020_Lect01_OOExtra_02_Solution
2110215_2020_Lect02_Exercise_solution
2110215_2020_Lect02_Inheritance
2110215_2020_Lect03_Abstract
2110215_2020_Lect03_Exercise
2110215_2020_Lect03_Exercise_solution
2110215_2020_Lect04_Exercise
2110215_2020_Lect04_Exercise_solution
2110215_2020_Lect04_Interface
2110215_2020_Lect05_Exercise
2110215_2020_Lect05_GUI
2110215_2020_Lect05_GUI_ExportExample
2110215_2020_Lect05_JavaFxImage

☐ Show content of elements

Now do the actual export

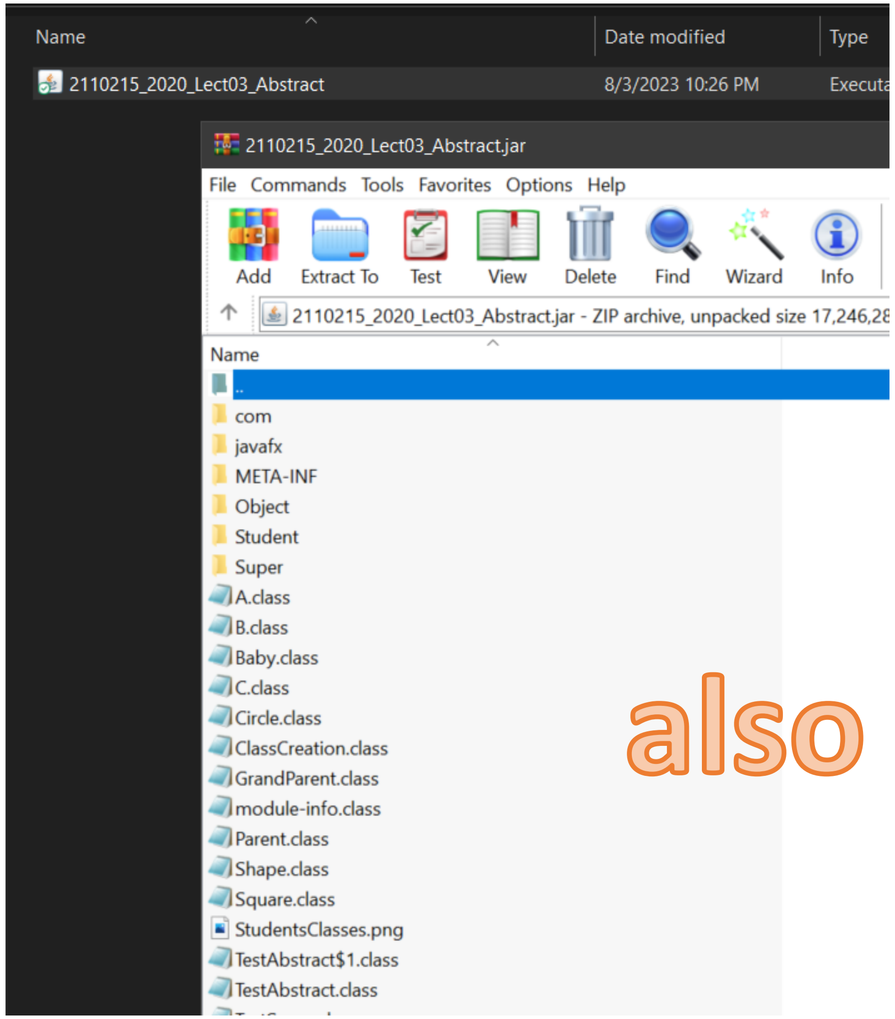

r
Build
Run
Tools
Git
Window
Help
workspace [E:\D
Build Project Ctrl+F9
Build Module '2110215_2020_Lect02_Inheritance'
Recompile 'Student.java' Ctrl+Shift+F9
Rebuild Project
Build Artifacts...
Analyze Build Performance
Run Ant Target Ctrl+Shift+F10



```
    String image_path = ClassLoader.getSystemResource(
    ImageView imageView = new ImageView(new Image(image

    imageView.setPreserveRatio(true);

                                                    );

    Scene scene = new Scene(root);

    primaryStage.setScene(scene);
    primaryStage.setTitle("Image Loa
    primaryStage.show();
}
```

Build Artifact
All Artifacts
2110215_2020_Lect02_Inheritance:jar
2110215_2020_Lect03_Abstract:jar
Action
Build
Rebuild
Clean
Edit...

All class files will be zipped!



2110215_2020_Lect03_Abstract 8/3/2023 10:26 PM Executa

2110215_2020_Lect03_Abstract.jar

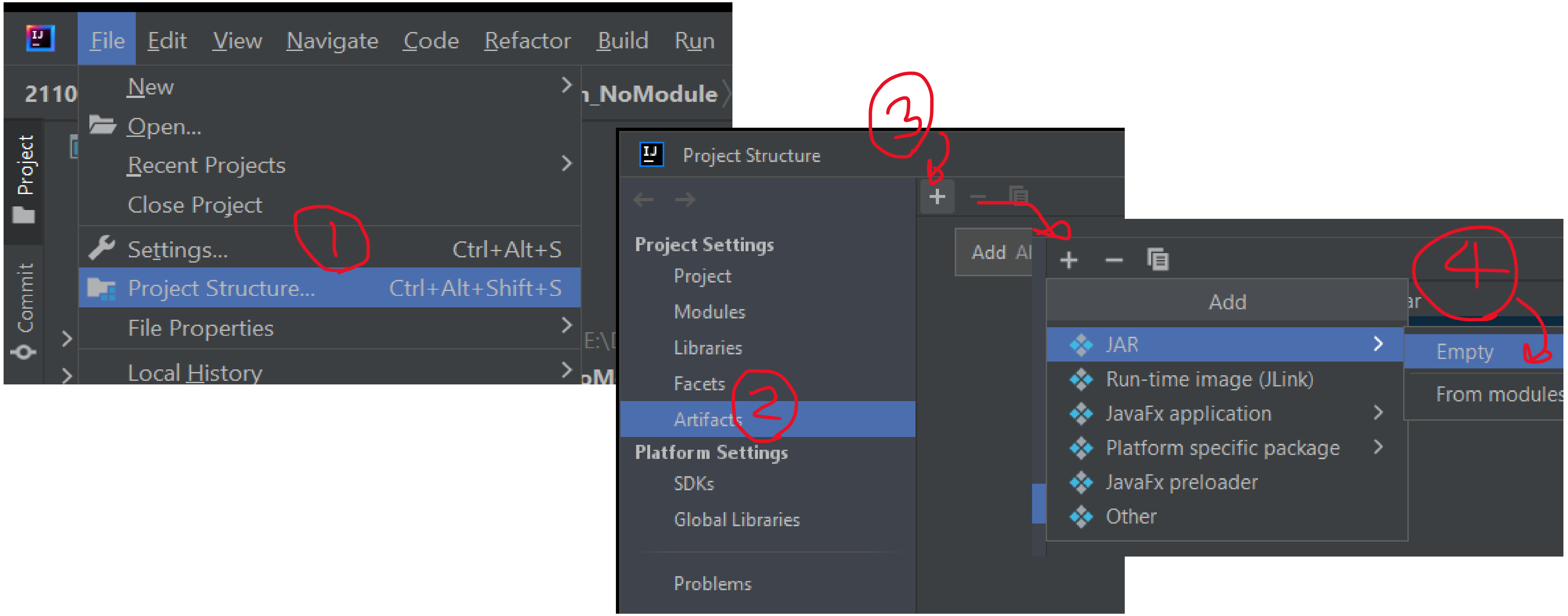
2110215_2020_Lect03_Abstract.jar - ZIP archive, unpacked size 17,246,28

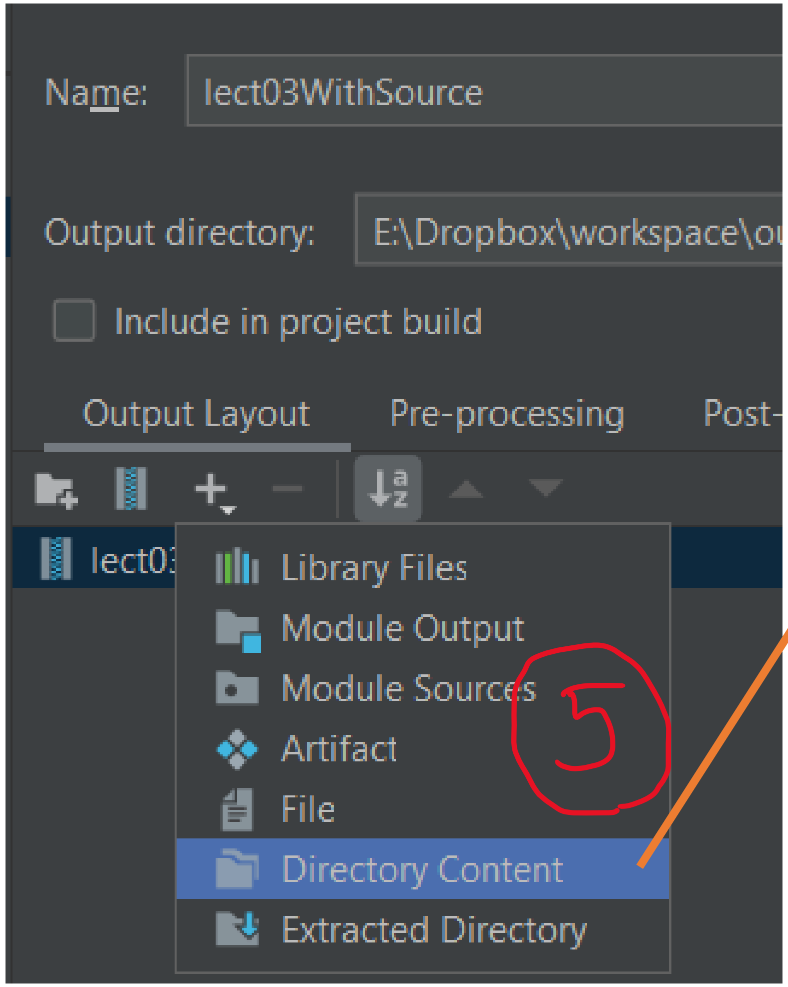
Name
..
com
javafx
META-INF
Object
Student
Super
A.class
B.class
Baby.class
C.class
Circle.class
ClassCreation.class
GrandParent.class
module-info.class
Parent.class
Shape.class
Square.class
StudentsClasses.png
TestAbstract$1.class
TestAbstract.class

But what if you also want the .java files?

To include .java files in JAR

- Make custom artifact configuration.



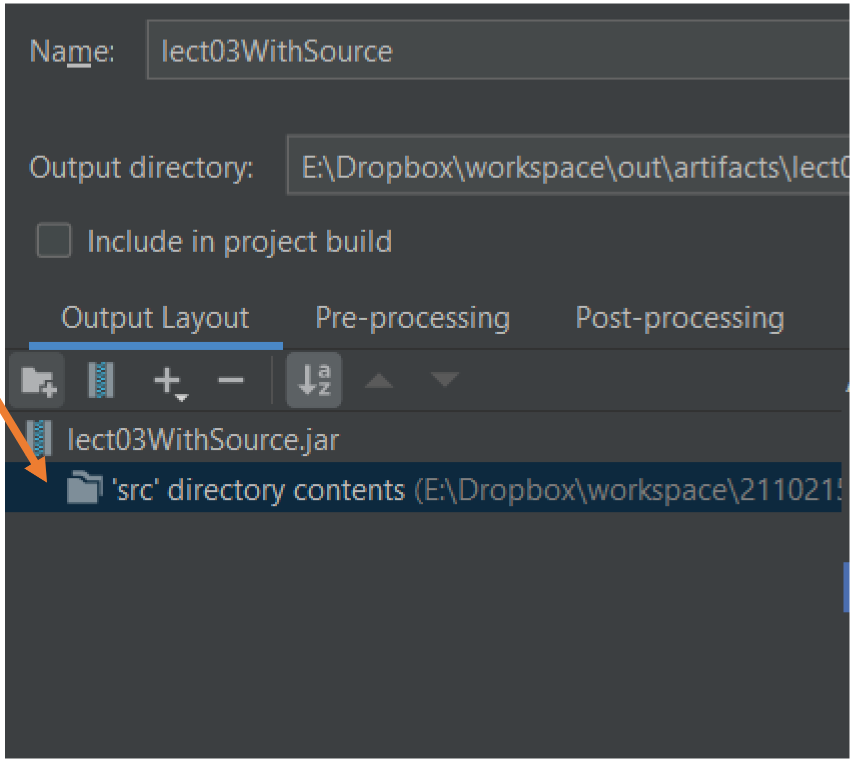


Name: lect03WithSource

Output directory: E:\Dropbox\workspace\ou

Include in project build

Choose src folder, it will then show.



Name: lect03WithSource

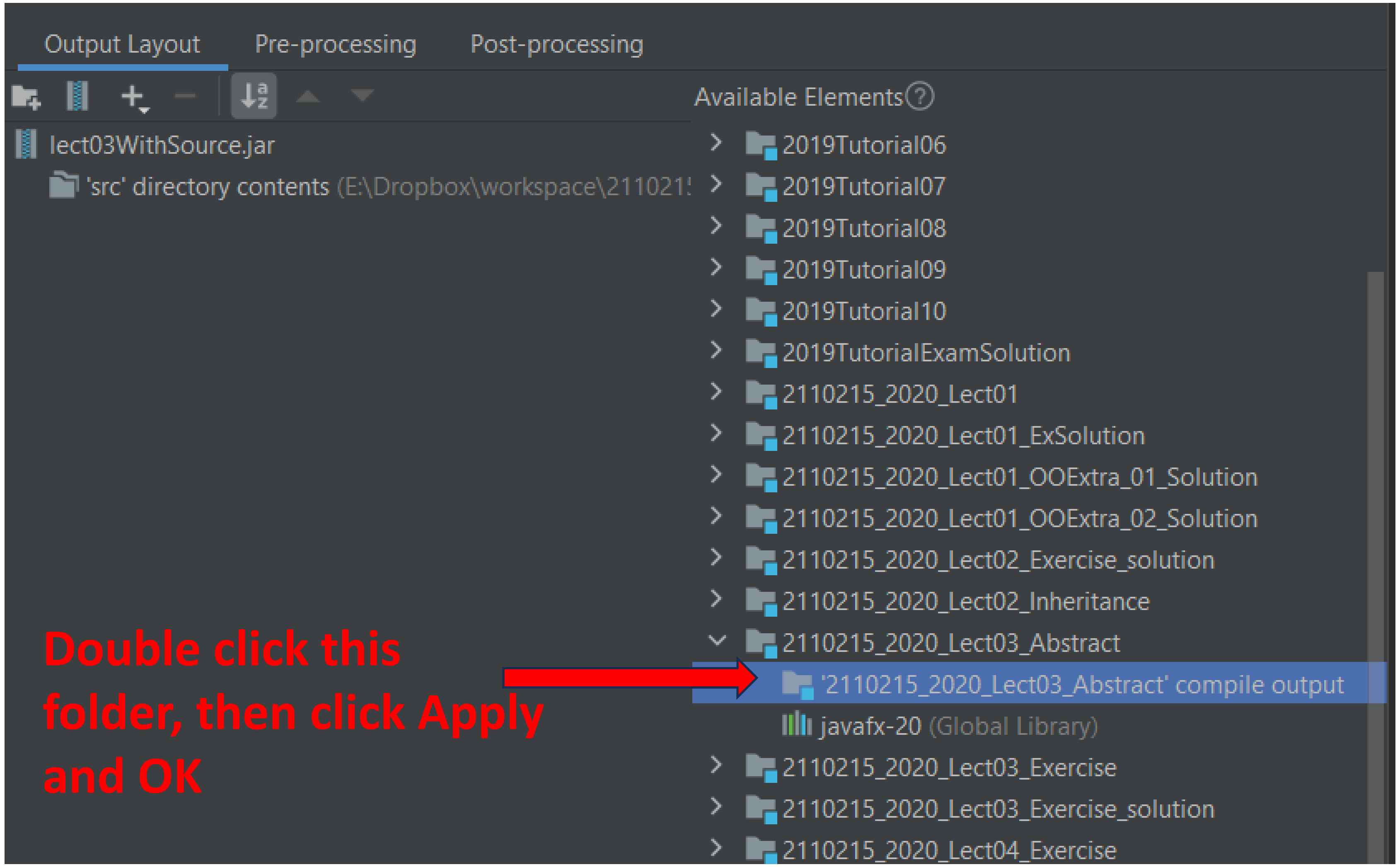
Output directory: E:\Dropbox\workspace\out\artifacts\lect0

Include in project build

lect03WithSource.jar

'src' directory contents (E:\Dropbox\workspace\211021

But you'll have to select .class files too



lect03WithSource.jar
'src' directory contents (E:\Dropbox\workspace\211021

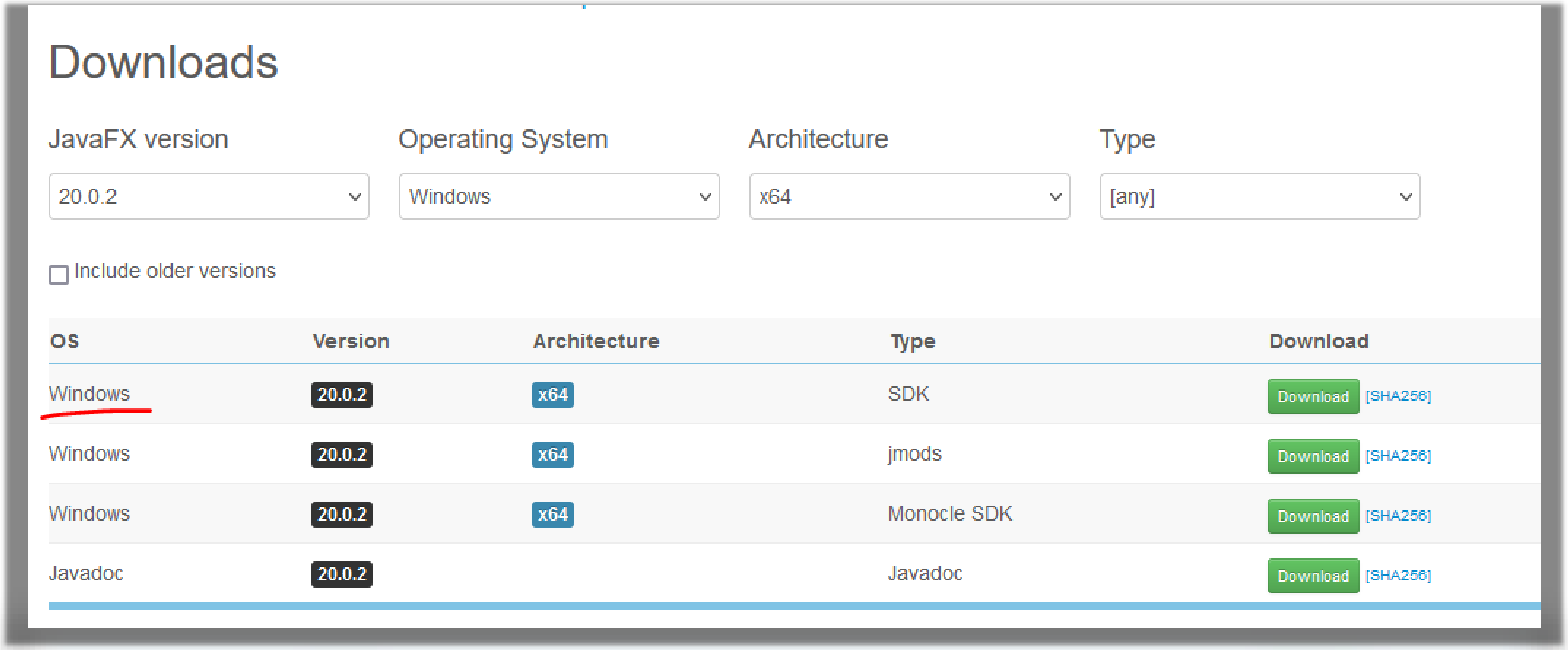
2019Tutorial06
2019Tutorial07
2019Tutorial08
2019Tutorial09
2019Tutorial10
2019TutorialExamSolution
2110215_2020_Lect01
2110215_2020_Lect01_ExSolution
2110215_2020_Lect01_OOExtra_01_Solution
2110215_2020_Lect01_OOExtra_02_Solution
2110215_2020_Lect02_Exercise_solution
2110215_2020_Lect02_Inheritance
2110215_2020_Lect03_Abstract
'2110215_2020_Lect03_Abstract' compile output
javafx-20 (Global Library)
2110215_2020_Lect03_Exercise
2110215_2020_Lect03_Exercise_solution
2110215_2020_Lect04_Exercise

Double click this folder, then click Apply and OK

Now when you build JAR,
It will contain both .java and .class (but you must make sure all files are compiled because IntelliJ does not compile some files unless you explicitly do it)

javaFX

- First, download it from https://gluonhq.com/products/javafx/ and unzip to any folder you like.



Downloads

JavaFX version: 20.0.2
Operating System: Windows
Architecture: x64
Type: [any]

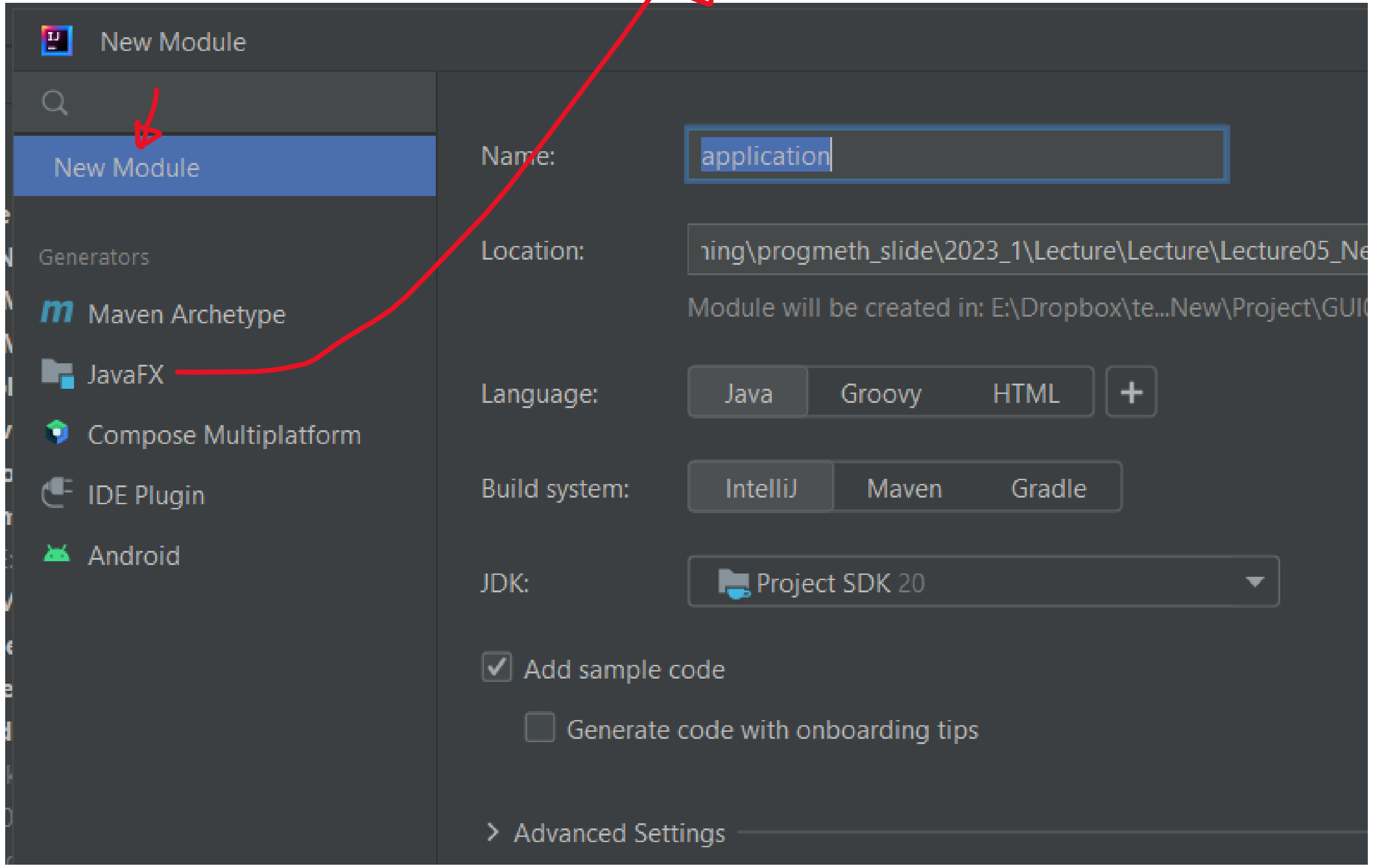
☐ Include older versions

OS	Version	Architecture	Type	Download
Windows	20.0.2	x64	SDK	Download [SHA256]
Windows	20.0.2	x64	jmods	Download [SHA256]
Windows	20.0.2	x64	Monocle SDK	Download [SHA256]
Javadoc	20.0.2		Javadoc	Download [SHA256]

How to create a Java (including JavaFX) project

- File- > New Module

Do not choose this. The system will lock only 1 file to Be runnable if you choose this.



New Module

New Module

Generators

Maven Archetype

JavaFX

Compose Multiplatform

IDE Plugin

Android

Name: application

Location: ɿing\progmeth_slide\2023_1\Lecture\Lecture\Lecture05_Ne

Module will be created in: E:\Dropbox\te...New\Project\GUI(

Language: Java | Groovy | HTML | +

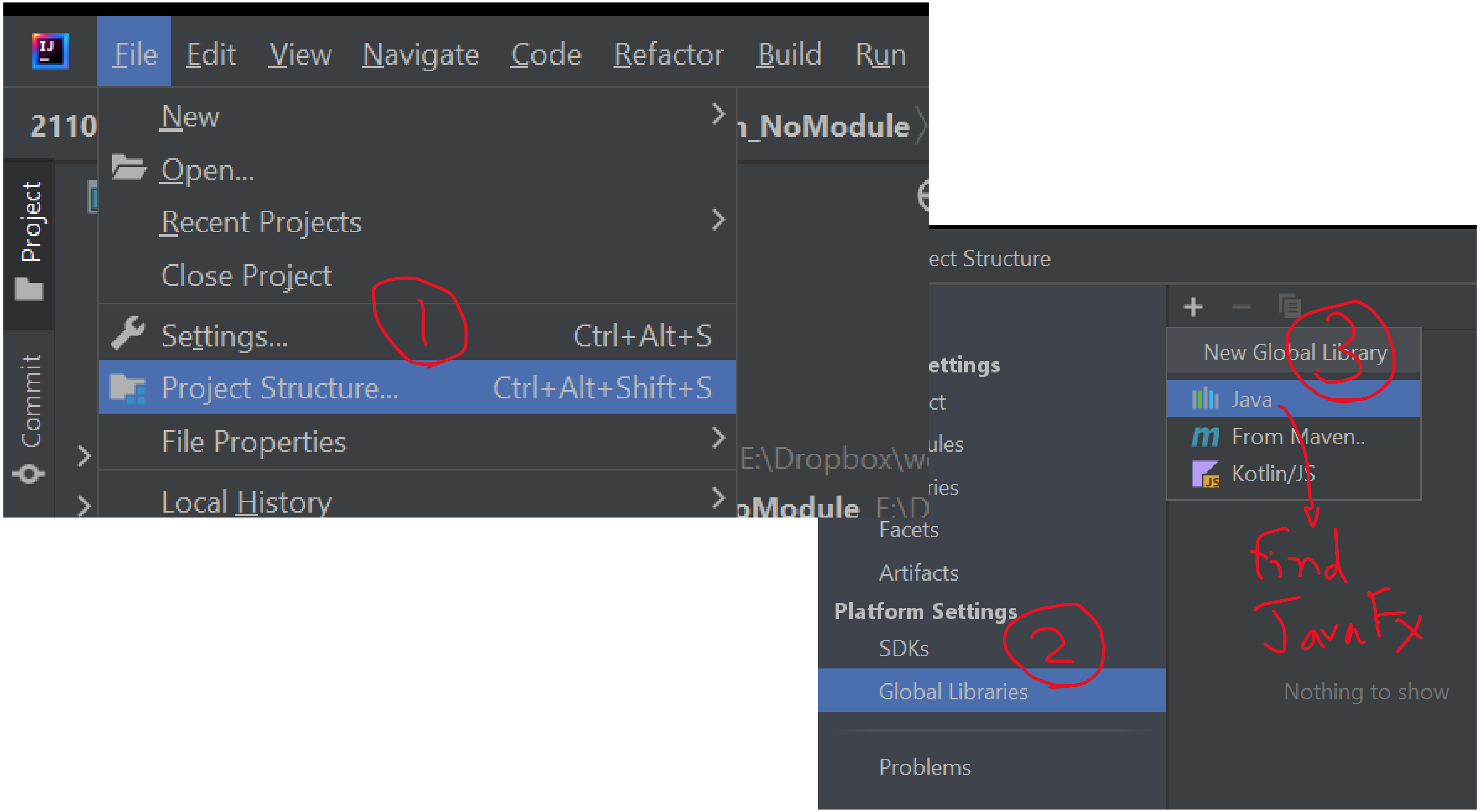
Build system: IntelliJ | Maven | Gradle

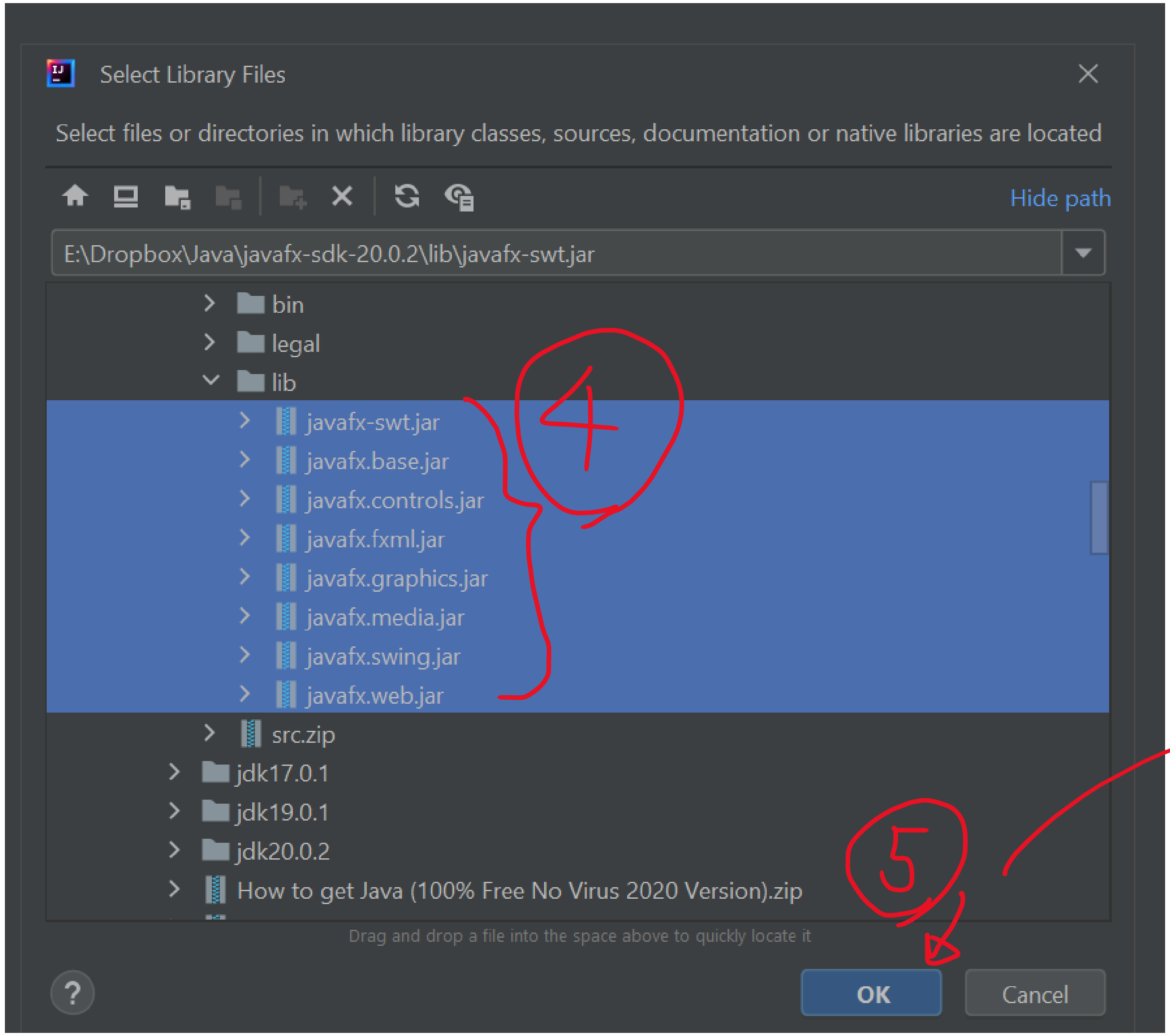
JDK: Project SDK 20

[✓] Add sample code

[ ] Generate code with onboarding tips

> Advanced Settings


File
Edit
View
Navigate
Code
Refactor
Build
Run
2110
_NoModule
New
Open...
Recent Projects
Close Project
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
File Properties
Local History
Project
Commit
E:\Dropbox\w
oModule E:\D
1
ect Structure
ettings
ct
ules
ries
Facets
Artifacts
Platform Settings
SDKs
Global Libraries
Problems
2
New Global Library
Java
From Maven..
Kotlin/JS
3
find JavaFx
Nothing to show


Select Library Files

Select files or directories in which library classes, sources, documentation or native libraries are located

Hide path

E:\Dropbox\Java\javafx-sdk-20.0.2\lib\javafx-swt.jar

```
bin
legal
lib
  javafx-swt.jar
  javafx.base.jar
  javafx.controls.jar
  javafx.fxml.jar
  javafx.graphics.jar
  javafx.media.jar
  javafx.swing.jar
  javafx.web.jar
src.zip
jdk17.0.1
jdk19.0.1
jdk20.0.2
How to get Java (100% Free No Virus 2020 Version).zip
```

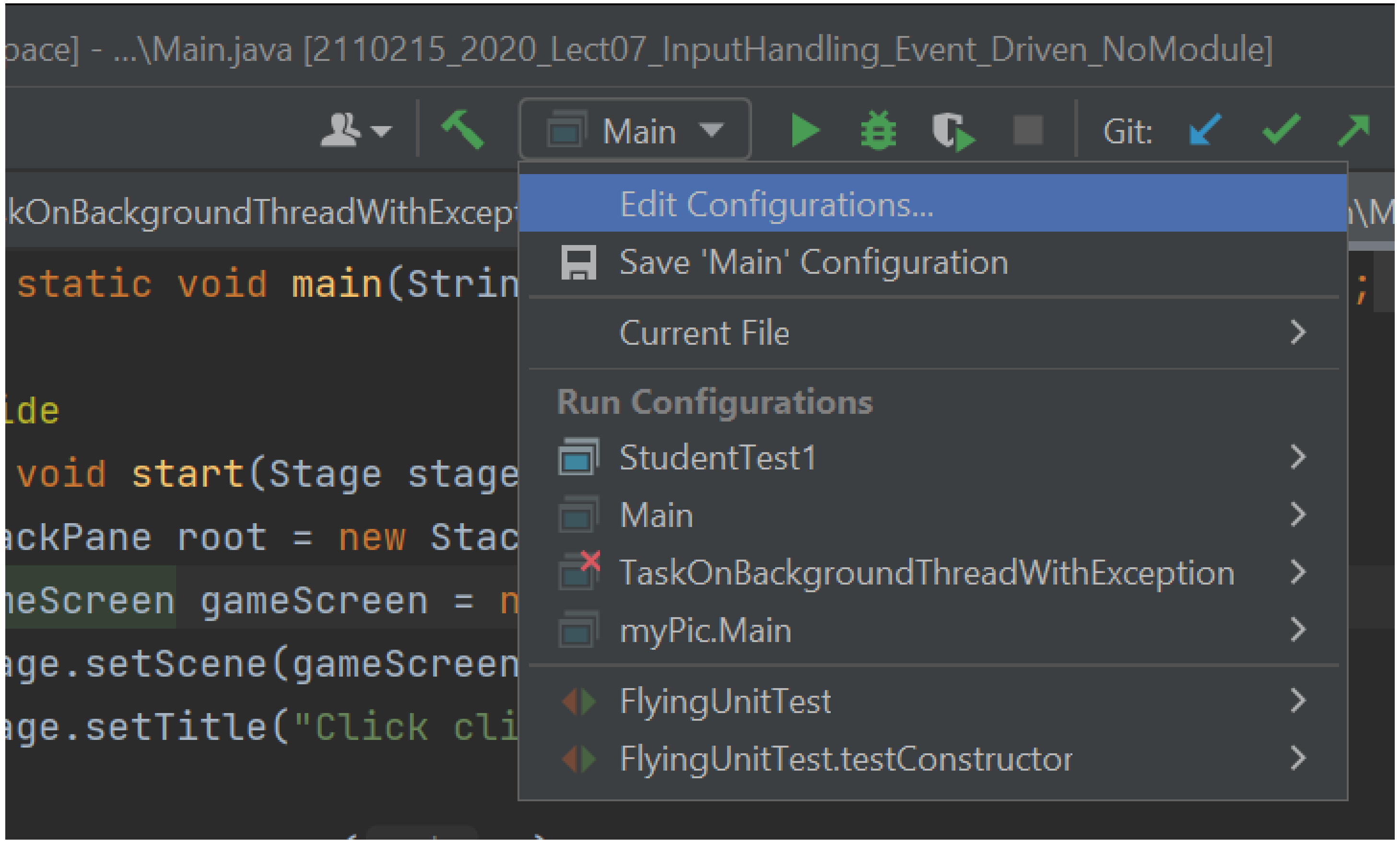
Drag and drop a file into the space above to quickly locate it

6

Choose modules you want to apply JavaFx

NOW JavaFX compiles, but still won't run!!

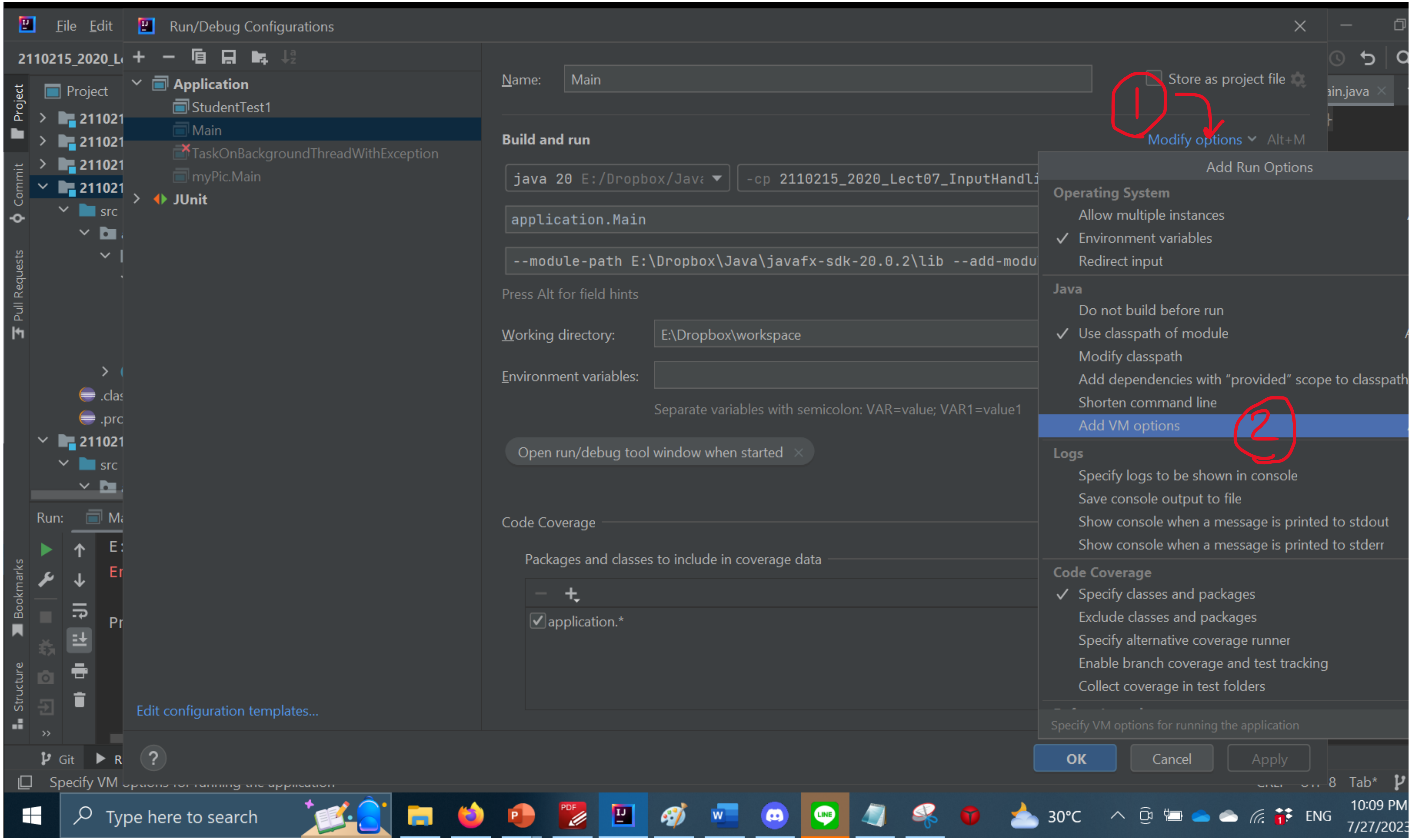
To make it run!



pace] - ...\Main.java [2110215_2020_Lect07_InputHandling_Event_Driven_NoModule]

```
static void main(Strin
ide
void start(Stage stage
ackPane root = new Stac
neScreen gameScreen = n
age.setScene(gameScreen
age.setTitle("Click cli
```

Main
Git:
kOnBackgroundThreadWithExcep
Edit Configurations...
Save 'Main' Configuration
Current File
Run Configurations
StudentTest1
Main
TaskOnBackgroundThreadWithException
myPic.Main
FlyingUnitTest
FlyingUnitTest.testConstructor


Run/Debug Configurations

Name: Main

Store as project file

Build and run

Modify options Alt+M

java 20 E:/Dropbox/Java

-cp 2110215_2020_Lect07_InputHandli

application.Main

--module-path E:\Dropbox\Java\javafx-sdk-20.0.2\lib --add-modu

Press Alt for field hints

Working directory: E:\Dropbox\workspace

Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started

Code Coverage

Packages and classes to include in coverage data

☑ application.*

Add Run Options

Operating System
- Allow multiple instances
- ✓ Environment variables
- Redirect input

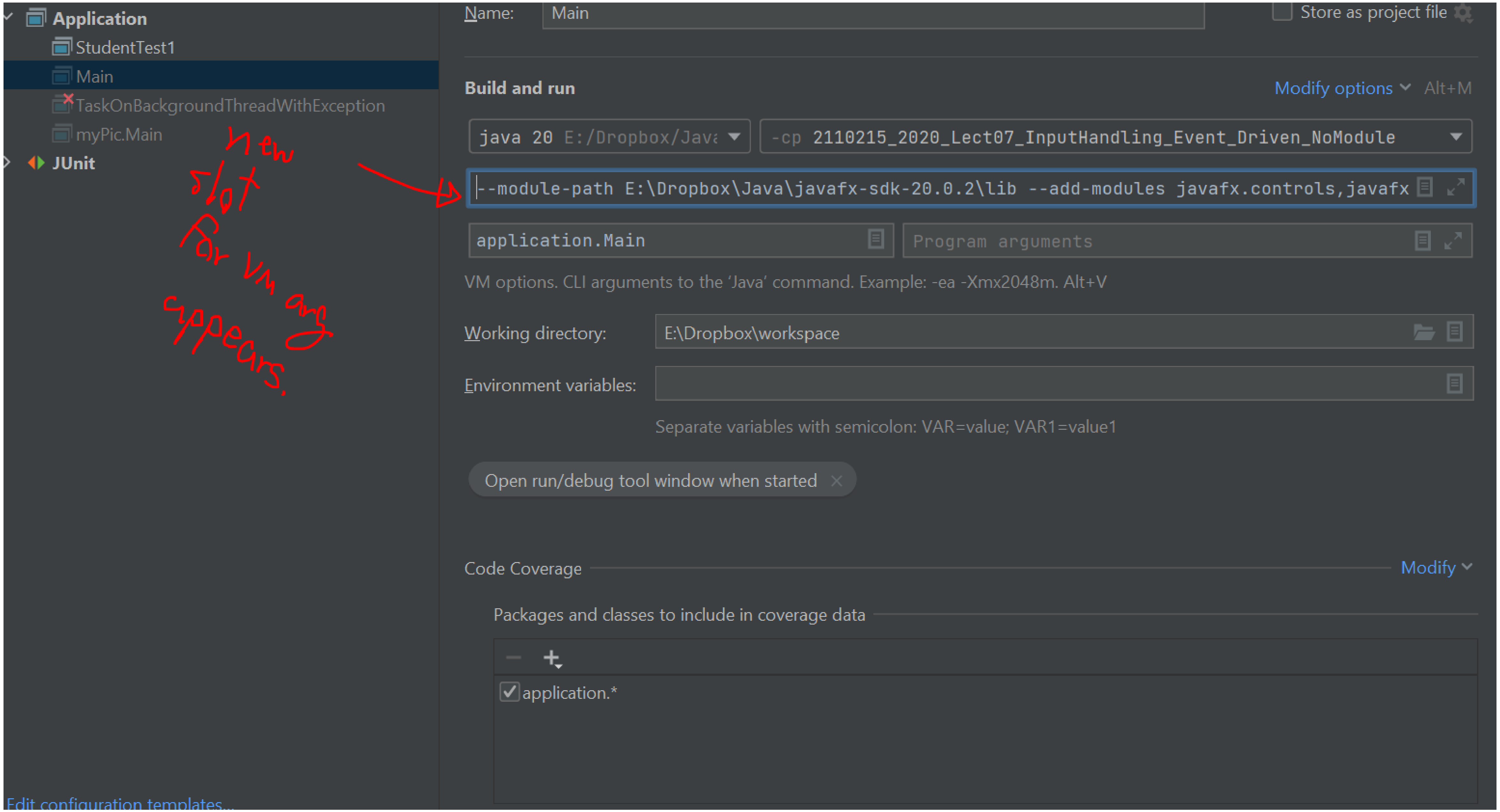
Java
- Do not build before run
- ✓ Use classpath of module
- Modify classpath
- Add dependencies with "provided" scope to classpath
- Shorten command line
- Add VM options

Logs
- Specify logs to be shown in console
- Save console output to file
- Show console when a message is printed to stdout
- Show console when a message is printed to stderr

Code Coverage
- ✓ Specify classes and packages
- Exclude classes and packages
- Specify alternative coverage runner
- Enable branch coverage and test tracking
- Collect coverage in test folders

Specify VM options for running the application

1
2
Application
StudentTest1
Main
TaskOnBackgroundThreadWithException
myPic.Main
JUnit
Edit configuration templates...
OK
Cancel
Apply



Application
StudentTest1
Main
TaskOnBackgroundThreadWithException
myPic.Main
JUnit

new slot for vm arg appears.

Name: Main

Store as project file

Build and run

Modify options Alt+M

java 20 E:/Dropbox/Java

-cp 2110215_2020_Lect07_InputHandling_Event_Driven_NoModule

```
--module-path E:\Dropbox\Java\javafx-sdk-20.0.2\lib --add-modules javafx.controls,javafx
```

application.Main

Program arguments

VM options. CLI arguments to the 'Java' command. Example: -ea -Xmx2048m. Alt+V

Working directory: E:\Dropbox\workspace

Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

Open run/debug tool window when started

Code Coverage

Modify

Packages and classes to include in coverage data

☑ application.*

Edit configuration templates...

Fill the slot with the following argument and click Apply!

Example:

Path to lib folder

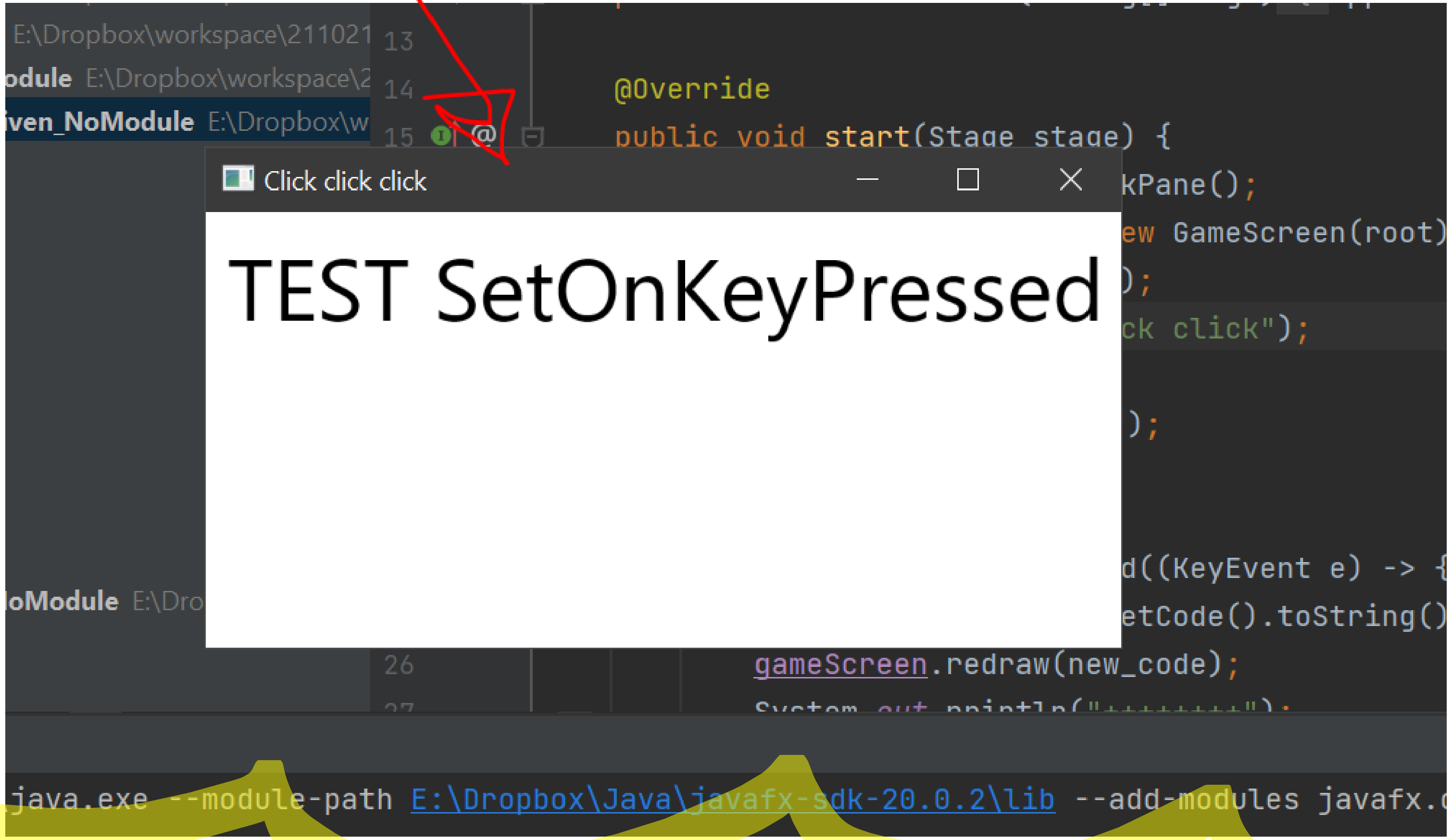
```
-module-path E:\Dropbox\Java\javafx-sdk-20.0.2\lib --add-modules
javafx.controls,javafx.fxml,javafx.graphics,javafx.media
```

standard for forms

picture

sound

It runs now!!



```
@Override
public void start(Stage stage) {
    kPane();
    ew GameScreen(root)
    );
    ck click");
    );
    d((KeyEvent e) -> {
    etCode().toString()
        gameScreen.redraw(new_code);
```

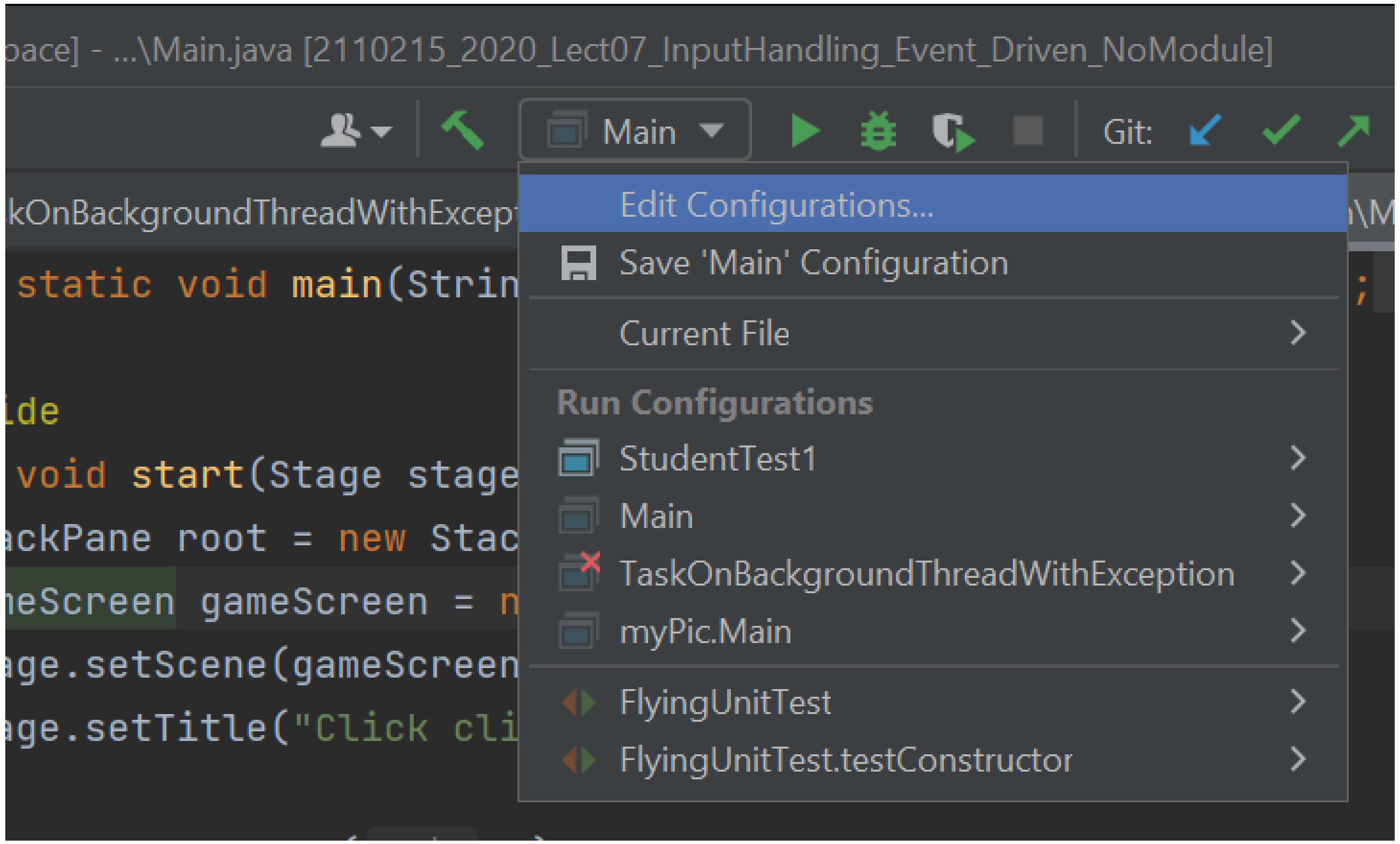
Click click click

TEST SetOnKeyPressed

```
java.exe --module-path E:\Dropbox\Java\javafx-sdk-20.0.2\lib --add-modules javafx.
```

But you have to set run configuration for each main (a lot to do if you have many JavaFx modules)

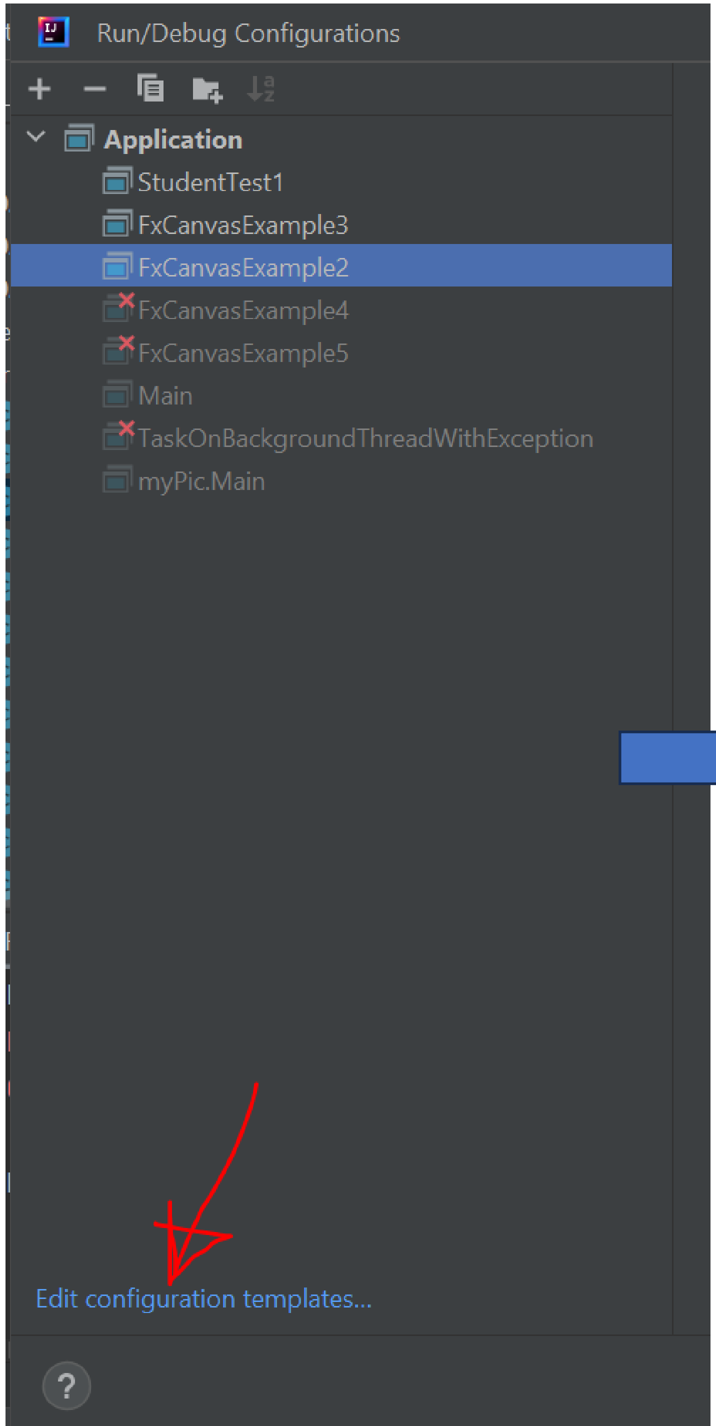
How to set a common run config.

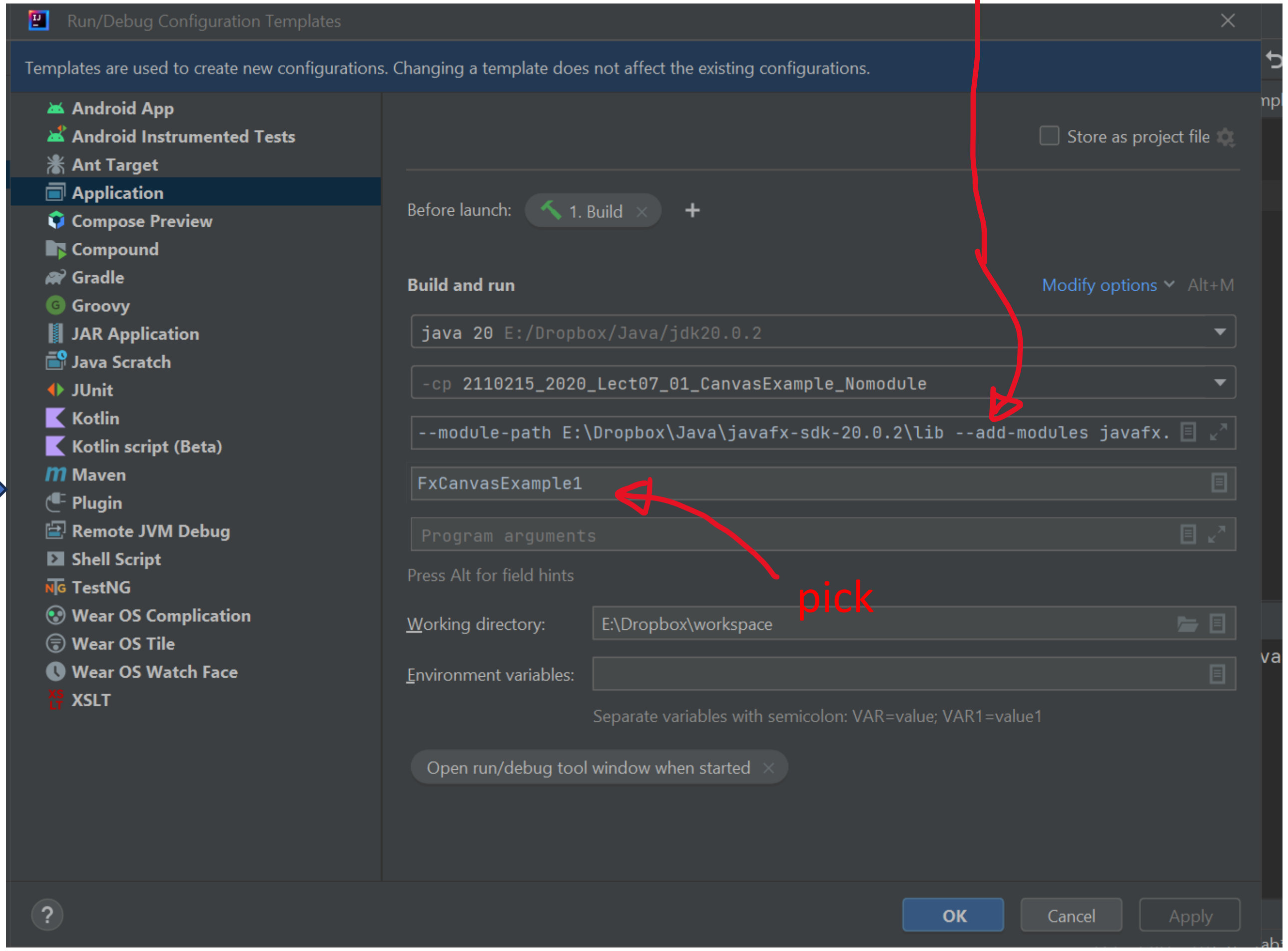


pace] - ...\Main.java [2110215_2020_Lect07_InputHandling_Event_Driven_NoModule]
Main
Git:
kOnBackgroundThreadWithExcep
Edit Configurations...
Save 'Main' Configuration
Current File
Run Configurations
StudentTest1
Main
TaskOnBackgroundThreadWithException
myPic.Main
FlyingUnitTest
FlyingUnitTest.testConstructor

```
 static void main(Strin

ide
 void start(Stage stage
ackPane root = new Stac
neScreen gameScreen = n
age.setScene(gameScreen
age.setTitle("Click cli
```

Set it just like the one you did before!!





Templates are used to create new configurations. Changing a template does not affect the existing configurations.

Before launch: 1. Build

Build and run

```
java 20 E:/Dropbox/Java/jdk20.0.2
-cp 2110215_2020_Lect07_01_CanvasExample_Nomodule
--module-path E:\Dropbox\Java\javafx-sdk-20.0.2\lib --add-modules javafx.
FxCanvasExample1
Program arguments
```

Press Alt for field hints

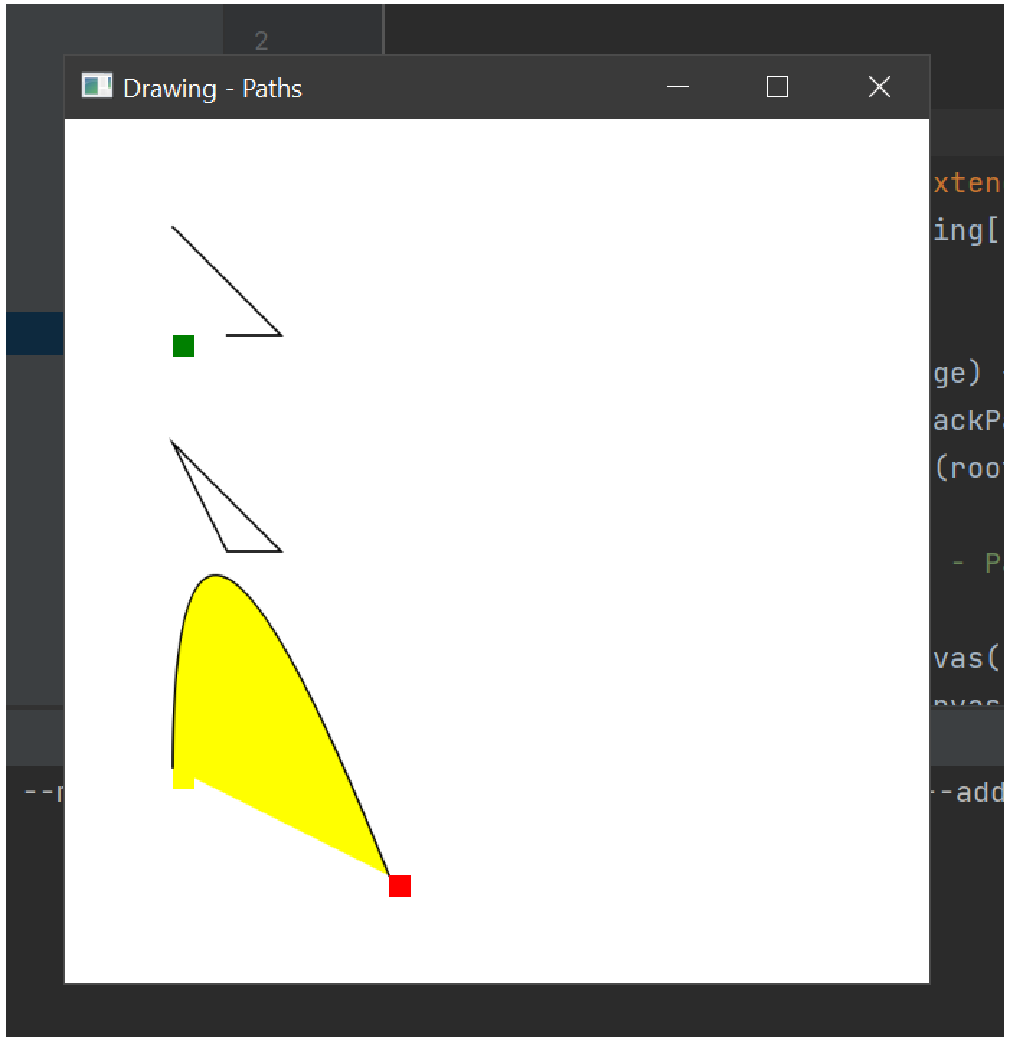
Working directory: E:\Dropbox\workspace

Environment variables:

Separate variables with semicolon: VAR=value; VAR1=value1

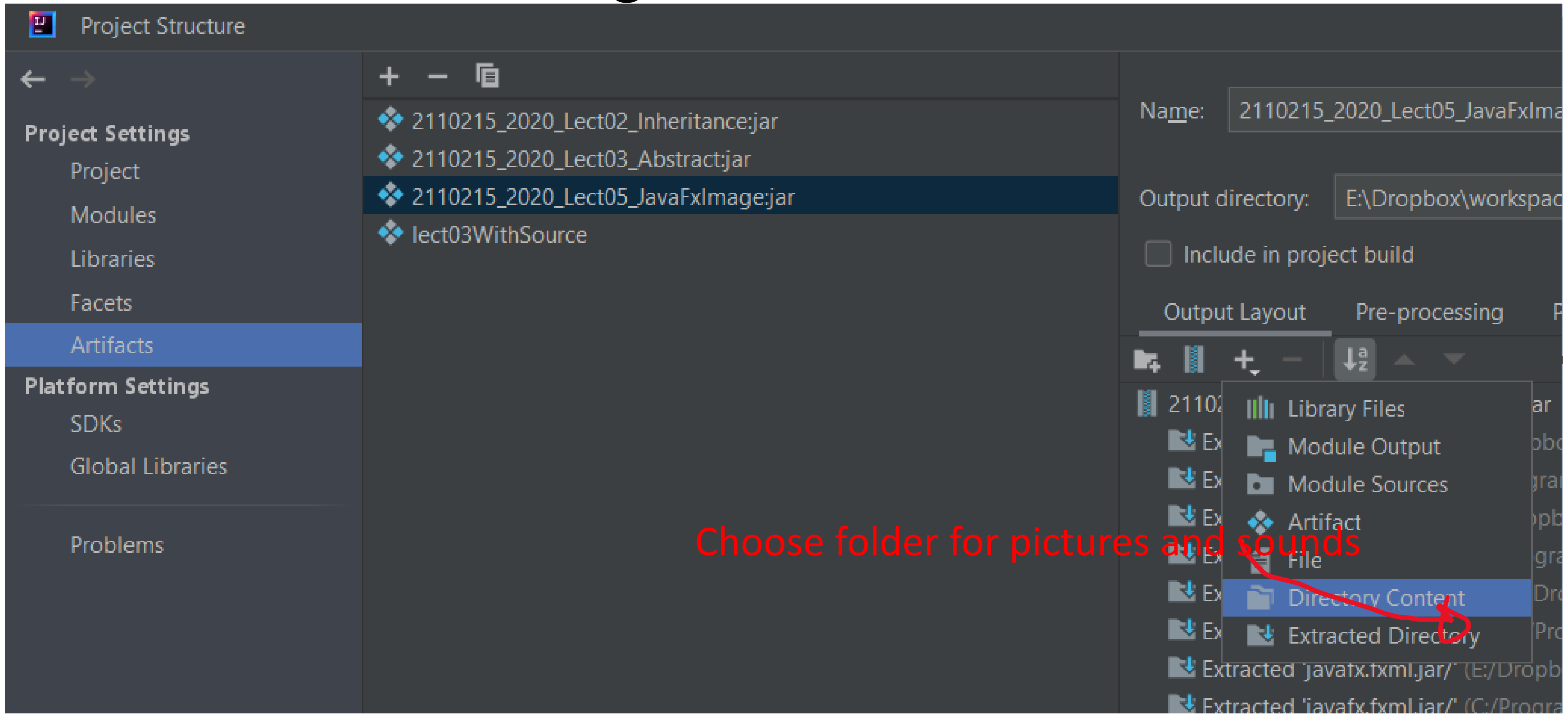
Open run/debug tool window when started

Now you can run any file!



How to export JavaFx JAR file (with resources)!

- Do it just like a normal export.
- But after choosing the main class ->



Project Structure

Project Settings
- Project
- Modules
- Libraries
- Facets
- Artifacts

Platform Settings
- SDKs
- Global Libraries

Problems

- 2110215_2020_Lect02_Inheritance:jar
- 2110215_2020_Lect03_Abstract:jar
- 2110215_2020_Lect05_JavaFxImage:jar
- lect03WithSource

Name: 2110215_2020_Lect05_JavaFxIma

Output directory: E:\Dropbox\workspac

Include in project build

Output Layout | Pre-processing

- Library Files
- Module Output
- Module Sources
- Artifact
- File
- Directory Content
- Extracted Directory

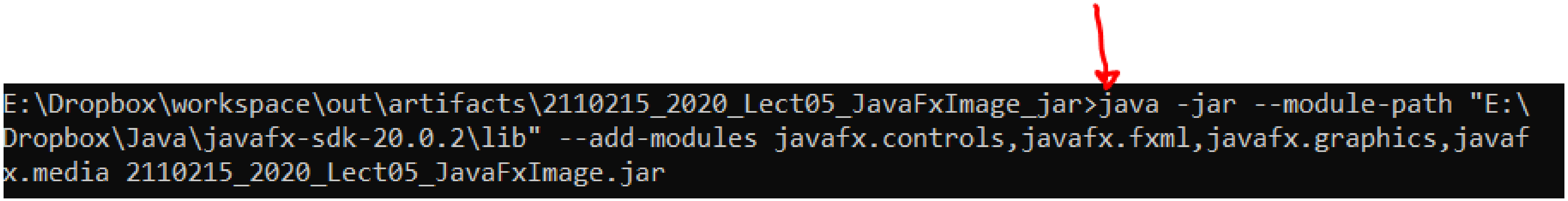
Extracted 'javafx.fxml.jar/' (E:/Dropb

Extracted 'javafx.fxml.jar/' (C:/Progra

Choose folder for pictures and sounds

After this step, you can build artifact normally.

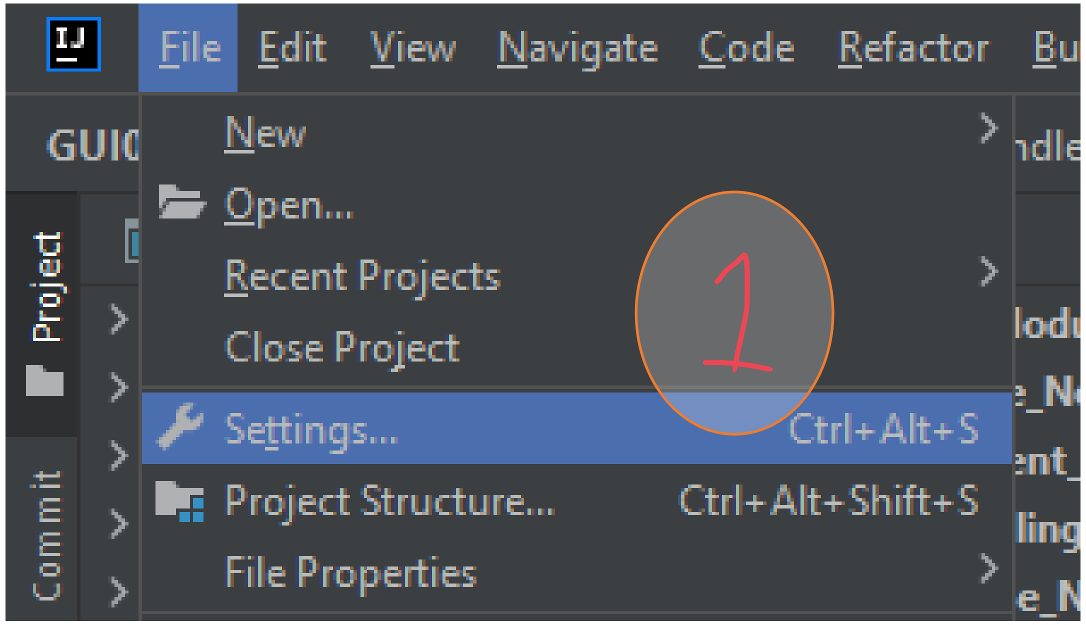
Then you can run JAR file

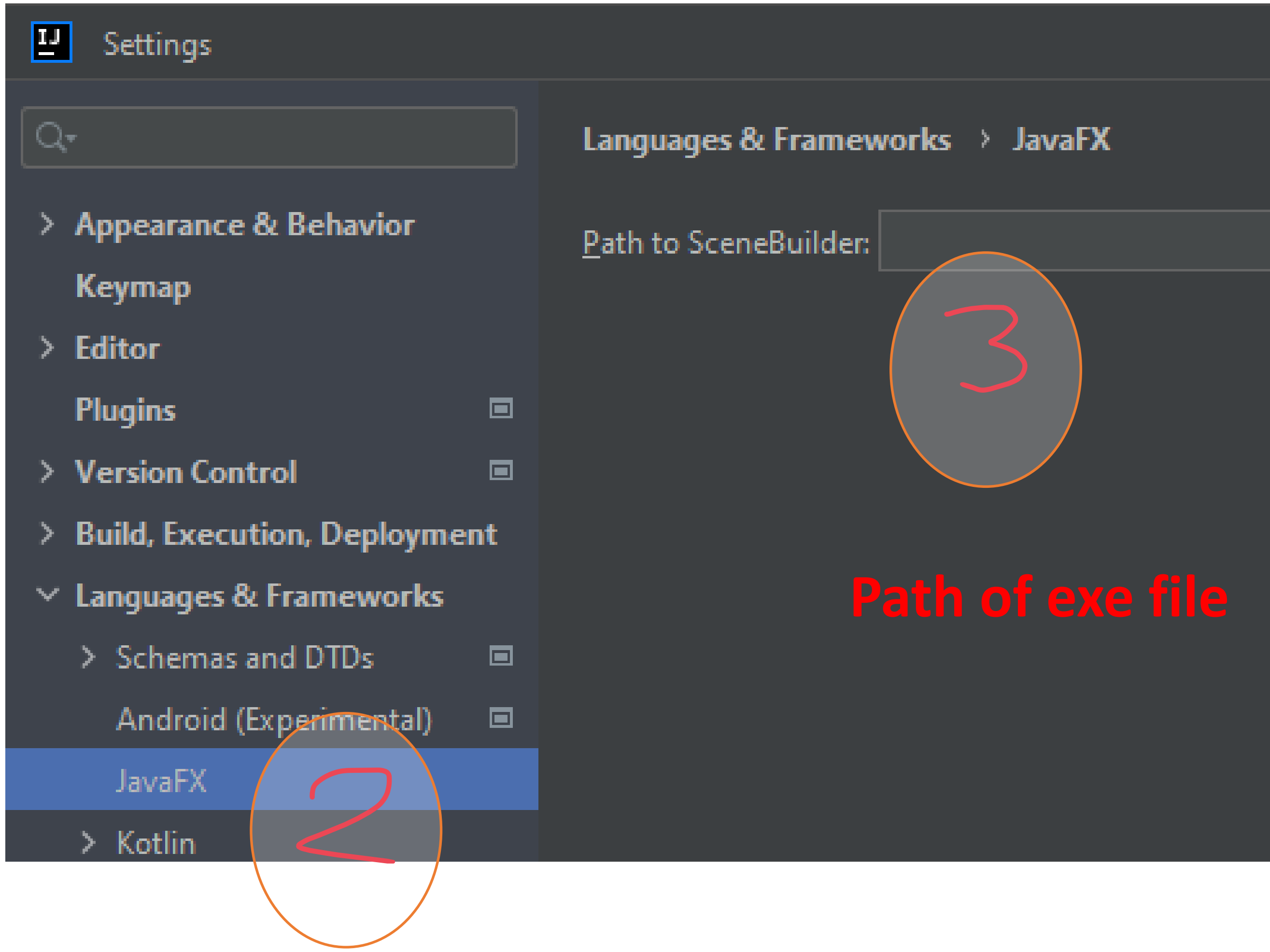


```
E:\Dropbox\workspace\out\artifacts\2110215_2020_Lect05_JavaFxImage_jar>java -jar --module-path "E:\
Dropbox\Java\javafx-sdk-20.0.2\lib" --add-modules javafx.controls,javafx.fxml,javafx.graphics,javaf
x.media 2110215_2020_Lect05_JavaFxImage.jar
```

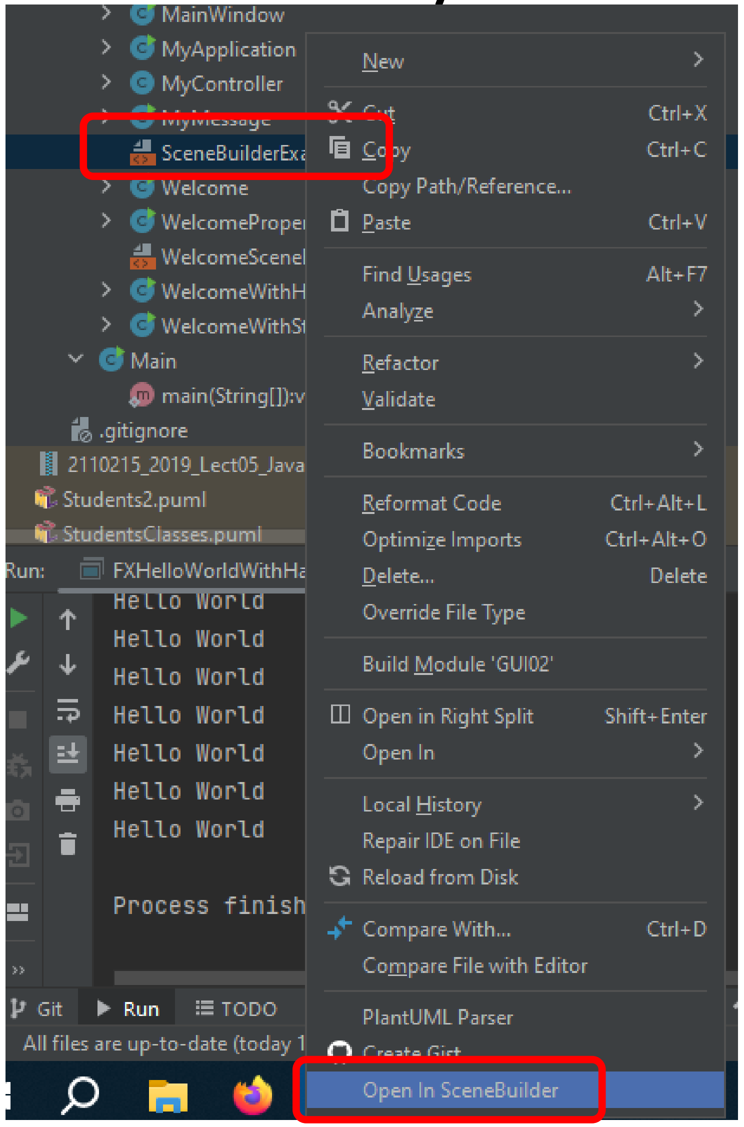
SceneBuilder

- After SceneBuilder is installed,

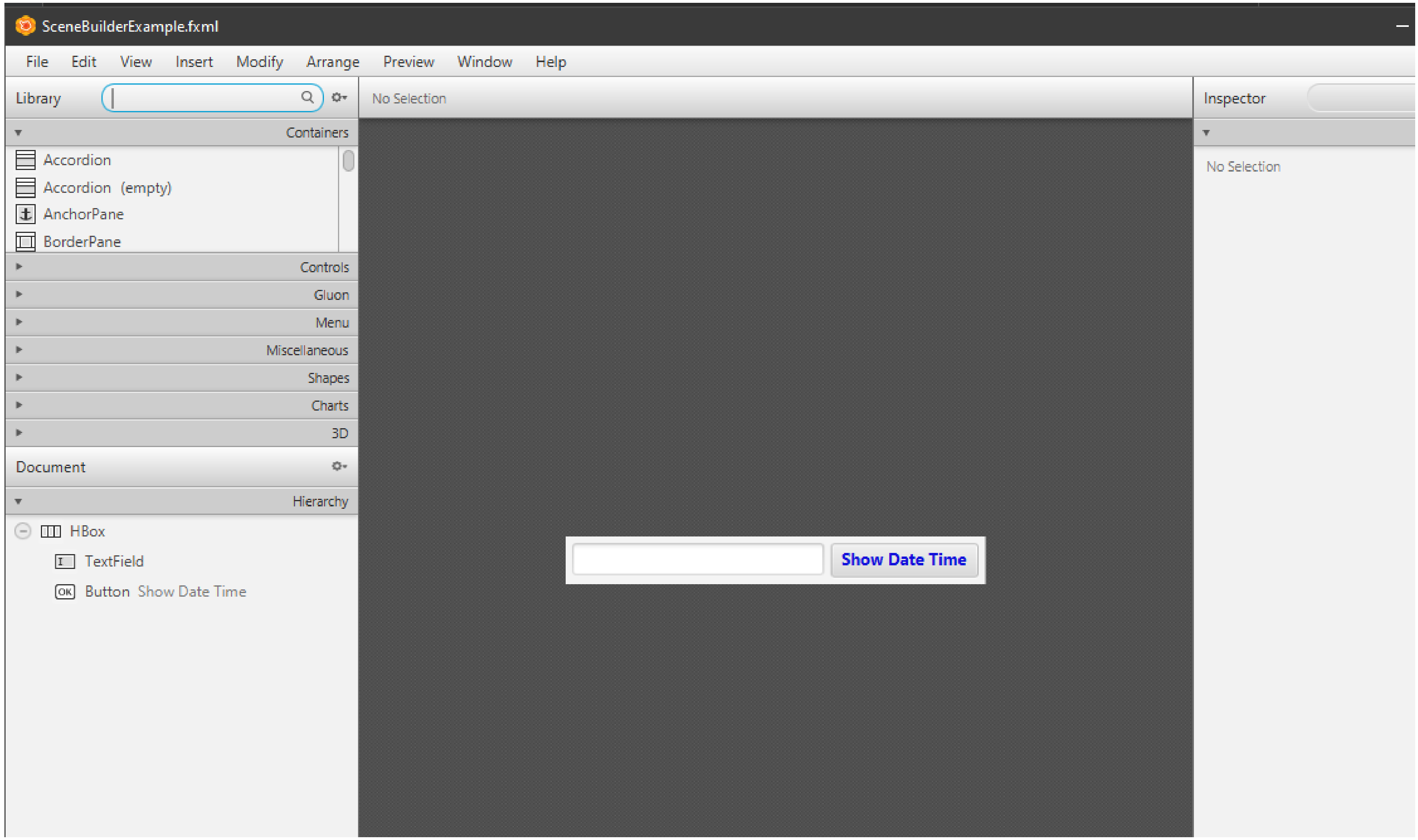




Now you can open fxml file in SceneBuilder



```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World

Process finish
```



How to export runnable jar with SceneBuilder file

- Just export like any other JavaFX module.
- As long as you included .fxml file with class files, it will run.