

installation

download either [ide.7z] or [ide.zip] depending on whether u have 7-zip/winrar

extract the files somewhere

navigate to the [bin] folder

the exe is [idea64.exe] (64-bit) or [idea.exe] (32-bit)

i recommend that you create a shortcut and move the shortcut to the desktop or somewhere else

scroll to the end and accept (unless u actually read it and find an issue)

don’t send usage statistics (unless you want to, idk if it makes things slower)

u should now see something like this



setting up the development environment

click this button



on the left side choose [java] (unless u want python)

don’t choose [empty project] bc idk if you’ll have sdk issues or something

click the [new] button in the top right to tell idea where ur sdk is

mine was [c:\program files\java\jdk1.8.0\_191], not [jre1.8.0\_191] or a [bin]

under additional libraries and frameworks don’t check python

(unless you’ll use python and java at the same time)

if ur using python, configure ur python sdk

mine was [c:\program files\python37\python.exe] so it’s the exe file

click [next] at the bottom

idk what [create project from template] is so don’t check it (seems useful maybe)

click [next]

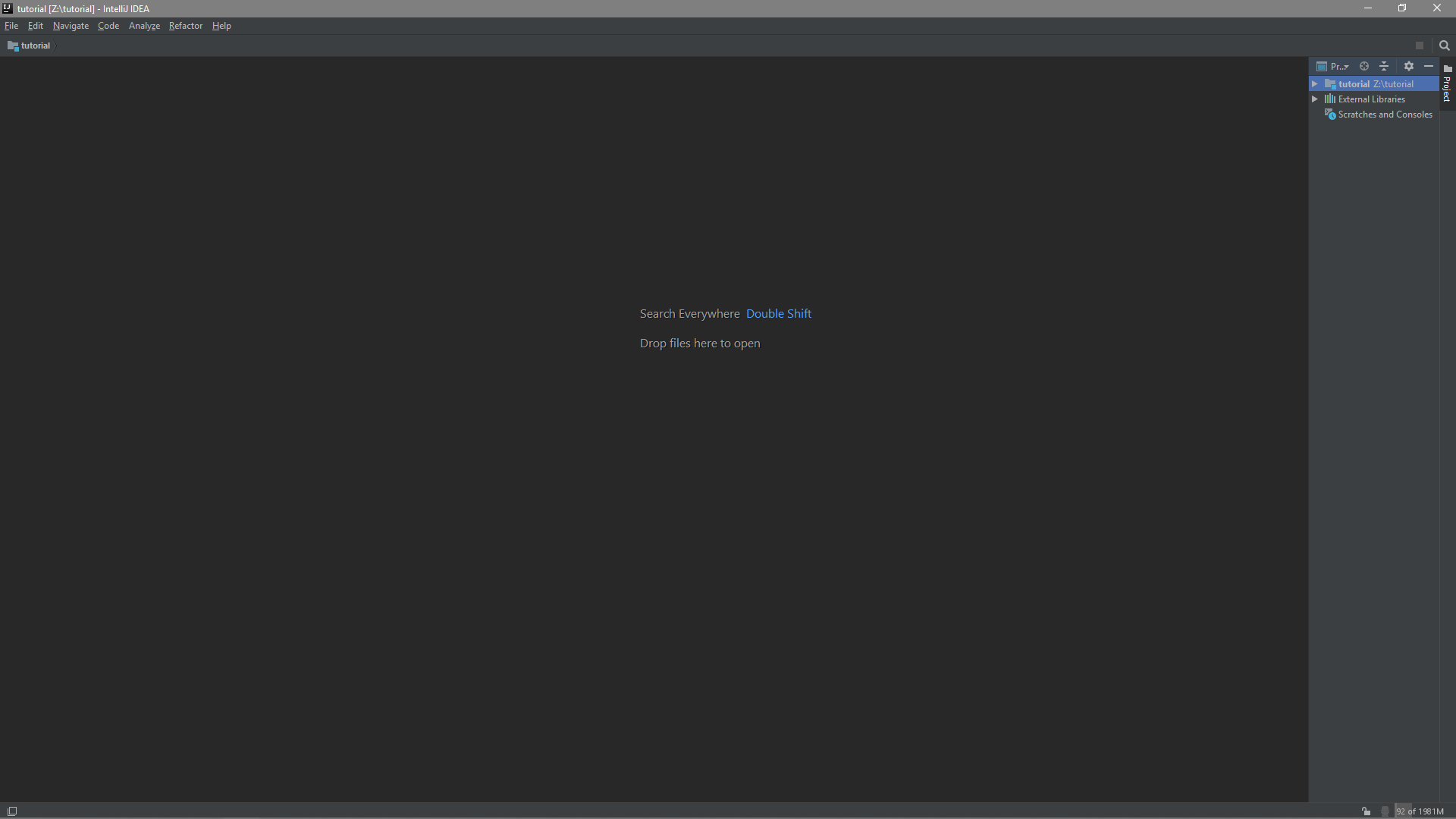
in project location choose a folder to store the project

i recommend making the new folder on the desktop or somewhere accessible

the project name doesn’t have to be the same as the folder name but imo it’s better if it’s the same

click [finish]

now ur thing should look something like this (tutorial is the project name)



making a file

on the sidebar, expand the main folder

right click [src]

make a [new] [java class]

type the name of the file (it automatically adds the .java extension)

click [ok]

now u can start typing stuff

u can right click the file or folder and click [show in explorer]

other [new] options:

[file] creates a blank file, u have to type an extension if u want one

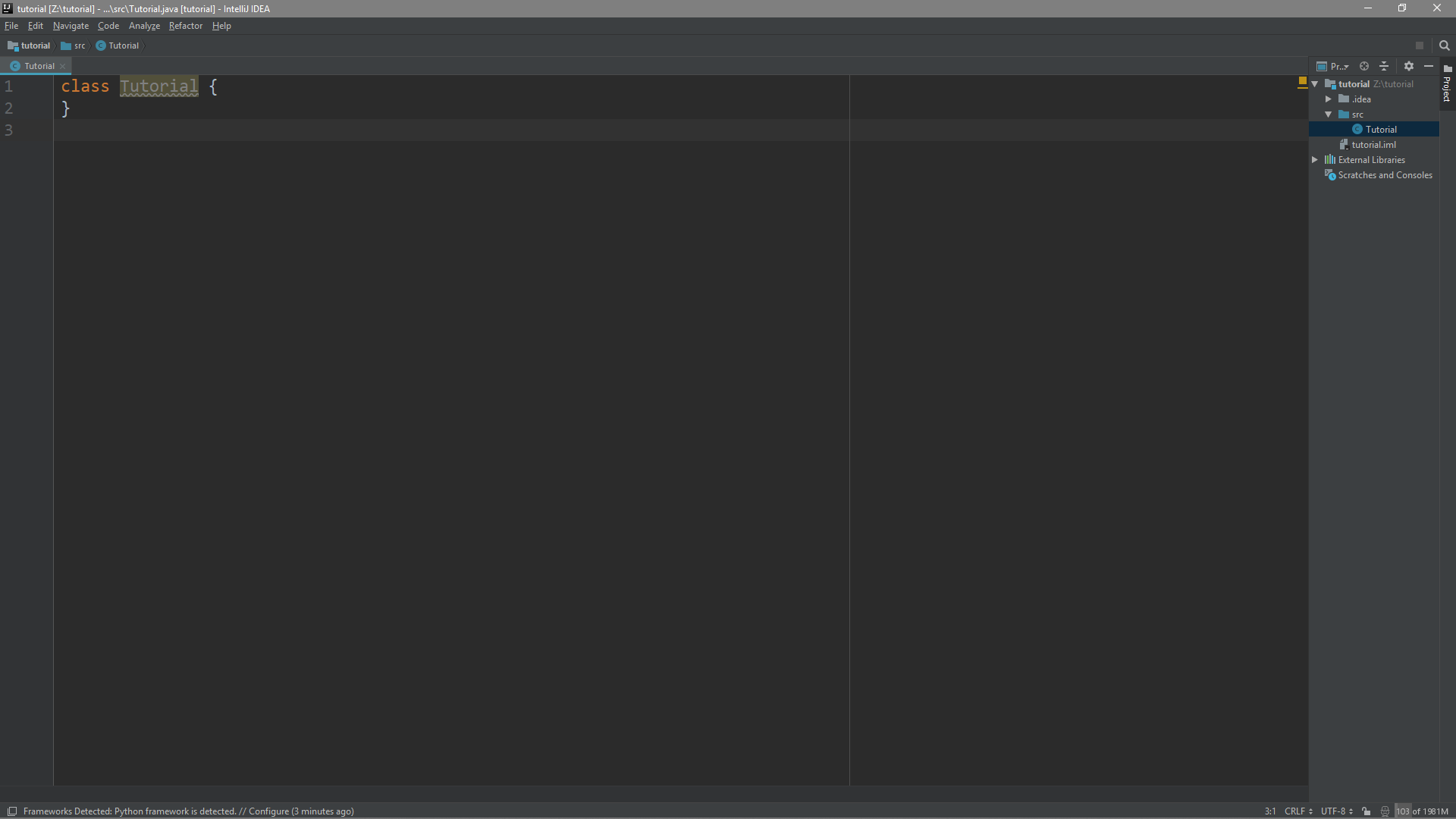
i see this as a way to make sample input files if they’re not provided

[python file] makes a new [.py] file for python

the rest are templates, i’ll cover those later

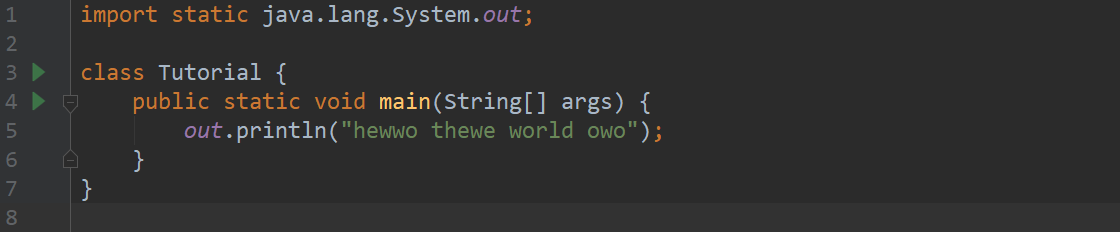
you can also press f5 to create a new file from template

now it looks like this



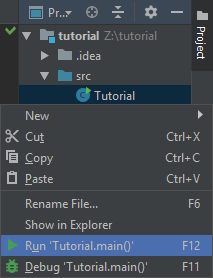
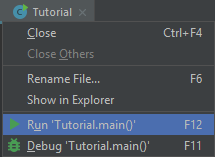
running and debugging

let’s say that u typed up ur stuff



because there’s now a main method, this file is runnable

run button locations:

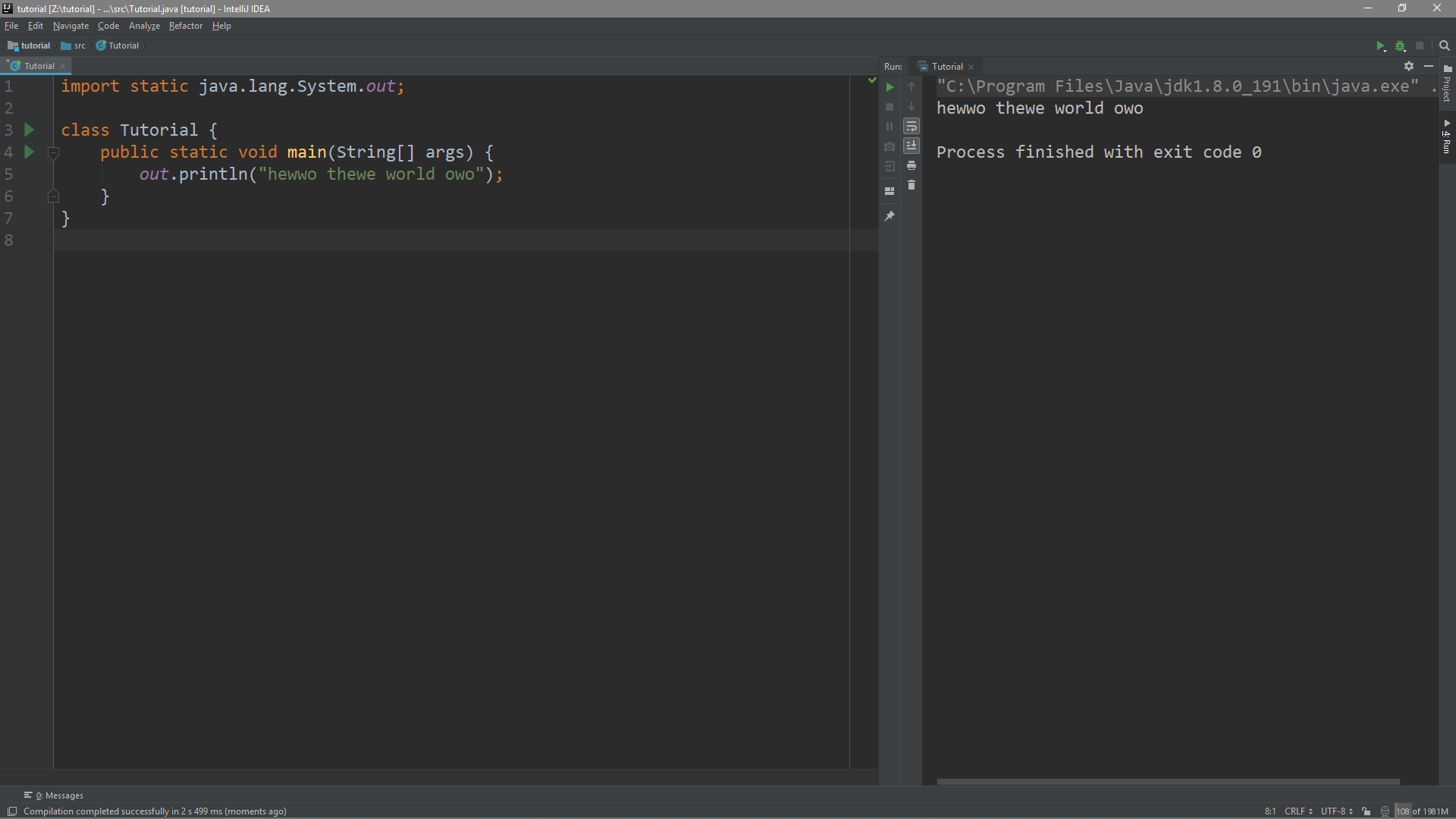


but i prefer pressing f12

u can also debug, if u dont know how to debug look it up

idk why but u need to allow it through the firewall or else u cant debug

now it looks like this

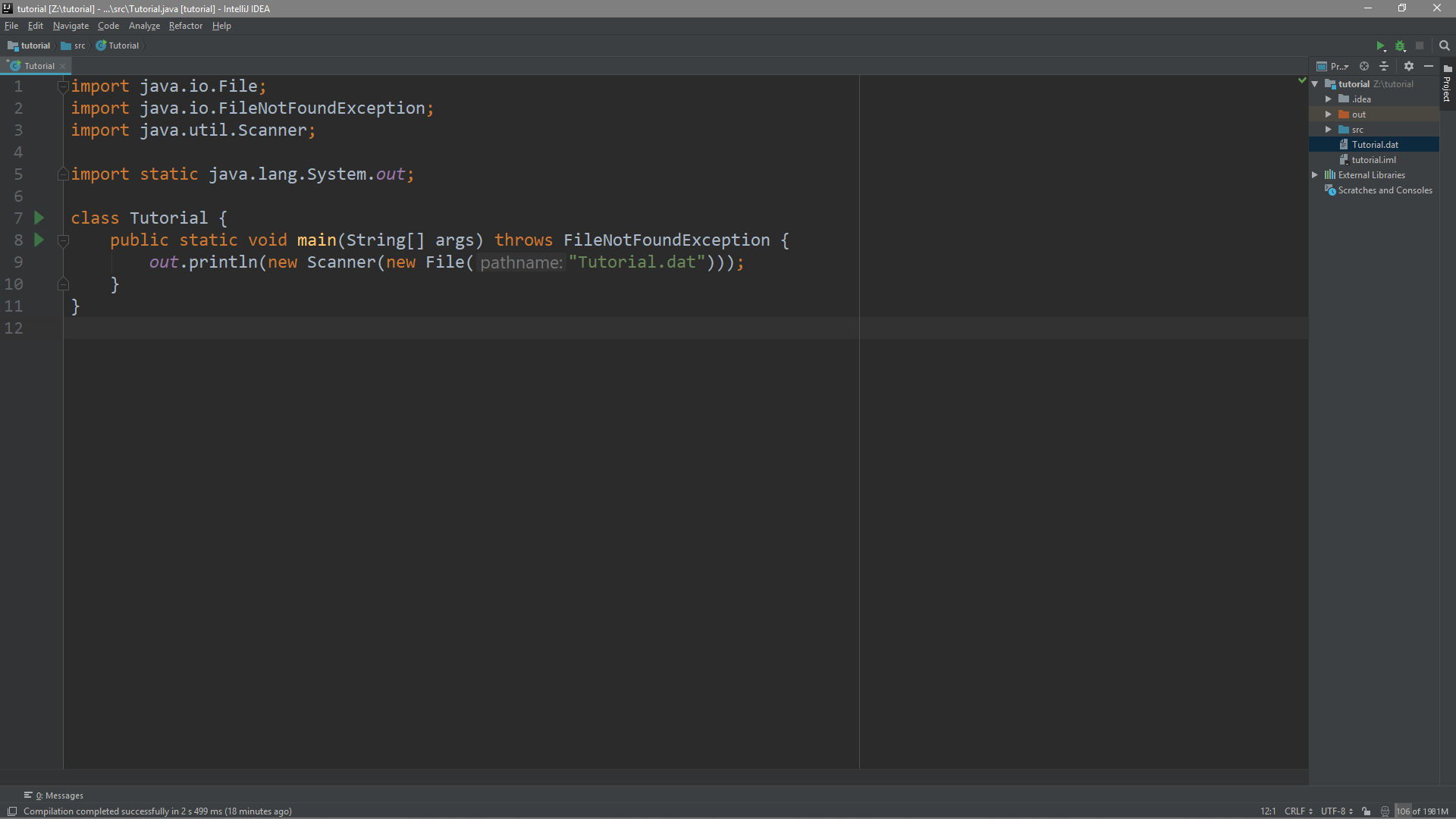


i should’ve named it tutowial.java

input files

they have to be in the root folder (not [src])

if u look at this image you should understand what i mean



note the location of [tutorial.dat] on the right panel

file templates

[looper]: for problems with inputs like this:

The first line of input will contain a single integer n that indicates the number of <something>.

(eleven out of twelve of the uil state 2012 problems)

[printer]: for problems where you’re just doing ascii art

pls make more templates for common patterns

live templates

basically prewritten code snippets (methods, symbols, etc)

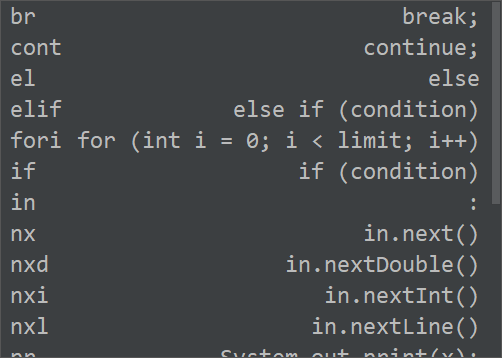
insertion method a:

just type the shortcut and hit space or tab

for example, typing “prl” [space] will expand to System.out.println();

insertion method b:

[ctrl]+[ins] will show a list of things u can insert at the caret position

 left: shortcut ; right: what it expands to

then u just select one of them to insert

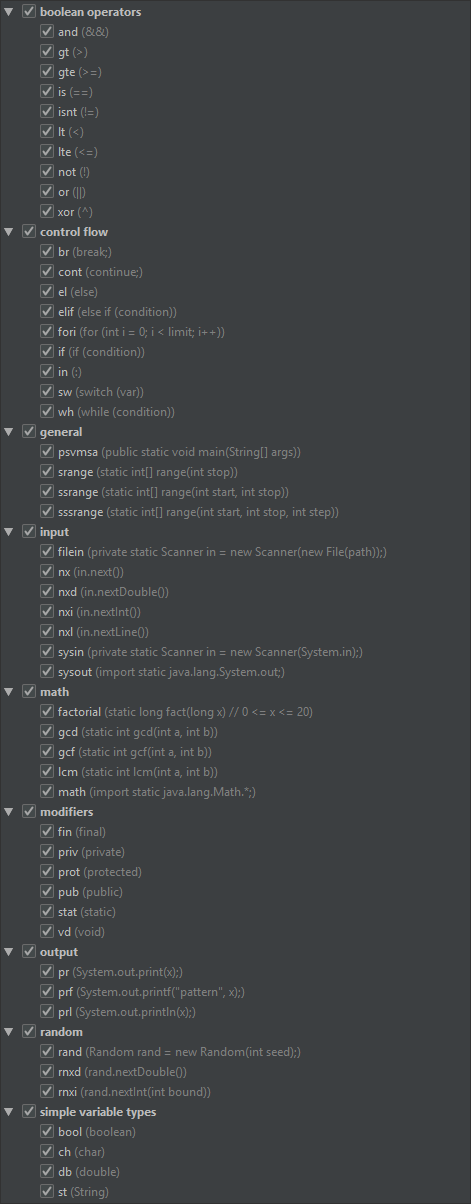
to view all the live templates go to settings > editor > live templates

i also put a list on the next page

lots of them are inspired by python bc python is cool

to disable/enable certain live templates use the checkboxes

i think that this feature has lots of potential for saving coding time



notable keymappings

f1: reformat code (retabs and respaces and stuff)

f2: inspect code (can delete unused code, otherwise not that useful)

f3/f4: previous/next error (jump between syntax errors)

f5: new file from template

f6: rename variable/file

f7/f8: step into/step over (debugging, google if u dont know)

shift+f8/shift+f7: step out (debugging)

f9: settings

alt+f9: finds a setting or an action (searches all of the ide’s functions)

f10: edit configurations (i dont think you’ll need this)

f11: debug current file

f12: run current file

shift+f11,shift+f12: stop running/debugging

ctrl+tab,ctrl+shift+tab,ctrl+f4,ctrl+shift+t: next tab, previous tab, close tab,

reopen tab (basically like google chrome)

alt+(p,r,d): switches to project panel, run panel, or debug panel

shift+escape: like alt+tab but inside the ide

shift+delete: delete line

alt+delete: join line (combines current and next lines)

alt+enter: show suggestions

ctrl+(w,shift+w,a): select more, select less, select all

ctrl+(f,r): find, replace

ctrl+(z,shift+z): undo, redo (ctrl+y doesn’t do anything)

ctrl+(x,c,v,shift+v): cut, copy, paste, paste raw

ctrl+s: save all

alt+leftclick: rectangular selection

alt+(i,j,k,l): simulates arrow keys

alt+arrowkeys: moves some stuff around

ctrl+(l/r arrowkeys,alt+(j,l)): move caret to previous/next word

shift+(arrowkeys,alt+ijkl): move caret with selection

ctrl+shift+(l/r arrowkeys,alt+(j,l): combines the two above

experiment with [ctrl,shift,alt]+[arrowkeys,alt+(ijkl)] bc lots of combos

there are some more so go to settings > keymap to see all

yeah that’s it, pls suggest improvements if u want

also idk if this guide was good, if it was bad then oops