

Forum:[Hack Forums](#)

Thread: [Basic Java help \(will rep\) cannot find symbol even though class exists](#)

Shxtdown - February 7, 2019, 12:31 pm

So as part of my assignment I was given two .java source files and a separate class file, my professor didn't give me the java source code for the class file as this is part of the assignment. However I'm having a basic problem even creating an instance of this said class in IntelliJ (which in my case is the GameArena class).

Here's an image of what I'm describing:

As you can see, the class I'm trying to create an instance of does exist - however IntelliJ thinks it doesn't. How do I fix this? I've tried so many different ways and it's getting incredibly frustrating lol - especially since this isn't even one of the tasks... it's the first step. I've tried rebuilding the project, invalidating... creating root files or w/e. I'm stuck here - I'm a beginner to Java so that's why I've been stuck on this for so long.

I need a way for IntelliJ to be able to recognise it, I've tried moving it into the src directory but that doesn't work either (probably not good practice too). Does anyone know a solution to this?

GenericSkid - February 7, 2019, 4:05 pm

It'd actually be easier to just invoke javac directly for such a small assignment.

```
javac -cp <prof's-class> <your-java>
```

Anyways:

<https://stackoverflow.com/questions/31342689/how-do-i-use-classes-from-another-project-in-intellij-idea>

Linxy - February 7, 2019, 4:08 pm

Make sure they're all in the same package.

Also, GameArena.class isn't in your source folder.

Caliban - February 7, 2019, 4:12 pm

Click on your GameArena.class File and select "Refactor" -> "Rename File". Then change it to "GameArena.java" instead of ".class". Now it should find it.

Also yeah move it into the same package that could be the problem

Shxtdown - February 7, 2019, 4:34 pm

(Feb 7, 2019 05:08 PM)Linxy Wrote: [\[CITE\]](#) Make sure they're all in the same package.

Also, GameArena.class isn't in your source folder.

Putting the class file in the same package doesn't work - also I'm not sure that's a good idea considering it's not actually a Java source file. Basically my professor gave me this class file on purpose, I'm not supposed to have the .java file (we're practicing with encapsulation).

(Feb 7, 2019 05:12 PM)Caliban Wrote: [\[CITE\]](#) Click on your GameArena.class File and select

"Refactor" -> "Rename File". Then change it to "GameArena.java" instead of ".class". Now it should find it.

Also yeah move it into the same package that could be the problem
It's supposed to be a .class file, making it .java sorta defeats the purpose. Putting it in the same package didn't solve the problem.

Caliban - February 7, 2019, 4:46 pm

(Feb 7, 2019 05:34 PM)Shxtdown Wrote: [\[CITE\]](#) Putting the class file in the same package doesn't work - also I'm not sure that's a good idea considering it's not actually a Java source file. Basically my professor gave me this class file on purpose, I'm not supposed to have the .java file (we're practicing with encapsulation).

It's supposed to be a .class file, making it .java sorta defeats the purpose. Putting it in the same package didn't solve the problem.

Well if its a .class file it means it has already been compiled and it usually will be in the "target" folder in IntelliJ then. If you want to be able to instantiate it it needs to be a .java file, unless you use the Reflection API (see

<https://stackoverflow.com/questions/36452834/using-reflection-to-instantiate-a-java-class-from-a-class-file>), but im not sure if thats what your prof wants you to do.

Shxtdown - February 7, 2019, 6:57 pm

(Feb 7, 2019 05:46 PM)Caliban Wrote: [\[CITE\]](#) Well if its a .class file it means it has already been compiled and it usually will be in the "target" folder in IntelliJ then. If you want to be able to instantiate it it needs to be a .java file, unless you use the Reflection API (see <https://stackoverflow.com/questions/36452834/using-reflection-to-instantiate-a-java-class-from-a-class-file>), but im not sure if thats what your prof wants you to do.

Ah I see - thanks man. That's odd though because this is what my prof said:

Notice you only have the .class files (not the java source code) for GameArena. This is because I'm delivering this to you as a module, as we discussed in the lectures. It's sealed. You can't get in. It's my responsibility, and I don't trust you (yet!) to not break it if you did have the code. You can however, make instances of this class, `[i][i]` and call the methods I've written for you.

Not sure how this is entirely possible?

Caliban - February 7, 2019, 7:20 pm

(Feb 7, 2019 07:57 PM)Shxtdown Wrote: [\[CITE\]](#) Ah I see - thanks man. That's odd though because this is what my prof said:

Notice you only have the .class files (not the java source code) for GameArena. This is because I'm delivering this to you as a module, as we discussed in the lectures. It's sealed. You can't get in. It's my responsibility, and I don't trust you (yet!) to not break it if you did have the code. You can however, make instances of this class, and call the methods I've written for you.

Not sure how this is entirely possible?

The only way i know of to instantiate a class from .class files is with the `.newInstance()` method of the java reflection api. And yes the reflection API can break encapsulation, you can even modify access modifiers lmao.

Maybe what he wants you to do is simply run the class with `javac classname then java classname` in cmd, which will create an instance in the sense that it will get loaded into the JVM/memory, not in

the sense that you can "new" it up in your application.