

Questions and Exercises

These questions and exercises is an opportunity to see what you've learnt from the lecture as well as practice the new things we've been talking about. In other words, these questions and exercises are completely optional but it's recommended to do them. In the end of the document you will find the answers to the questions as well as possible solutions to the exercises, note that one can solve an exercise in different ways. There will also be some suggestions about what one could code if one want to continue with some more advanced things. These suggestions will not come with a possible solution and might include things that haven't been covered in the lecture.

Question 1

What's wrong with the following code?

```
Level[] levels = new Level[3];  
levels[0].start();
```

Question 2

Why should one use packages? And what should one remember to do when using them?

Question 3

Which are the 3 named scope modifiers? What's the difference between them?

Exercises

Since this lecture is mostly an extension of the last one there's no specific exercises here. What you're supposed to do is to modify the exercises from the last lecture so they use proper scopes and packages. When that is done, take a look on the further explorations below.

Further explorations

Expand the game example shown in the lecture (the files can be found on the lecture page). Add so the player can have different items that can be found in the world. Expand the world by adding different areas that you can move between. Also add monsters that you can find and fight against which will drop loot when killed. Allow the player to use armor and weapons.

Answers and solutions

Answer to Question 1

The given code will crash the program. A `NullPointerException` will be thrown. This is because we're referring to the method `start()` of `null`, since we haven't assign `levels[0]` a value. What we would have to do to make this work would be to create a `Level` and store it in the array. Something like the following code would work:

By Vidar Swenning

```
Level[] levels = new Level[3];  
levels[0] = new Level();  
levels[0].start();
```

Answer to Question 2

Packages are good to use to organize a lot of code. Instead of having a ton of classes just mixed you can organize it by putting them in different themed packages, which also results in you making sub folders for your files. When using different package you need to remember to import the classes from other packages. Also, if you change the package of a class to be TestPackage you will need to put the java file in a sub folder called TestPackage.

Answer to Question 3

The three different named one are private, protected and public. There are however also the default one which is used if you don't use any of the named ones. When using private in front of a field or a method it can only be accessed from within the same class. Protected allows you to access them from another class if the that class is in the same package. Public methods and fields can be accessed from anywhere.