


New Java Class

Java Class

 This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder:

Package:

Enclosing type:

Name:

Modifiers: public package private protected
 abstract final static
 none sealed non-sealed final

Superclass:

Interfaces:

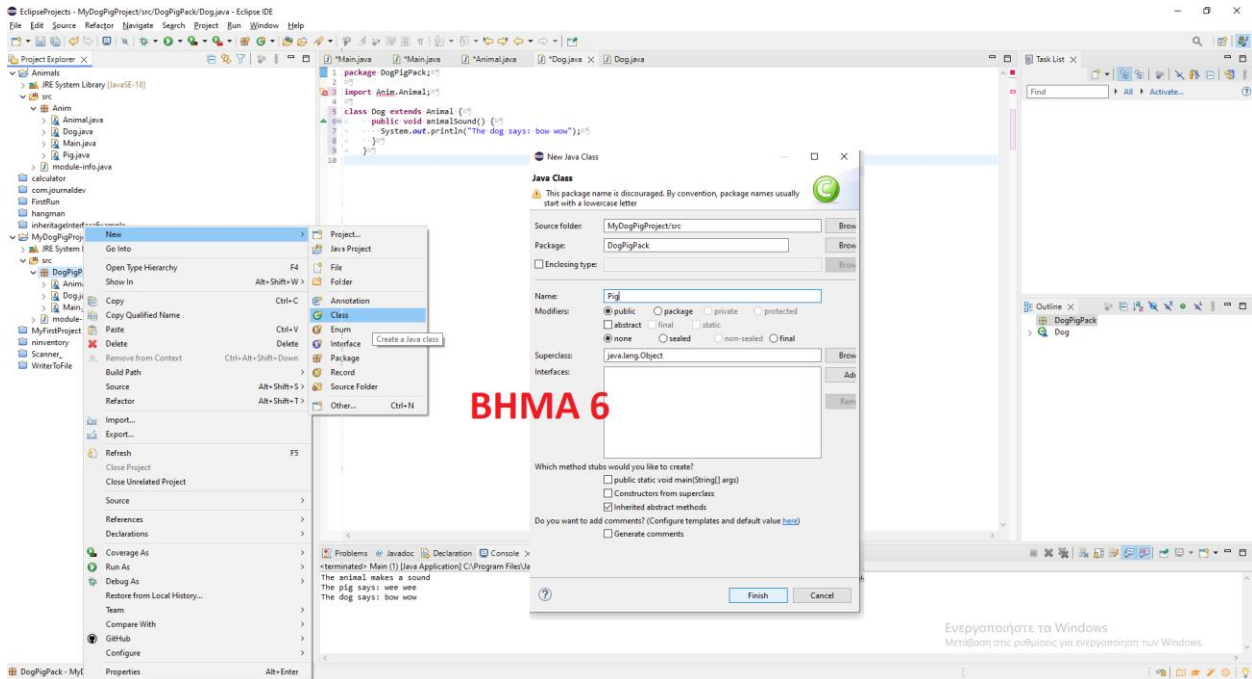
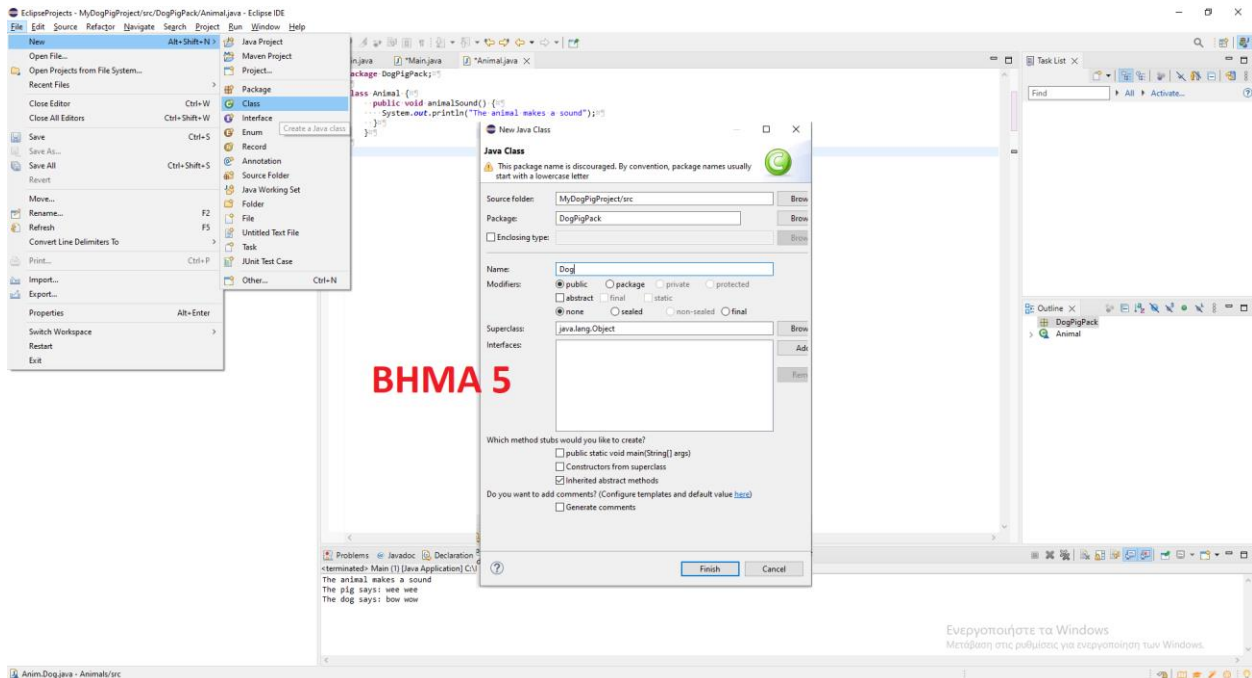
BHMA 4

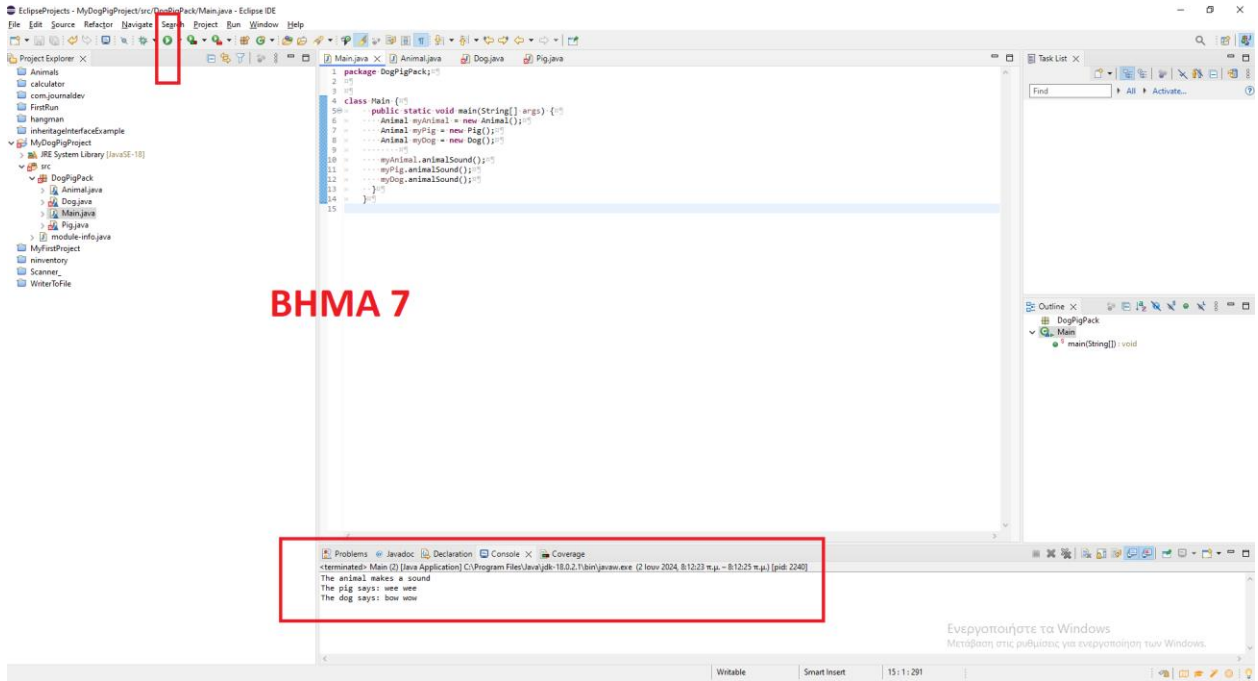
Which method stubs would you like to create?

public static void main(String[] args)
 Constructors from superclass
 Inherited abstract methods

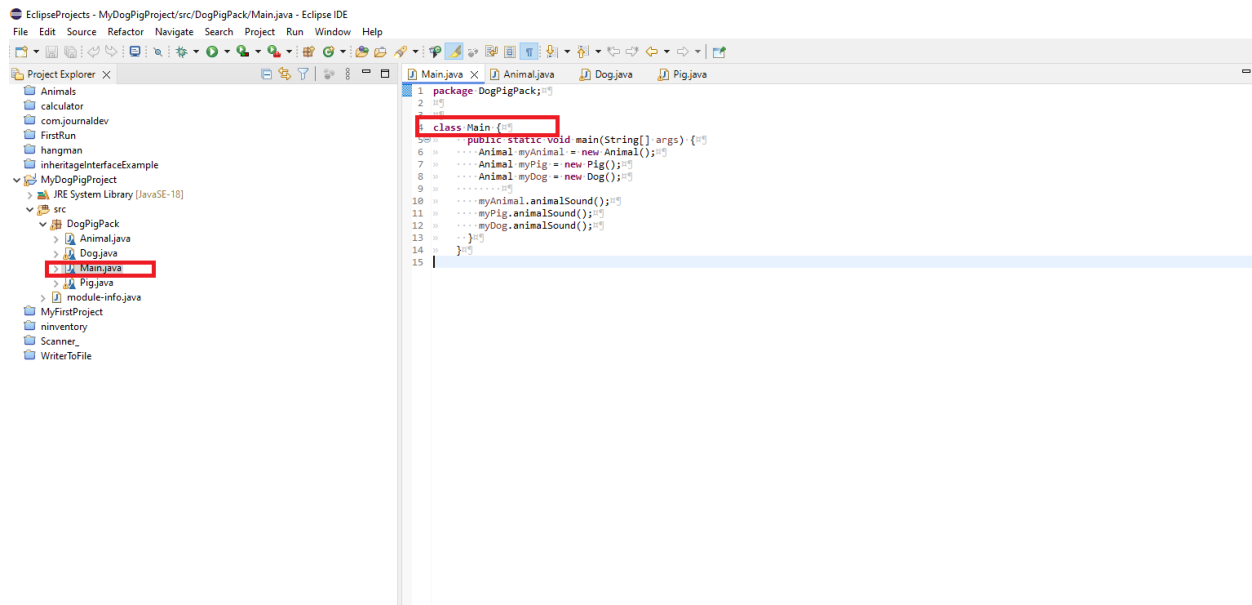
Do you want to add comments? (Configure templates and default value [here](#))

Generate comments



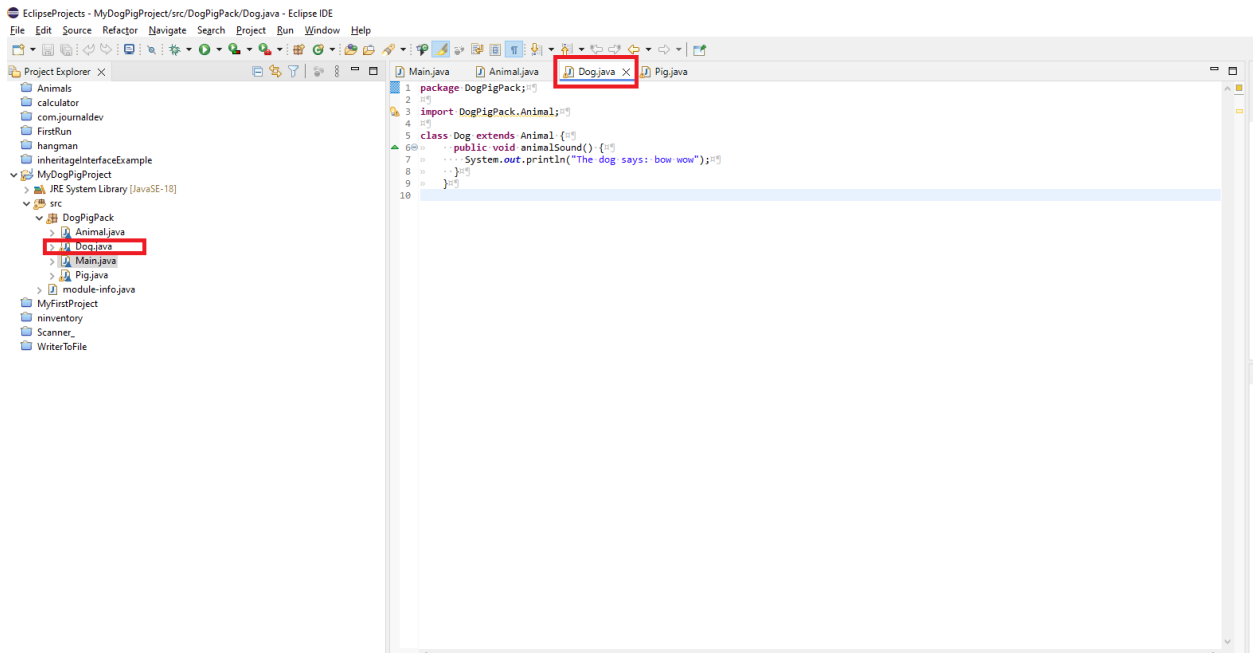
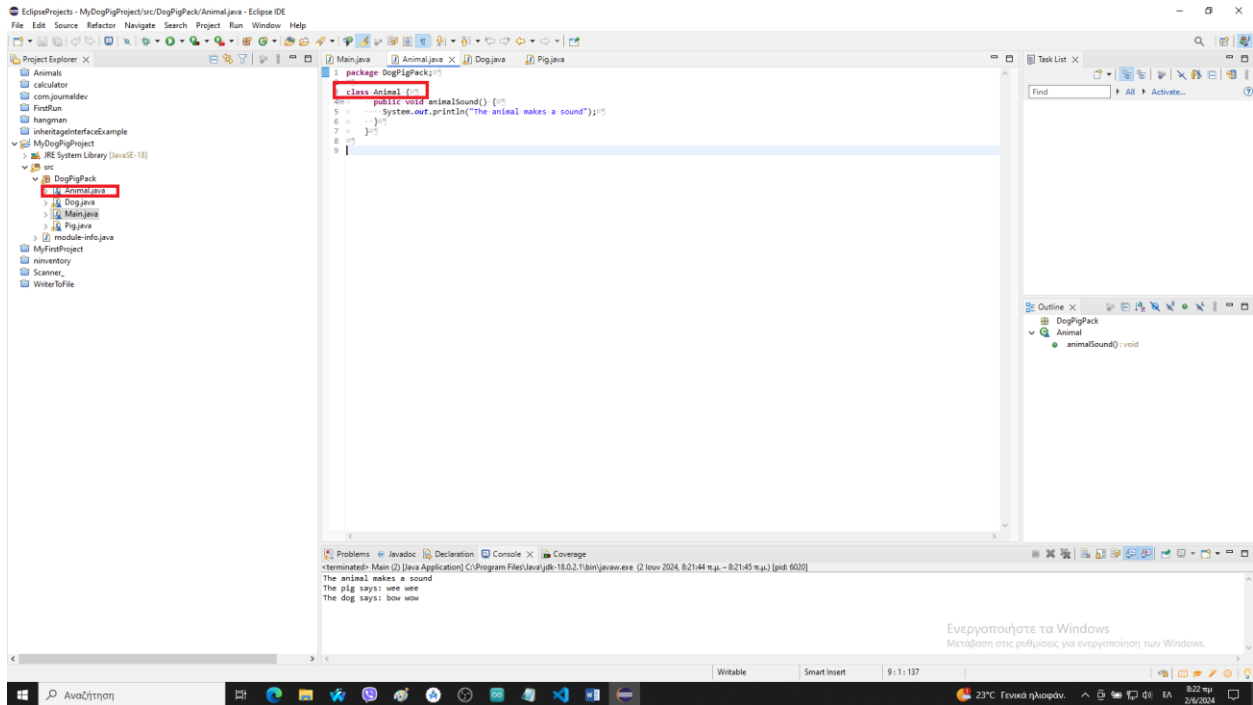


ΠΩΣ ΦΑΙΝΟΝΤΑΙ ΟΙ CLASSES ΜΑΣ.



The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays a project named 'MyDogPigProject' with a source folder 'src' containing subfolders 'DogPigPack', 'Animal.java', 'Dog.java', and 'Pig.java'. The 'Main.java' file is selected and highlighted with a red box. The main editor window shows the code for 'Main.java' with the following content:

```
1 package DogPigPack;
2
3 class Main {
4     public static void main(String[] args) {
5         Animal myAnimal = new Animal();
6         Animal myPig = new Pig();
7         Animal myDog = new Dog();
8         myAnimal.animalSound();
9         myPig.animalSound();
10        myDog.animalSound();
11    }
12 }
13
14
15
```



- Animals
- calculator
- com-journaldev
- FirstRun
- hangman
- inheritageInterfaceExample
- MyDogPigProject
 - JRE System Library [JavaSE-10]
 - src
 - DogPigPack
 - Animal.java
 - Dog.java
 - Main.java
 - Pig.java
 - module-info.java
 - MyFirstProject
 - nInventory
 - Scanner
 - WriterToFile

```
1 package DogPigPack;
2
3 import DogPigPack.Animal;
4
5 class Pig extends Animal {
6     public void animalSound() {
7         System.out.println("The pig says: wee wee");
8     }
9 }
```