# Simple Java Program

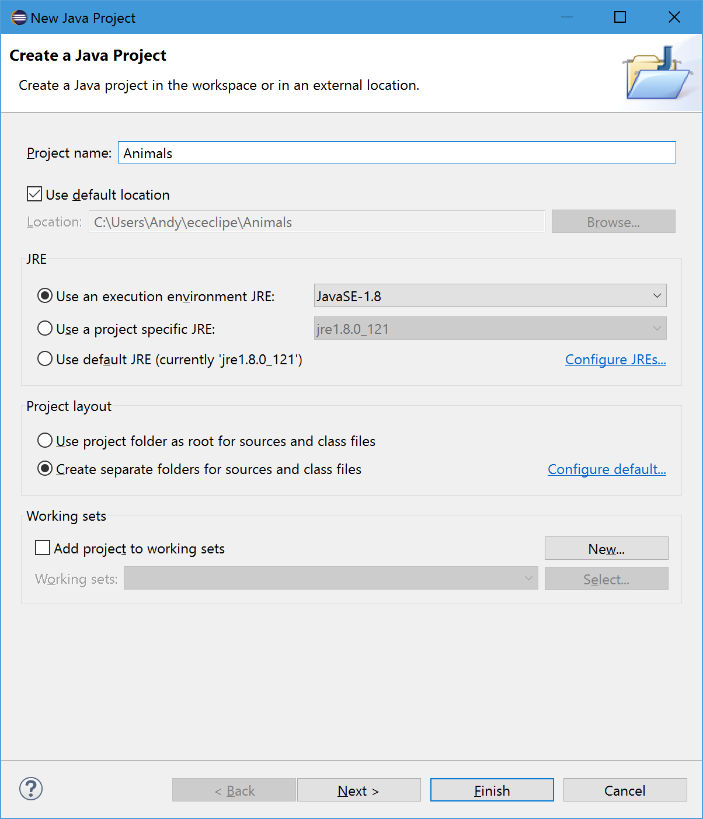
This exercise shows some of the principles of Object Oriented Programming (OOP). OOP is used very widely in professional software development, pretty much every language supports it to some extent, including Python and JavaScript.

OOP is useful because it allows developers to create code that feels closer to “real life” than other types of program “paradigms” (this just means “ways to organise code”, such as functional programming or logic programming).

OOP means that you create “classes” and “objects”. Objects have “attributes” and “methods”. This is like the real world. For example: You are a person. In OOP terms, this makes you an object. Your class would be “Person”. You have attributes (such as height, age, hair colour, and so on). You can also do things such as eat, drink, sleep, play computer games, and so on. These actions are called methods in OOP.

Obviously, people are very complicated things. You have hundred, thousands, or millions of attributes and methods. Therefore, in this exercise we’re going to create a very simple animal, a sheep.

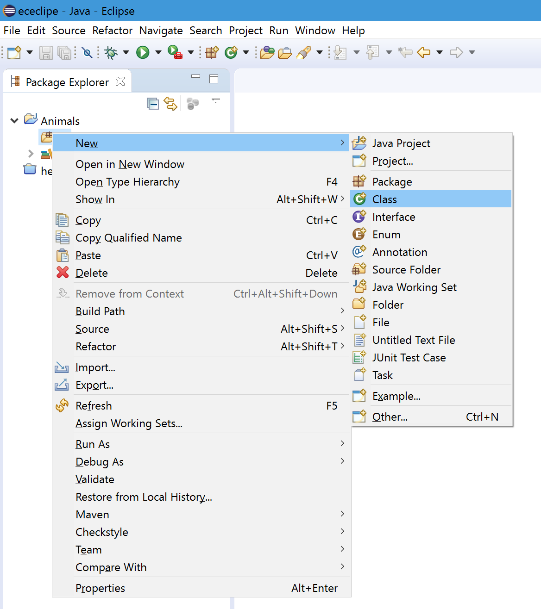
Create a new Java Project and call it “Animals”. File -> Project -> New Java Project



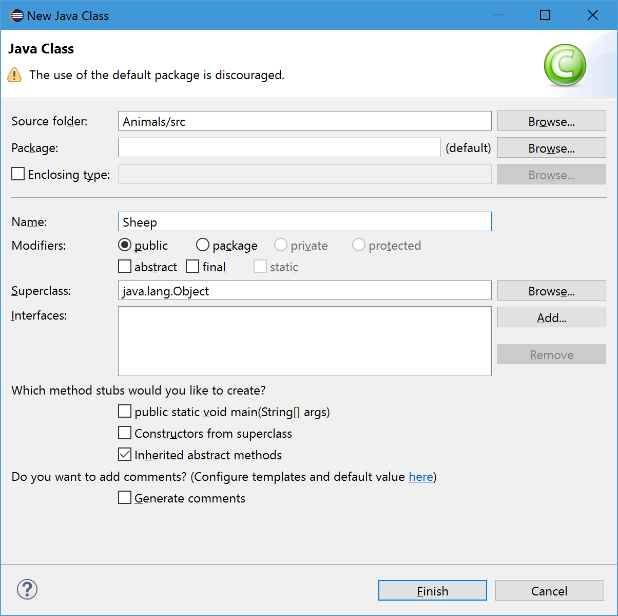
Type the name of the project and click Finish.

Make sure any other projects are closed (right click on them and select “Close project”) so Animals is the only open project.

In Java, each class is typically held inside its own file. The name of the file should match the name of the class. Let’s create a new class called Sheep:



Right click the “src” directory, then New -> Class.



Ignore the warning about the use of the default package, enter the class name as shown and click Finish. The source code for Sheep.java will be opened on the screen. Create two attributes and a method on the Sheep class:

**public** **class** Sheep {

**public** String name;

**public** **int** weightInKg;

**public** Sheep(String name, **int** weightInKg) {

**this**.name = name;

**this**.weightInKg = weightInKg;

}

**public** **void** sayBaa() {

System.***out***.println("Baa!");

}

}

You can see that this gives every sheep a name and a weight in kilograms. There’s a special method, called Sheep, which is called a *constructor*. This is used to set the attributes. That’s enough for us to demonstrate a sheep for now, so now create a new separate class that will create some *instances* of sheep, i.e. some sheep objects.

Create a new class called Program (File -> New -> Class). Fill it in as shown here:

**public** **class** Program {

**public** **static** **void** main(String[] args) {

Sheep dolly = **new** Sheep("Dolly", 30);

Sheep shaun = **new** Sheep("Shaun", 10);

}

}

You can see that this creates 2 *instances* of type “Sheep”. This means you’ve created 2 Sheep objects.

To call the *sayBaa* method, you put the name of the object, followed by a dot, followed by the method. Like this:

**public** **static** **void** main(String[] args) {

Sheep dolly = **new** Sheep("Dolly", 30);

Sheep shaun = **new** Sheep("Shaun", 10);

dolly.sayBaa();

shaun.sayBaa();

}

Make the changes to add the two new lines and run the code (Ctrl + F11). If you’ve got it right you should see:

Dolly: Baa!

Shaun: Baa!

You can see how OOP works. You *invoked* the sayBaa method on Dolly and Shaun and they both did something.

You can change attributes of an object in a similar way to calling methods:

**public** **static** **void** main(String[] args) {

Sheep dolly = **new** Sheep("Dolly", 30);

Sheep shaun = **new** Sheep("Shaun", 10);

dolly.sayBaa();

shaun.sayBaa();

System.***out***.println("Dolly weighs " + dolly.weightInKg + " kilograms");

dolly.weightInKg = 40;

System.***out***.println("Dolly now weighs " + dolly.weightInKg + " kilograms");

}

Methods can also modify attributes. Go back to Sheep.java and add the following methods:

**public** **void** eatGrass() {

System.***out***.println("Yum!");

weightInKg++;

}

**public** **void** poop() {

System.***out***.println("That's better!");

weightInKg--;

}

And update Program.java:

**public** **static** **void** main(String[] args) {

Sheep dolly = **new** Sheep("Dolly", 30);

Sheep shaun = **new** Sheep("Shaun", 10);

dolly.sayBaa();

shaun.sayBaa();

System.***out***.println("Dolly weighs " + dolly.weightInKg + " kilograms");

dolly.eatGrass();

dolly.eatGrass();

dolly.eatGrass();

dolly.poop();

System.***out***.println("Dolly now weighs " + dolly.weightInKg + " kilograms");

}

You should see output like this:

Dolly: Baa!

Shaun: Baa!

Dolly weighs 30 kilograms

Yum!

Yum!

Yum!

That's better!

Dolly now weighs 32 kilograms

Now try adding other methods to your Sheep and call the methods on Shaun, instead of Dolly.