**Class and object**

Object-oriented is the soul of Java and also an important theory and practice foundation for object-oriented design. To understand object-oriented, we will first need to learn about class and object - these two fundamental concepts, which will be covered in this class.

**Class**

The saying goes "things of one kind come together", which well defines the meaning of class: things with similar features will be grouped into one class.

*A class is a template for describing objects with the same attributes and methods.*

For example, Zhangsan and Lisi share the same attributes of man, such as speaking, thinking and fighting etc. In this case, we can abstract the similarities between Zhangsan and Lisi to create a class: Person.

Through above example, you may find out that programming is a process learning from nature and simulating nature, which will be futher explained later in this course.

**Grammar**

modifier **class** **class** **name** {

**class** **variables**

**member** **variables**

**constructor** **methods**

**class** **methods**

**member** **methods**

...

}

Introduction:

* Modifier: It is used to modify class access permissions, which will be further explained later.
* Class: This is the keyword to indicate and define a class.
* Class name: A class name refers to the name of a class, which should have explicit meanings. It should follow the naming conventions of Java identifiers. Camel Case should be used where capital letters are used to delimit the word parts.
* Class varialbes: Class variable is also declared in class and outside of the method body. But you should declare it is of the static class. External access can be done directly through the class.
* Member variables: Member variable is the variable defined in class and outside of the method body.It is instantiated (allocating the memory) when creating an object. It can be accessed by the method in a class and certain statements.
* Constructor methods: please find detailed introduction below.
* Class methods: As the method of class, class method is defined in class. It is modified by static keyword. External access can be done directly through the class.
* Member methods: It is defined in class and will be instantiated when creating an object. External access must be done through specific instances.

*For example*

**public** **class** **Person** {

**public** String name; *// member variable*

**private** **int** age; *// member variable*

*// constructor method*

**public** **Person**(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

*// member method*

**public** **int** **getAge**() {

**return** age;

}

*// member method*

**public** **void** **setAge**(**int** age) {

**this**.age = age;

}

}

**Constructor methods**

Instead of calling manually, constructor methods can be automatically executed when the class is instantiated. For this method, you can add some initialization during instantiation. You can take the first method in above example for reference: public Person(String name, int age) {...}

*For constructor methods, please be noted the below points:*

* The name of the construction method should be the same with that of the class. There is no return value for this method.
* Each class has a construction method. If there is no construciton method explicitly defined for a class, Java compiler will provide a default construciton method for this class.
* Constructor method cannot be called by display.

**Object**

Object is the instantiated result of class. Instantiation is the process of creating an object. Take above examples as reference, Zhangsan and Lisi are the instantiated result of the class - Person.

Generally there are three steps to create an object:

* Declaration: declare an object with object name and object type.
* Instantiation: create an object with the keyword of new.
* Initialization: When you use "new" to create an object, constructor methods will be called to initialize the object.

*Grammar:*

**class** **name** **object** **name** = **new** constructor mehtod of **class**

*For example*

Person zhangsan = **new** Person("zhangsan", 18);

Note: By using Person, a new object "zhangsan" is created with the name of zhangsan and age of 18.

**Access member variable and member method:**

You can access member variable and member method through instantiated object. The grammar is as below:

object name.variable name

object name.method

For example, you can access the name and age of zhangsan through the object of zhangsan.

zhangsan.name;

zhangsan.getAge();

**Package**

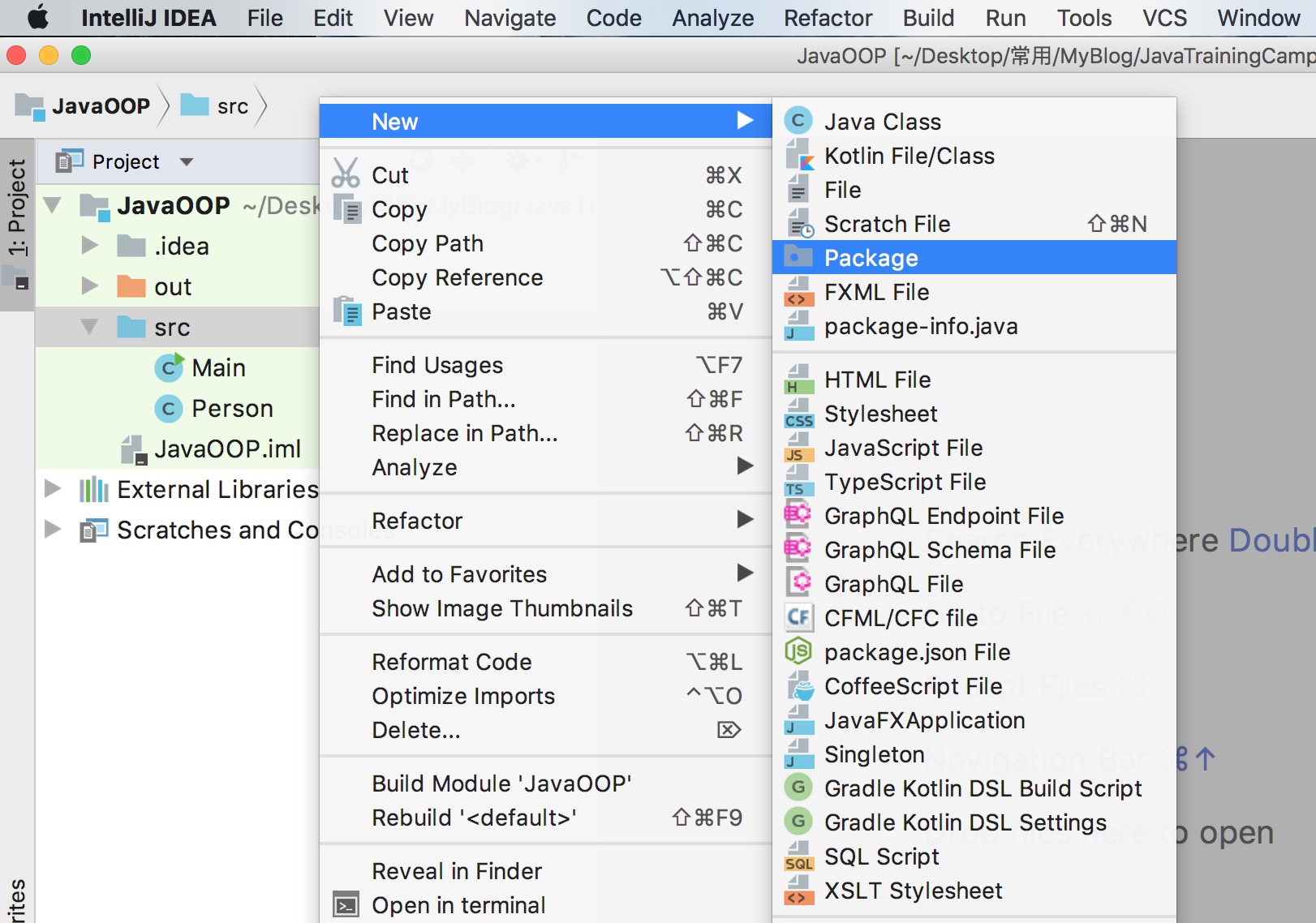
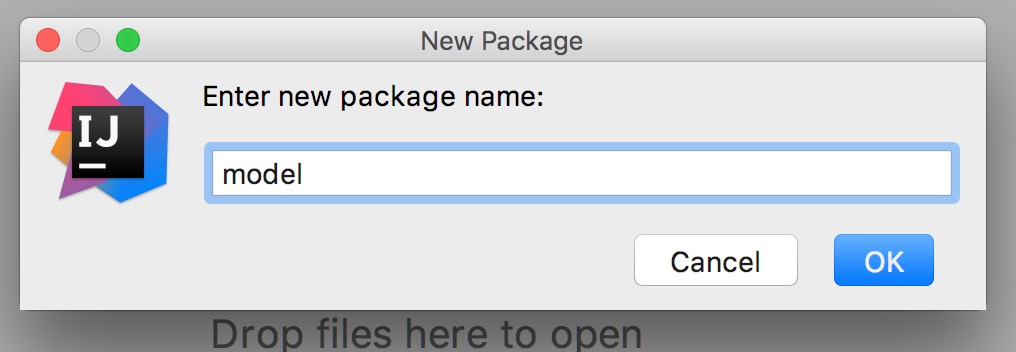
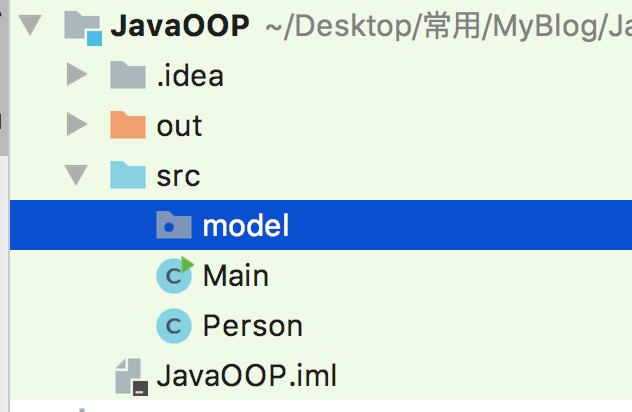
Package is the main approach in Java language to differentiate classes. It is of great significance to differentite different classes during Java programming as thousands of classes will be created wherein. That's why the concept of package is introduced. Package is similar to the folder in operating systems where the same class of files are put under one folder for easy management. In addition, package can also help with the problem of files with the same name. Though classes with the same name cannot be differentitated within one package, we can create different packages to hold them. From this perspective, package will help create an independent naming space.

**Create a package**

To create a package, you should first name a package with explicit meaning. Its naming convention is: all letters are in lower case, with two adjacent words divided by an underline.

*For example:*

Create a package in the source directory: model

* 
* 
* 

Create a class under the model package: Student

**package** model;

**public** **class** **Student** {

**private** String name;

**private** **int** studentNumber;

**public** String **getName**() {

**return** name;

}

**public** **void** **setName**(String name) {

**this**.name = name;

}

**public** **int** **getStudentNumber**() {

**return** studentNumber;

}

**public** **void** **setStudentNumber**(**int** studentNumber) {

**this**.studentNumber = studentNumber;

}

}

You will find that added in the class is a declaration of package: package model;

**import**

In Java, if you have a fully qualified name, including a package name and class name, Java compiler can easily locate the source code or class. Import statements are used as an appropriate path to help compiler find a certain class.

*For example:*

If you want to refer to the Student class defined in above model package:

**import** model.Student;

**public** **class** **Main** {

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

Student student = **new** Student();

student.setName("zhangsan");

System.out.println(student.getName());

}

}

**For more information on class, object and package, please visit**[**here**](https://docs.oracle.com/javase/tutorial/java/concepts/index.html)