## I，Non-Access Modifiers

For ****classes****, you can use either final or abstract:

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| final | The class cannot be inherited by other classes |
| abstract | The class cannot be used to create objects (To access an abstract class, it must be inherited from another class. |

For ****attributes and methods****, you can use the one of the following:

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| final | Attributes and methods cannot be overridden/modified |
| static | Attributes and methods belongs to the class, rather than an object |
| abstract | Can only be used in an abstract class, and can only be used on methods. The method does not have a body, for example ****abstract void run();****. The body is provided by the subclass (inherited from). |
| transient | Attributes and methods are skipped when serializing the object containing them |
| synchronized | Methods can only be accessed by one thread at a time |
| volatile | The value of an attribute is not cached thread-locally, and is always read from the "main memory" |

## Static

A static method means that it can be accessed without creating an object of the class

public class Main {

static void myStaticMethod() {

System.out.println("Static methods can be called without creating objects");

}

// Main method

public static void main(String[ ] args) {

myStaticMethod(); // Call the static method

II，

To use a class or a package from the library, you need to use the import keyword:

import *package*.*name*.*Class*; // Import a single class

import *package*.*name*.\*; // Import the whole package

# III, Java Exceptions

The throw statement allows you to create a custom error.

The throw statement is used together with an ****exception type****. There are many exception types available in Java: ArithmeticException, FileNotFoundException, ArrayIndexOutOfBoundsException, SecurityException, etc:

public class Main {

static void checkAge(int age) {

if (age < 18) {

throw new ArithmeticException("Access denied - You must be at least 18 years old.");

}

else {

System.out.println("Access granted - You are old enough!");

}

}

public static void main(String[] args) {

checkAge(15); // Set age to 15 (which is below 18...)

}}