Java Basics

Jianjun wu

Content

Overview

Overview

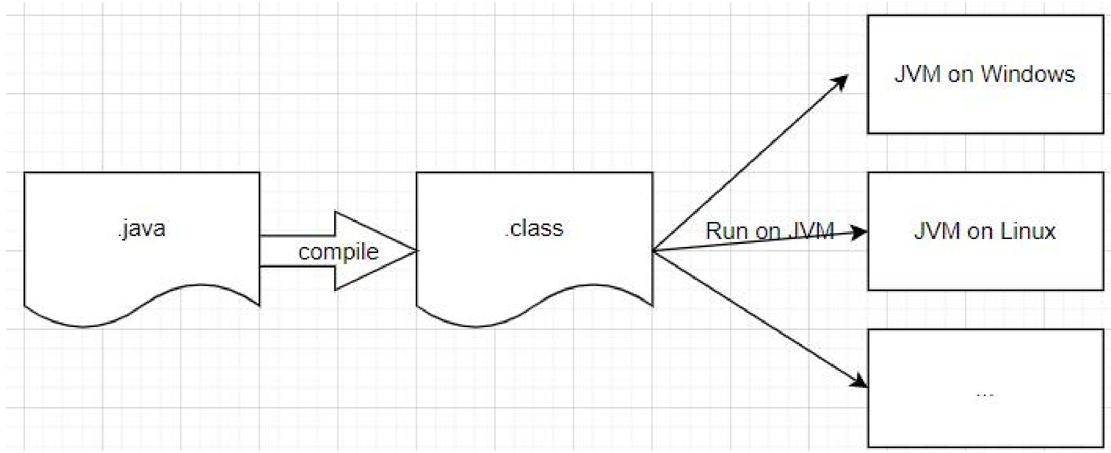
- What’s Java?
  - is a programming language, similar to c++.
  - developed by James Gosling and his team at Sun Microsystems.
    - James Gosling is known as the Father of Java.
  - cross-platform running is the obvious advantage compare to C++.
    - write once run anywhere
  - first released in 1995 by Sun Microsystems.
  - Java 1.8 is released in 2014。

Overview

- What’s Java?
  - some terminologies:
    - JVM: short for Java Virtual Machine. Our code must run on JVM. Every OS has a different JVM.
    - Bytecode: generated from source code by comiler and saved as .class file, can be executed by JVM.
    - JRE : short for Java Runtime Environment. it includes JVM and some core libraries.
    - JDK: short for Java Development Kit. In addition to JRE, compiler and other development tools are also included.

Overview

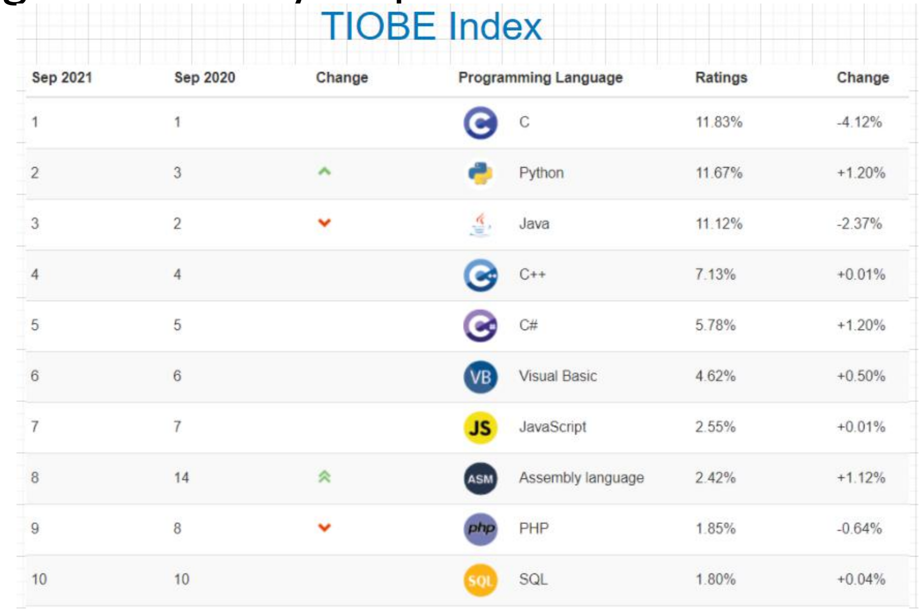
- What's Java?
  - the steps to develop a Java program:
    - write your source codes with kinds of tools.
    - compile your code by compiler , for example javac, to generate bytecode.
    - run bytecode on JVM.



Overview

- Where is Java used?
  - Java is one of the popular programming languages and always top ranks.
  - Some Applications
    - Android： considered as the official language.
    - BigData: the backbone for developing Big Data.

      many famous bigdata sorftwares are writen in Java:
      - Hive,
      - HDFS,
      - Flink,
      - etc.


TIOBE Index

Sep 2021	Sep 2020	Change	Programming Language	Ratings	Change
1	1		C	11.83%	-4.12%
2	3	▲	Python	11.67%	+1.20%
3	2	▼	Java	11.12%	-2.37%
4	4		C++	7.13%	+0.01%
5	5		C#	5.78%	+1.20%
6	6		Visual Basic	4.62%	+0.50%
7	7		JavaScript	2.55%	+0.01%
8	14	▲▲	Assembly language	2.42%	+1.12%
9	8	▼	PHP	1.85%	-0.64%
10	10		SQL	1.80%	+0.04%

Overview

- What does Java have?
  - In a nutshell, Java is a huge aircraft carrier and the learning course may be quite long if you want to harness it at your will.
  - one possible learning procedure for backend development is:
    - step1: learn java basics. the real one is far beyond this tutorial.
    - step2: use components. such as MySQL, Redis, etc.
    - step3: use frameworks. such as Spring, Dubbo, Zookeeper,etc.
    - step4: understand underlying mechanism, such as Netty, JVM, etc.
    - step5: read excellent open projects, such as elasticsearch,
    - ...

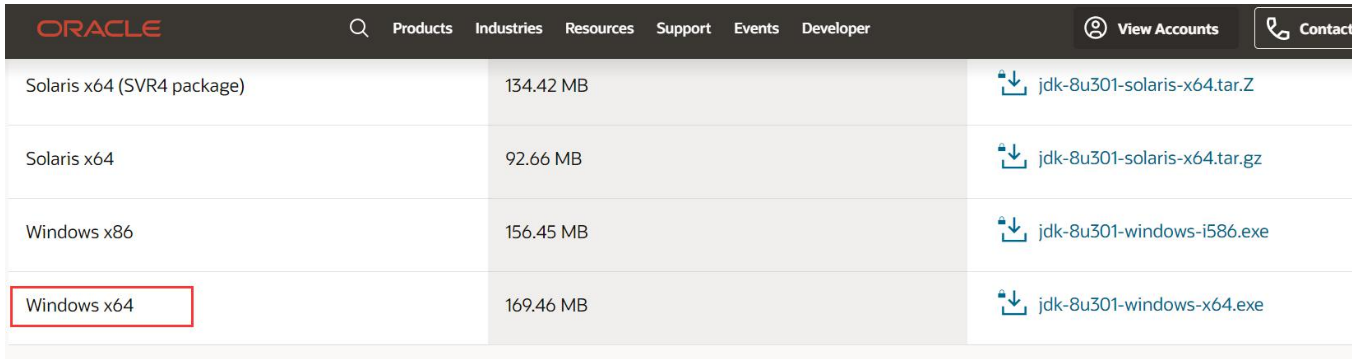
First Program

First Program

- Environment construction:

  there are many methods and options varying with your machine and preferances. here we take the windows for example:

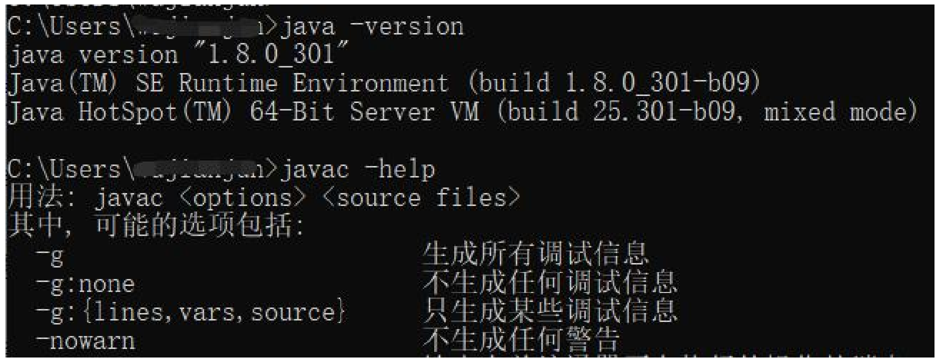
  - step1: downloading JDK,
    - https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html



Solaris x64 (SVR4 package)	134.42 MB	jdk-8u301-solaris-x64.tar.Z
Solaris x64	92.66 MB	jdk-8u301-solaris-x64.tar.gz
Windows x86	156.45 MB	jdk-8u301-windows-i586.exe
Windows x64	169.46 MB	jdk-8u301-windows-x64.exe

First Program

- Environment construction:
  - step2: Edit the following Environment variables
    - JAVA_HOME: ${install_root}\Java\jdk1.8.0_301
    - CLASSPATH: .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar;
    - Path: $PATH:%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;
      - note , to added, not to replace.
  - step3: verify



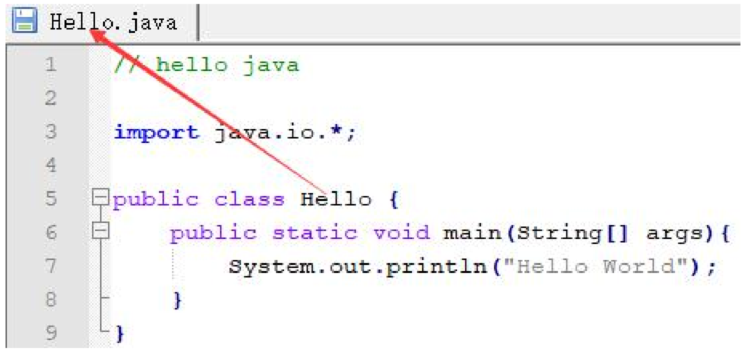
```
C:\Users\[redacted]>java -version
java version "1.8.0_301"
Java(TM) SE Runtime Environment (build 1.8.0_301-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.301-b09, mixed mode)

C:\Users\[redacted]>javac -help
用法: javac <options> <source files>
其中, 可能的选项包括:
  -g                         生成所有调试信息
  -g:none                    不生成任何调试信息
  -g:{lines,vars,source}     只生成某些调试信息
  -nowarn                    不生成任何警告
```

First Program

- Hello World Example：

  let’s see the simplest code,

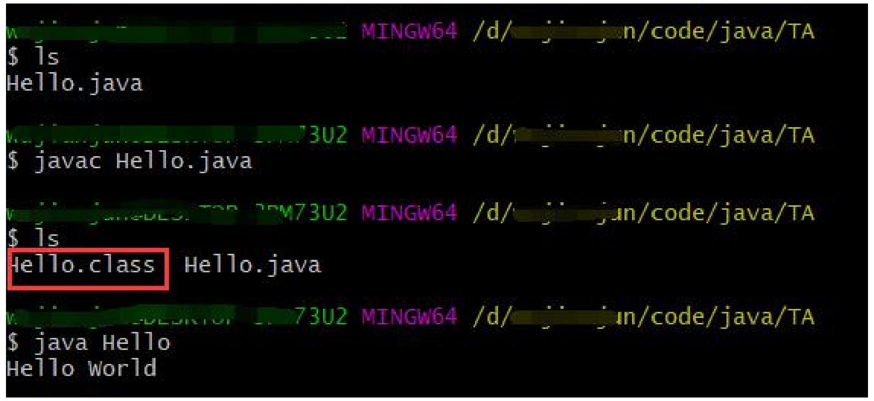
  - step1: write source codes



```
Hello.java
// hello java

import java.io.*;

public class Hello {
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

  - step2： compile and run

    a new .class file is generated



```
[redacted] MINGW64 /d/[redacted]n/code/java/TA
$ ls
Hello.java

[redacted]73U2 MINGW64 /d/[redacted]n/code/java/TA
$ javac Hello.java

[redacted]M73U2 MINGW64 /d/[redacted]un/code/java/TA
$ ls
Hello.class  Hello.java

[redacted]73U2 MINGW64 /d/[redacted]un/code/java/TA
$ java Hello
Hello World
```

First Program

- Hello World Example：
  - // represents a commented line.
  - import java.io.* means all the classes of io package can be imported.
  - class is used to declare a class in Java.
  - public means it is visible to all.
  - static means there is no need to create an object to invoke the method.
  - void means a method doesn't return any value.
  - main represents the starting point of the program.
  - String[] args is used for command line argument.
  - System.out.println() is used to print statement on an output device like the computer screen.

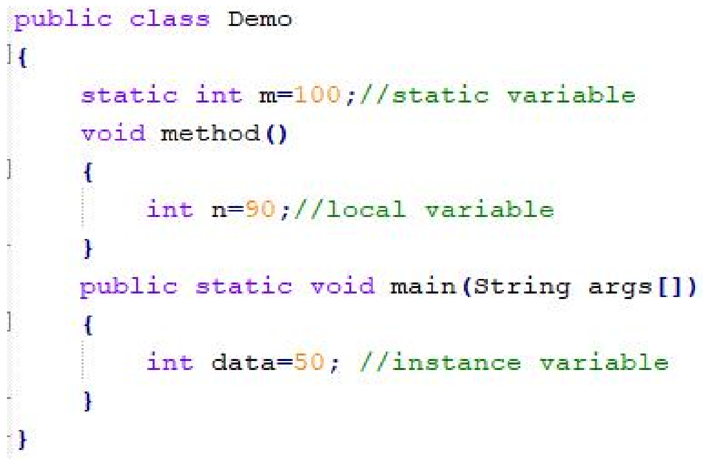
First Program

- some points:
    - Case Sensitivity: identifier Hello and hello would have different meaning.
    - Naming Convention: should follows camel syntax for naming ,
        - the first letter of every word shoud be in upper case,except class name, the rest letters shoud be in lower case. e.g: class MyFirstClass.
        - the first letter of class names should be in Upper Case, e.g: class MyFirstClass.
    - Name of the file should exactly match the class name.
    - Coding Guidelines is important for large projects, some of them are:
        - Alibaba: https://alibaba.github.io/Alibaba-Java-Coding-Guidelines/
        - Google: https://google.github.io/styleguide/javaguide.html
        - etc.

Basic Syntax

Basic Syntax

- Variables:
  - A variable is a container which holds the value. A variable is assigned with a type.
  - Variable is a name of memory location.
  - There are three types of variables in Java:
    - local variable: A variable declared inside the body of the method.
    - instance variable: A variable declared inside the class but outside the body of the non-static method.
    - static variable: A variable that is declared as static. You can create a single copy of the static variable and share it among all the instances of the class.



```
public class Demo
{
    static int m=100;//static variable
    void method()
    {
        int n=90;//local variable
    }
    public static void main(String args[])
    {
        int data=50; //instance variable
    }
}
```

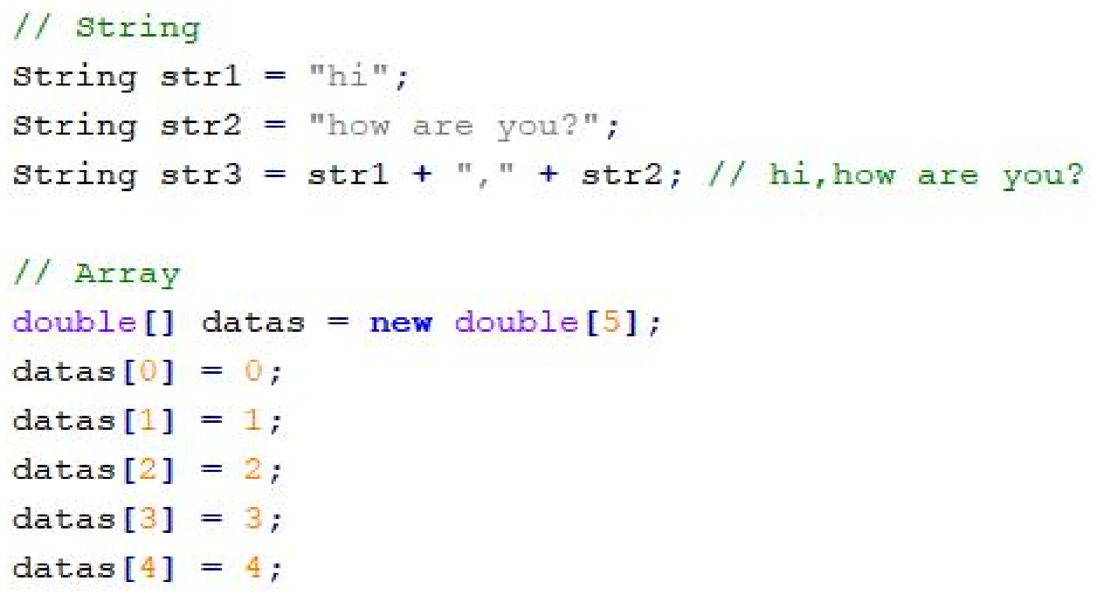
Basic Syntax

- Types：
  - primitive data types: int， long, float， double, char, etc.
  - wrapper classes: Wrapper classes are object representations of primitive data types.
    - Wrapper classes are used to represent primitive values when an Object is required. For example, Java collections only work with objects. They cannot take primitive types.
    - Wrapper classes also include some useful methods.
    - Placing primitive types into wrapper classes is called boxing. The reverse process is called unboxing.
    - Primitive types are faster than boxed types.

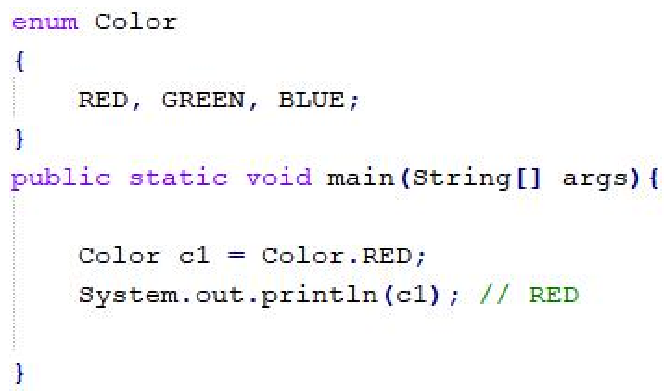
```java
double a = 3.14;
double b = 0.6;
System.out.println(a + b );

Double d = new Double(3.14);
Double e = new Double(0.6);
System.out.println(d + e);

Double g = a + 10; // boxing
double h = g - 10; // unboxing

System.out.println(g.toString().equals( "13.14")) ;
```

Basic Syntax

- Types：
  - non-primitive data types: class, enum, Interface, e.g, String,



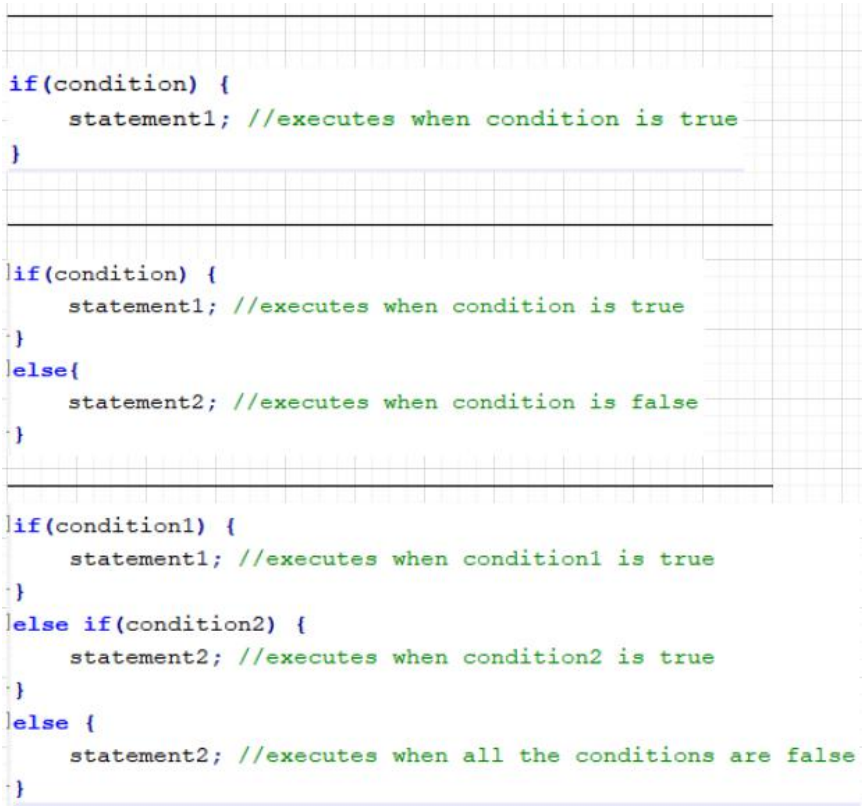
```
// String
String str1 = "hi";
String str2 = "how are you?";
String str3 = str1 + "," + str2; // hi,how are you?

// Array
double[] datas = new double[5];
datas[0] = 0;
datas[1] = 1;
datas[2] = 2;
datas[3] = 3;
datas[4] = 4;
```



```
enum Color
{
    RED, GREEN, BLUE;
}
public static void main(String[] args){

    Color c1 = Color.RED;
    System.out.println(c1); // RED

}
```

Basic Syntax

- Control Statements：
  - If Statement



```
if(condition) {
    statement1; //executes when condition is true
}
```

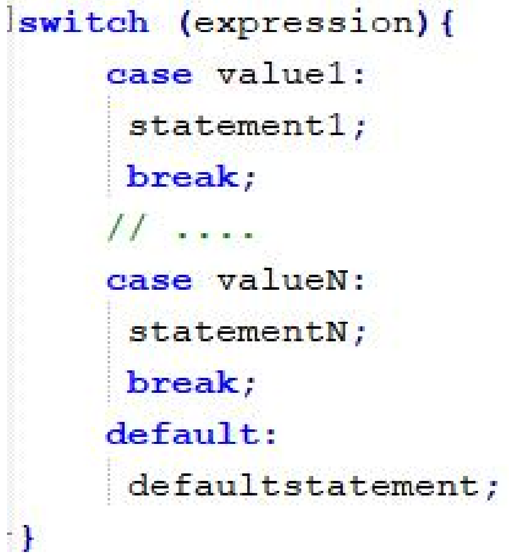
```
if(condition) {
    statement1; //executes when condition is true
}
else{
    statement2; //executes when condition is false
}
```

```
if(condition1) {
    statement1; //executes when condition1 is true
}
else if(condition2) {
    statement2; //executes when condition2 is true
}
else {
    statement2; //executes when all the conditions are false
}
```

if statement can be nested.

Basic Syntax

- Control Statements：
  - Switch Statement

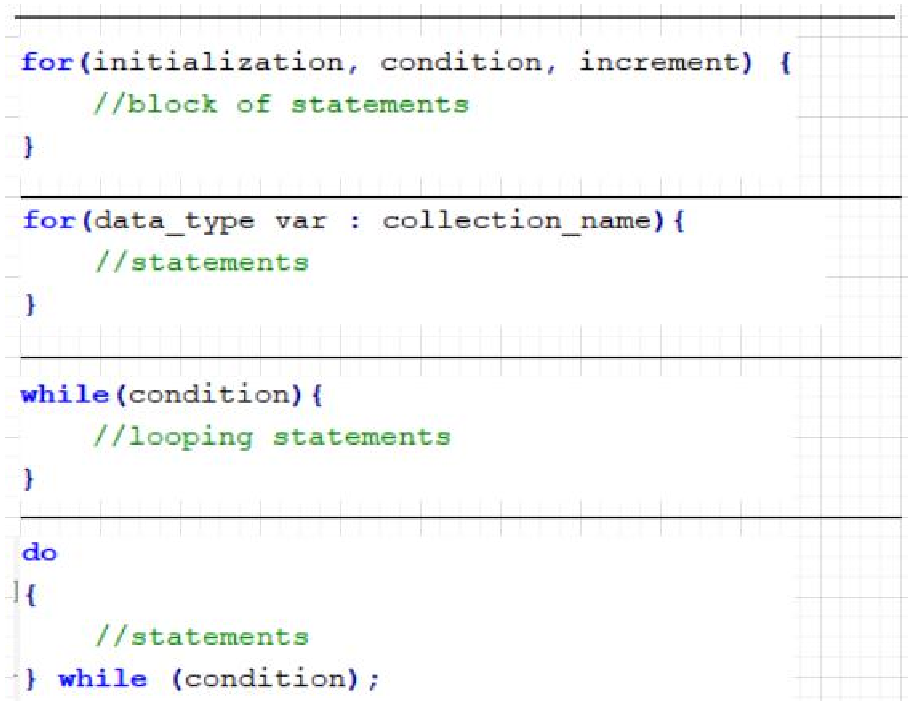


```
switch (expression){
    case value1:
     statement1;
     break;
    // ....
    case valueN:
     statementN;
     break;
    default:
     defaultstatement;
}
```

- The case variables can be int, short, byte, char, or enumeration. String type is also supported.
- Break statement terminates the switch block when the condition is satisfied.It is optional, if not used, next case is executed.

Basic Syntax

- Control Statements：
  - Loop Statements



```
for(initialization, condition, increment) {
    //block of statements
}
```

```
for(data_type var : collection_name){
    //statements
}
```

```
while(condition){
    //looping statements
}
```

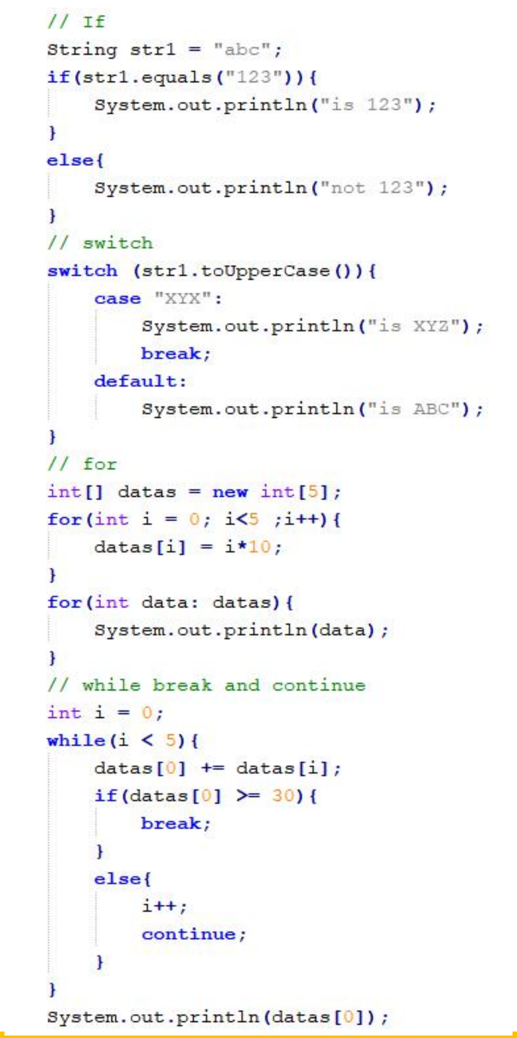
```
do
{
    //statements
} while (condition);
```

Basic Syntax

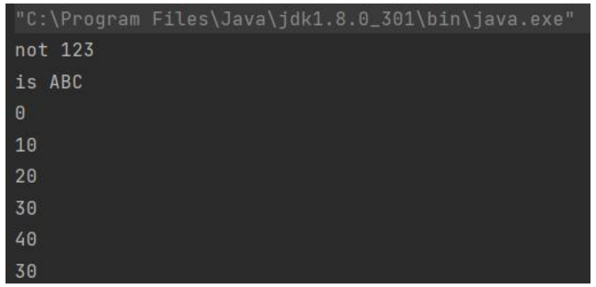
- Control Statements：
  - break statement: used to break the current flow and transfer the control to the next statement outside a loop or switch statement. However, it breaks only the inner loop in the case of the nested loop.
  - continue statement: skips the specific part of the loop and jumps to the next iteration of the loop immediately.

Basic Syntax

- Control Statements：
  - an examples:



```
// If
String str1 = "abc";
if(str1.equals("123")){
    System.out.println("is 123");
}
else{
    System.out.println("not 123");
}
// switch
switch (str1.toUpperCase()){
    case "XYX":
        System.out.println("is XYZ");
        break;
    default:
        System.out.println("is ABC");
}
// for
int[] datas = new int[5];
for(int i = 0; i<5 ;i++){
    datas[i] = i*10;
}
for(int data: datas){
    System.out.println(data);
}
// while break and continue
int i = 0;
while(i < 5){
    datas[0] += datas[i];
    if(datas[0] >= 30){
        break;
    }
    else{
        i++;
        continue;
    }
}
System.out.println(datas[0]);
```



```
"C:\Program Files\Java\jdk1.8.0_301\bin\java.exe"
not 123
is ABC
0
10
20
30
40
30
```

Q&A