

JAVA CORE

By Thuy Nguyen
26 August 2016

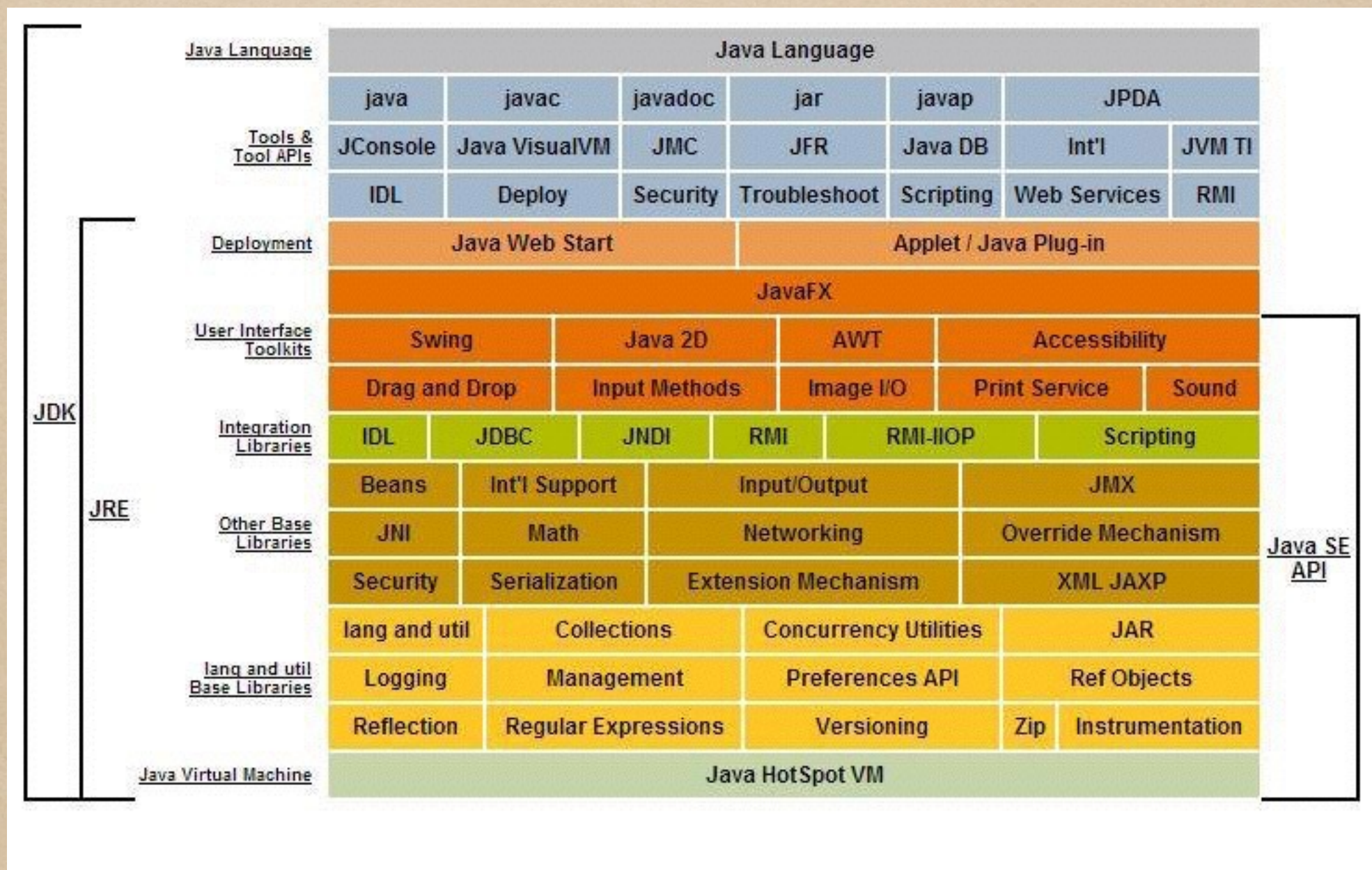
Agenda

- ◆ Getting Started with Java
- ◆ JDK vs JRE vs JVM
- ◆ Class & Object
- ◆ Java Variables
- ◆ Java Operators
- ◆ Java Statement
- ◆ String
- ◆ Array
- ◆ Exceptions
- ◆ Collection

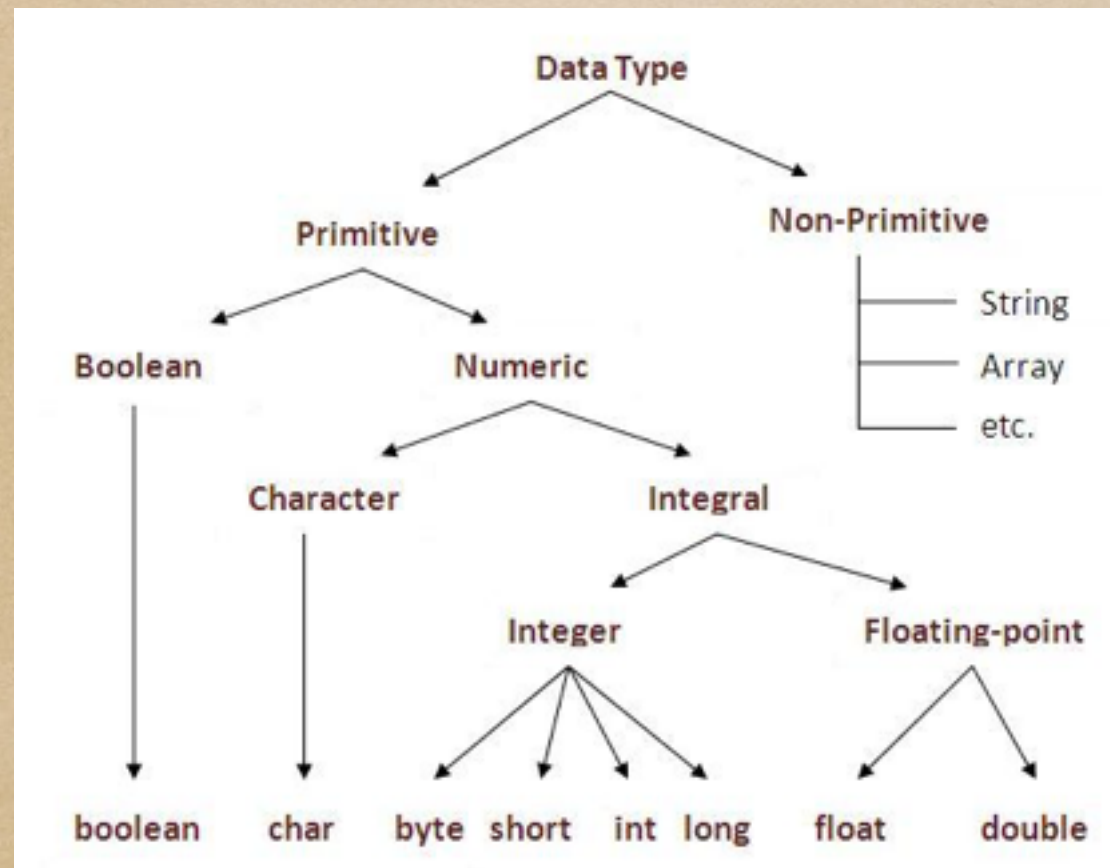
Getting Started with Java

- ◆ Where to download Java.
- ◆ How to install Java.
- ◆ Setting up the Environment Variables.
- ◆ Our First Java Program.

JDK vs JRE



Java Variables



| Data Type | Default Value |
|-----------|---------------|
| boolean | false |
| char | '\u0000' |
| byte | 0 |
| short | 0 |
| int | 0 |
| long | 0L |
| float | 0.0f |
| double | 0.0d |

```
public final int var = 9;
```


Class & Object

```
import java.io.*;
public class Employee {

    String name;
    int age;
    String designation;
    double salary;

    // This is the constructor of the class Employee
    public Employee(String name) {
        this.name = name;
    }

    // Assign the age of the Employee to the variable age.
    public void empAge(int empAge) {
        age = empAge;
    }

    /* Assign the designation to the variable designation.*/
    public void empDesignation(String empDesig) {
        designation = empDesig;
    }

    /* Assign the salary to the variable salary.*/
    public void empSalary(double empSalary) {
        salary = empSalary;
    }

    /* Print the Employee details */
    public void printEmployee() {
        System.out.println("Name:" + name );
        System.out.println("Age:" + age );
        System.out.println("Designation:" + designation );
        System.out.println("Salary:" + salary);
    }
}
```


Java Operators

| OPERATORS | PRECEDENCE(HIGH TO LOW) |
|----------------------|--|
| postfix | <code>expr++ expr--</code> |
| unary | <code>++expr --expr +expr -expr ~ !</code> |
| multiplicative | <code>* / %</code> |
| additive | <code>+ -</code> |
| shift | <code><< >> >>></code> |
| relational | <code>< > <= >= instanceof</code> |
| equality | <code>== !=</code> |
| bitwise AND | <code>&</code> |
| bitwise exclusive OR | <code>^</code> |
| bitwise inclusive OR | <code> </code> |
| logical AND | <code>&&</code> |
| logical OR | <code> </code> |
| ternary | <code>? :</code> |
| assignment | <code>= += -= *= /= %= &= ^= = <<= >>= >>>=</code> |

Java Statement

```
if (a == 2) {  
    System.out.println("a is 0");  
} else {  
    System.out.println("a is not 2");  
}
```

```
double r = 0;  
while(r < 0.99d) {  
    r = Math.random();  
    System.out.println(r);  
}
```

```
int i = 1;  
switch (i) {  
    case 1 :  
        System.out.println("One.");  
        break;  
    case 2 :  
        System.out.println("Two.");  
        break;  
    case 3 :  
        System.out.println("Three.");  
        break;  
    default:  
        System.out.println("You did not enter a valid  
}
```

```
for(int x : numbers ) {  
    System.out.print( x );  
    System.out.print(",");  
}
```

```
public class MainClass {  
    enum Season {  
        spring, summer, fall, winter  
    }  
  
    public static void main(String[] args) {  
        for (Season season : Season.values()) {  
            System.out.println(" The season is now " + season);  
        }  
    }  
}
```

```
public static void main(String[] args) {  
  
    OuterLoop: for (int i = 2;; i++) {  
        for (int j = 2; j < i; j++) {  
            if (i % j == 0) {  
                continue OuterLoop;  
            }  
        }  
  
        System.out.println(i);  
        if (i == 107) {  
            break;  
        }  
    }  
}
```


String

```
// Here following primitive char 'a'  
// is boxed into the Character object ch  
Character ch = 'a';  
  
// Here primitive 'x' is boxed for method test,  
// return is unboxed to char 'c'  
char c = test('x');
```

```
public class StringDemo {  
  
    public static void main(String args[]) {  
        char[] helloArray = { 'h', 'e', 'l', 'l', 'o', '.' };  
        String helloString = new String(helloArray);  
        System.out.println( helloString );  
    }  
}
```


Array

```
public class TestArray {  
  
    public static void main(String[] args) {  
        double[] myList = {1.9, 2.9, 3.4, 3.5};  
  
        // Print all the array elements  
        for (int i = 0; i < myList.length; i++) {  
            System.out.println(myList[i] + " ");  
        }  
  
        // Summing all elements  
        double total = 0;  
        for (int i = 0; i < myList.length; i++) {  
            total += myList[i];  
        }  
        System.out.println("Total is " + total);  
  
        // Finding the largest element  
        double max = myList[0];  
        for (int i = 1; i < myList.length; i++) {  
            if (myList[i] > max) max = myList[i];  
        }  
        System.out.println("Max is " + max);  
    }  
}
```


Exceptions

An exception can occur for many different reasons. Following are some scenarios where an exception occurs.

- ▣ A user has entered an invalid data.
- ▣ A file that needs to be opened cannot be found.
- ▣ A network connection has been lost in the middle of communications or the JVM has run out of memory.

```
// File Name : ExceptTest.java
import java.io.*;

public class ExceptTest {

    public static void main(String args[]) {
        try {
            int a[] = new int[2];
            System.out.println("Access element three :" + a[3]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Exception thrown  :" + e);
        }
        System.out.println("Out of the block");
    }
}
```


List

```
List<String> listStrings = new LinkedList<String>();  
listStrings.add("Five");  
listStrings.add("Six");  
listStrings.add("Seven");  
listStrings.add("Eight");  
  
System.out.println(listStrings);
```

```
List<Number> linkedNumbers = new LinkedList<>();  
  
linkedNumbers.add(new Integer(123));  
linkedNumbers.add(new Float(3.1415));  
linkedNumbers.add(new Double(299.988));  
linkedNumbers.add(new Long(67000));
```


Homework

Tính tổng nguyên: $S = 1 + 1/2 + 1/3 + 1/4 + \dots 1/n$

Tìm giá trị lớn nhất của 2 số

Tìm giá trị lớn nhất của 4 số a,b,c,d

Tìm ước số chung và bội số chung nhỏ nhất của 2 số a,b

Kiểm tra số N có phải là số nguyên tố không?

Liệt kê các số nguyên tố nhỏ hơn N

Homework

- In ra phần tử âm của mảng
- In ra các phần tử có giá trị lẻ của mảng
- In ra các phần tử chẵn và nhỏ hơn 20
- Tìm vị trí phần tử có giá trị x xuất hiện cuối cùng trong mảng
- Tìm vị trí của phần tử nhỏ nhất trong mảng 1 chiều
- Tìm vị trí của phần tử lớn nhất trong mảng 1 chiều
- Tính tổng các phần tử ở vị trí chẵn trong mảng 1 chiều
- Sắp xếp mảng theo thứ tự giảm dần
- Sắp xếp các phần tử chẵn giảm dần
- Sắp xếp các phần tử âm dương trong mảng
- Xóa phần tử tại vị trí lẻ trong mảng
- Xóa phần tử có giá trị lớn nhất trong mảng
- Tách 1 mảng thành 2 mảng chẵn và lẻ
- Đảo ngược dãy số

Homework

Một lớp đối tượng sách trong hệ thống quản lý thư viện có các thuộc tính

- Tên sách
- Tổng số quyển sách
- Số quyển sách đã cho mượn

Xây dựng lớp đối tượng trên với các phương thức sau:

- Phương thức in thông tin đối tượng ra màn hình bao gồm tên, tổng số và số đã cho mượn.
- Phương thức tính số sách còn lại trong thư viện (tổng số – số mượn).

Preference

- ◆ <http://javabeginnerstutorial.com/core-java-tutorial/>
- ◆ <http://www.tutorialspoint.com/java/>
- ◆ <http://www.javatpoint.com/>

Examples

- ◆ <http://www.java-examples.com/basic-java-examples>
- ◆ <http://www.java2s.com/Tutorial/Java/>