Object Oriented Programming with Java 8
PG-DAC

Rohan Paramane

Agenda

- Getter and Setter
- Enum
- Object class
- Tostring
- Static

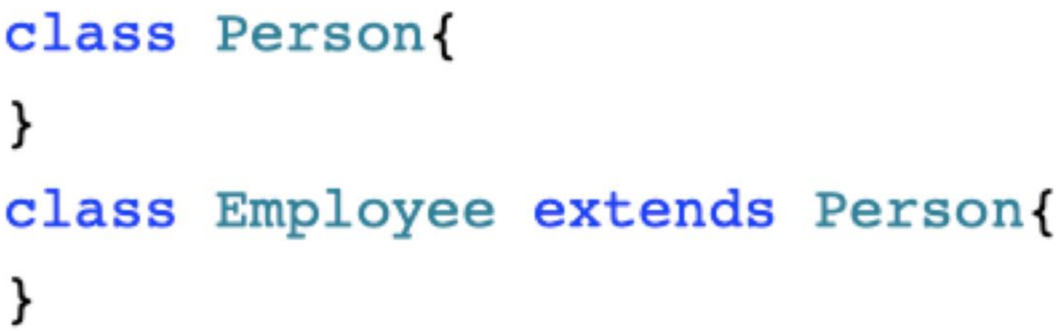
Object class

- It is a non final and concrete class declared in java.lang package.
- In java all the classes(not interfaces)are directly or indirectly extended from java.lang.Object class.
- In other words, java.lang.Object class is ultimate base class/super cosmic base class/root of Java class hierarchy.
- Object class do not extend any class or implement any interface.
- It doesn’t contain nested type as well as field.
- It contains default constructor.
  - ➢ `Object o = new Object(“Hello”);      //Not OK`
  - ➢ `Object o = new Object( );           //OK`

- Object class contains 11 methods.

Object class

- Consider the following code:



```
class Person{
}
class Employee extends Person{
}
```

- In above code, java.lang.object is direct super class of class Person.

- In case class Employee, class Person is direct super class and class object is indirect super class.

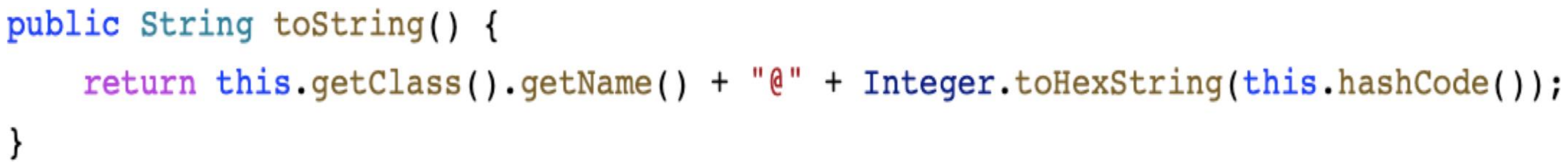
Methods Of Object class

```
1. public String toString();
2. public boolean equals(Object obj);
3. public native int hashCode( );
4. protected native Object clone( )throws CloneNotSupportedException
5. protected void finalize( void )throws Throwable


6. public final native Class<?> getClass( );
7. public final void wait( )throws InterruptedException
8. public final native void wait( long timeout)throws InterruptedException
9.public final void wait( long timeout, int nanos)throws InterruptedException
10.public final native void notify( );
11.public final native void notifyAll( );
```

toString( ) method

- It is a non final method of java.lang.Object class.
- Syntax:
  - **public String toString( );**
- If we want to return state of Java instance in String form then we should use toString() method.
- Consider definition of toString inside Object class:

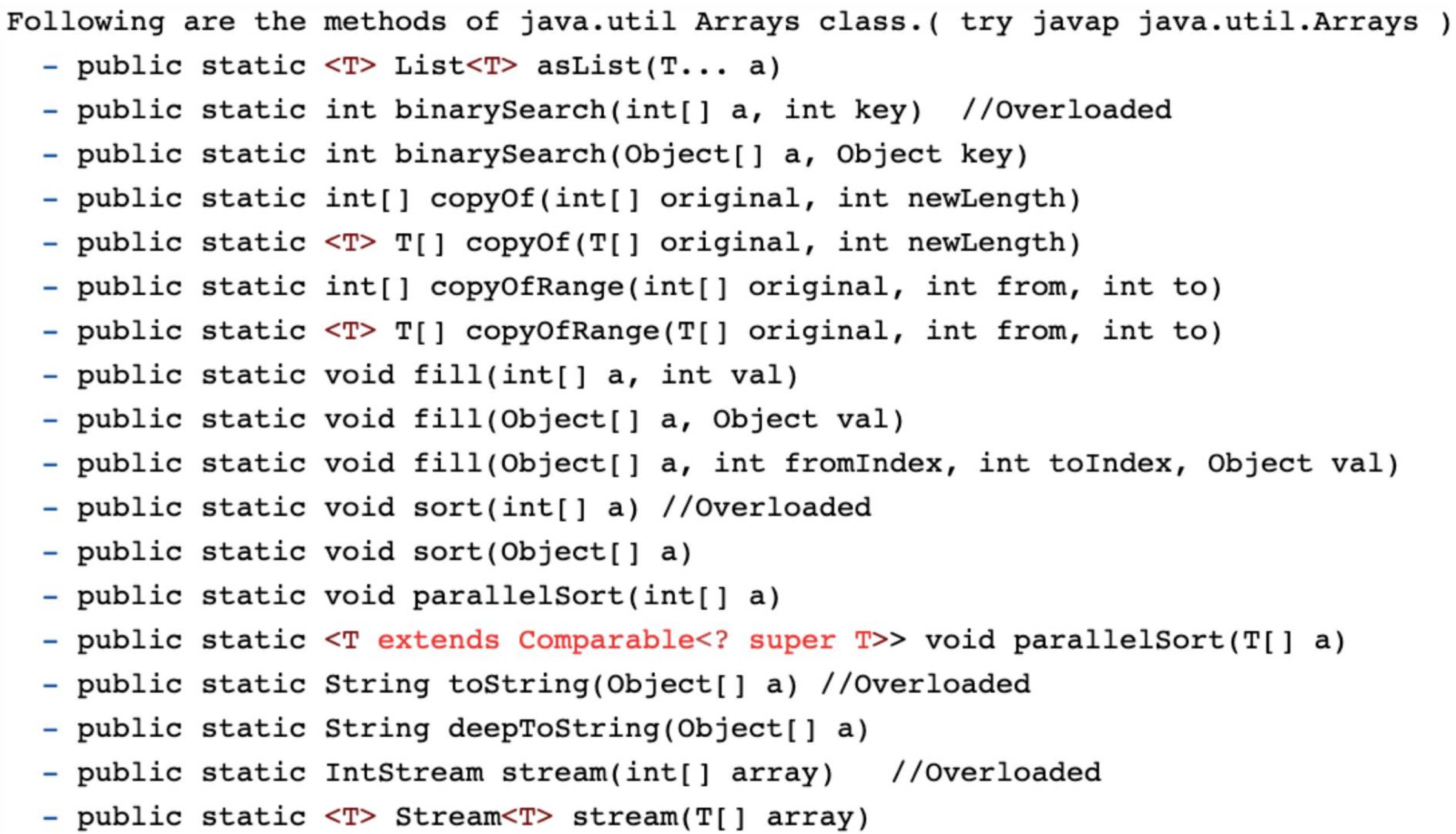


```
public String toString() {
    return this.getClass().getName() + "@" + Integer.toHexString(this.hashCode());
}
```

toString( ) method

- If we do not define toString() method inside class then super class's toString() method gets call.
- If we do not define toString() method inside any class then object class's toString() method gets call.
- It return String in following form:
  - **F.Q.ClassName@HashCode**
  - **Example : test.Employee@6d06d69c**
- If we want state of instance then we should override toString() method inside class.
- The result in toString method should be a concise but informative that is easy for a person to read.
- It is recommended that all subclasses override this method.

Methods Of java.util.Arrays Class

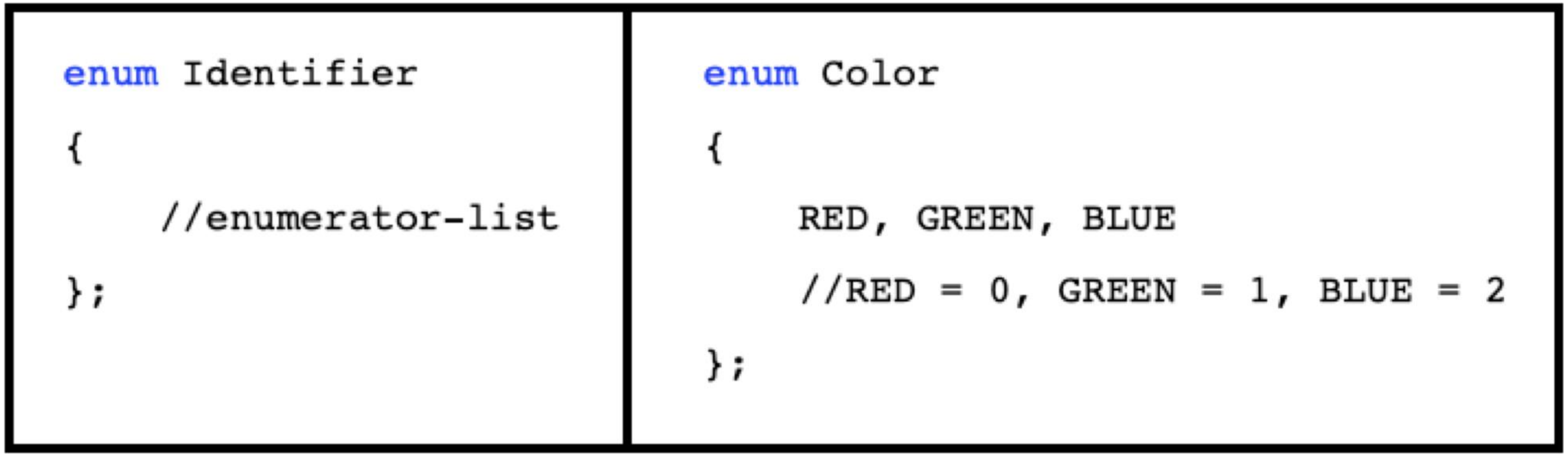


Following are the methods of java.util Arrays class.(try javap java.util.Arrays)

- public static <T> List<T> asList(T... a)
- public static int binarySearch(int[] a, int key) //Overloaded
- public static int binarySearch(Object[] a, Object key)
- public static int[] copyOf(int[] original, int newLength)
- public static <T> T[] copyOf(T[] original, int newLength)
- public static int[] copyOfRange(int[] original, int from, int to)
- public static <T> T[] copyOfRange(T[] original, int from, int to)
- public static void fill(int[] a, int val)
- public static void fill(Object[] a, Object val)
- public static void fill(Object[] a, int fromIndex, int toIndex, Object val)
- public static void sort(int[] a) //Overloaded
- public static void sort(Object[] a)
- public static void parallelSort(int[] a)
- public static <T extends Comparable<? super T>> void parallelSort(T[] a)
- public static String toString(Object[] a) //Overloaded
- public static String deepToString(Object[] a)
- public static IntStream stream(int[] array)   //Overloaded
- public static <T> Stream<T> stream(T[] array)

Enum In C/C++ Programming language.

- According ANSI C standard, if we want to assign name to the integer constant then we should use enum.
- Enum helps developer to improve readability of source code.
- enum is keyword in C. Let us consider syntax of enum:

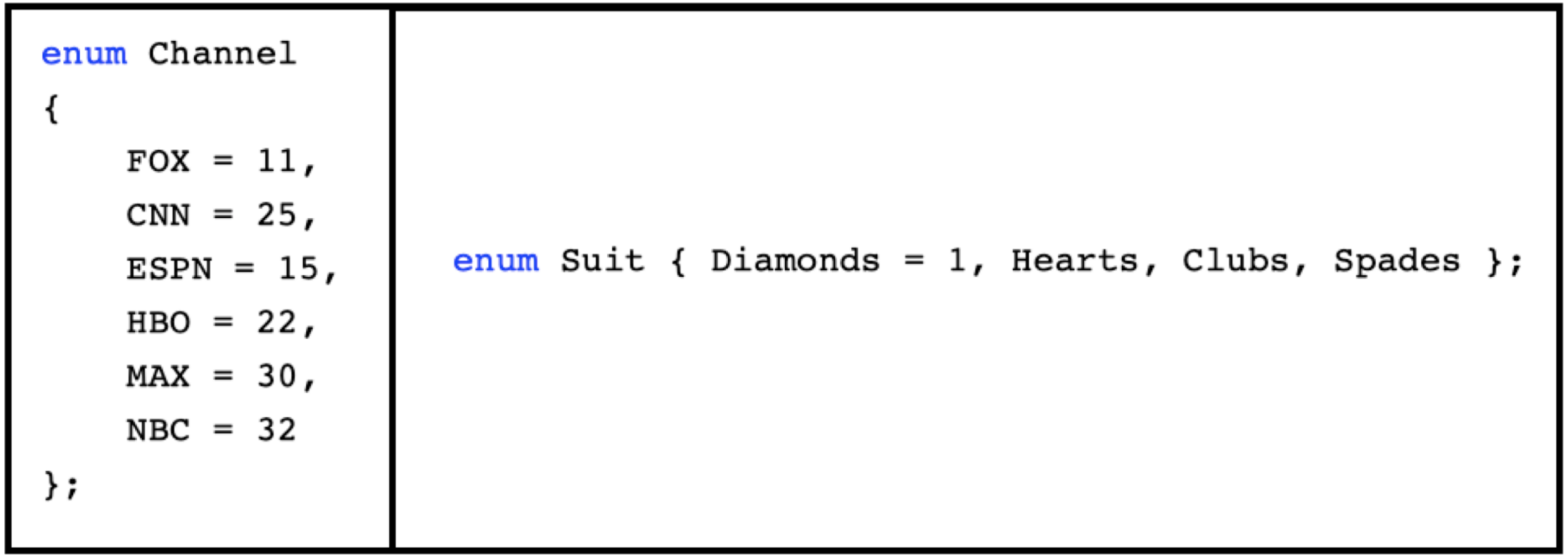


```
enum Identifier
{
    //enumerator-list
};
```

```
enum Color
{
    RED, GREEN, BLUE
    //RED = 0, GREEN = 1, BLUE = 2
};
```

Enum In C/C++ Programming language.

- By default, the first enumeration-constant is associated with the value 0. The next enumeration-constant in the list is associated with the value of (constant-expression + 1), unless you explicitly associate it with another value.



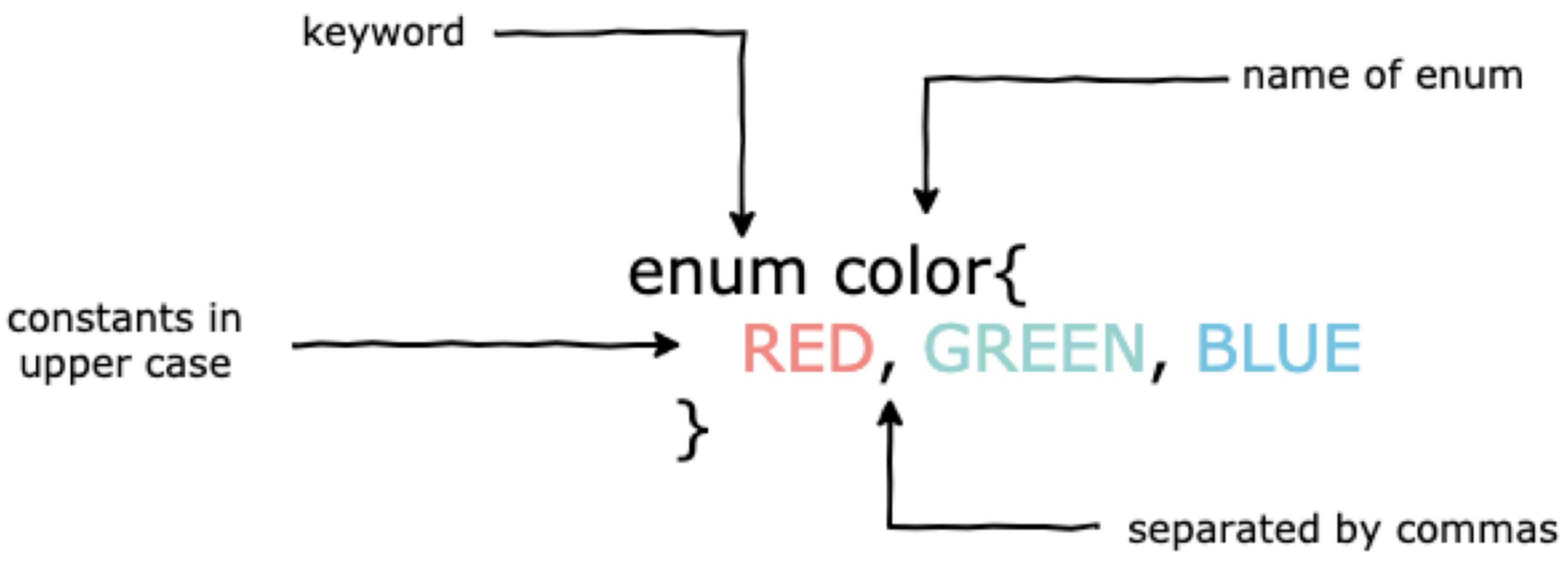
```
enum Channel
{
    FOX = 11,
    CNN = 25,
    ESPN = 15,
    HBO = 22,
    MAX = 30,
    NBC = 32
};
```

```
enum Suit { Diamonds = 1, Hearts, Clubs, Spades };
```

- constant-expression must have int type and can be negative.

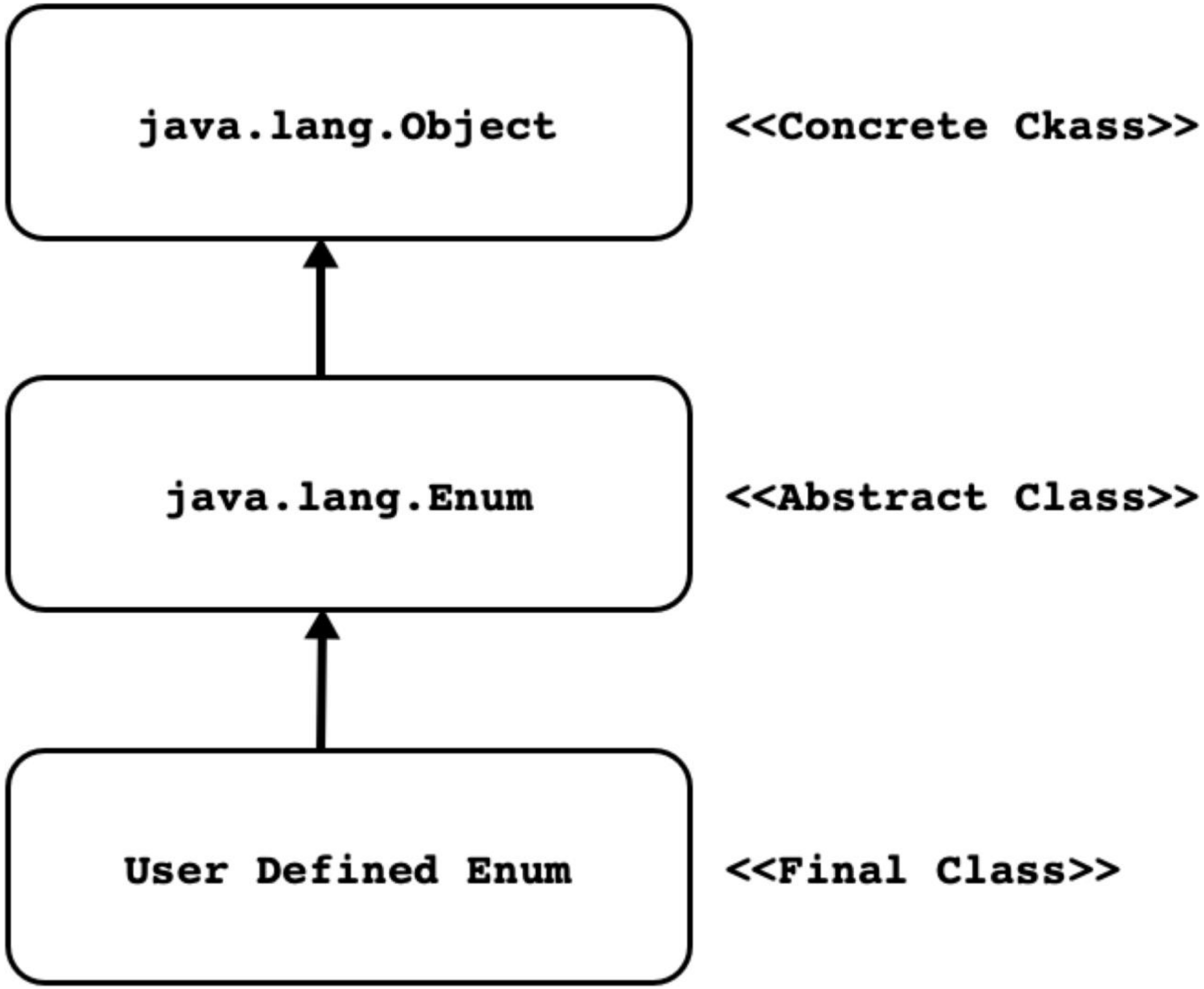
Enum In Java Programming language.

- An enum is a class that represents a group of constants.
- **Enum keyword** is used to create an enum. The constants declared inside are separated by a comma and should be in upper case.



```
enum color{
  RED, GREEN, BLUE
}
```

Enum Class Hierarchy



java.lang.Object

<<Concrete Ckass>>

java.lang.Enum

<<Abstract Class>>

User Defined Enum

<<Final Class>>

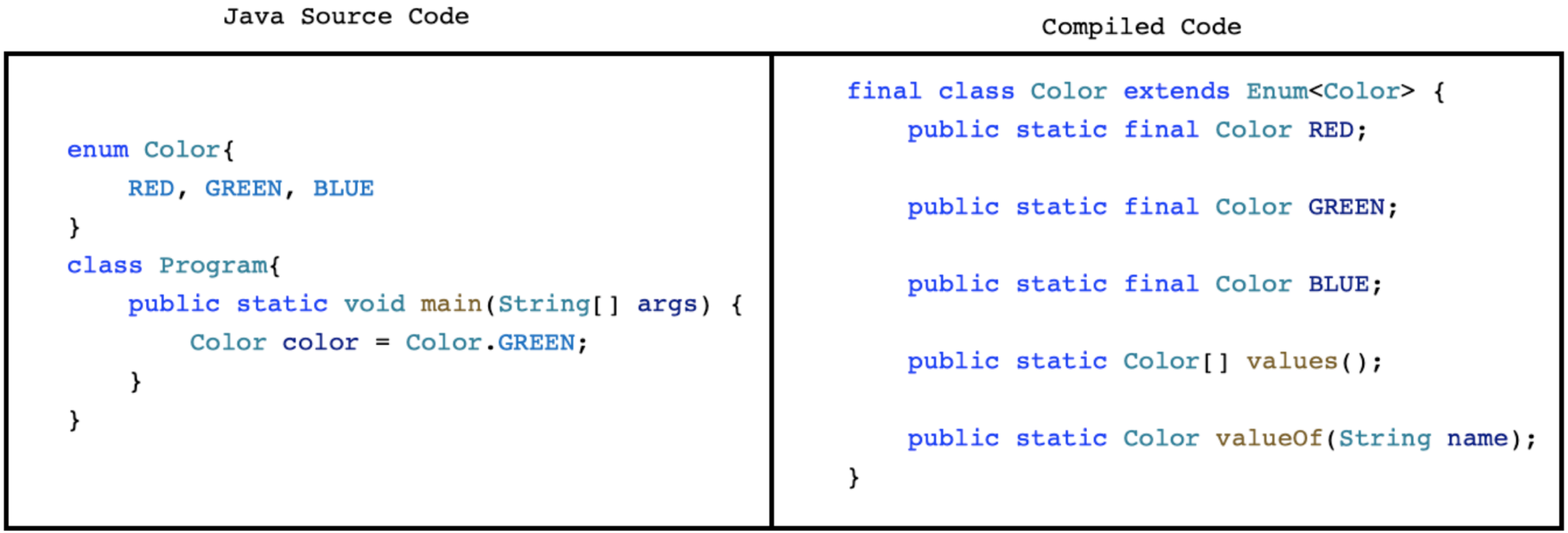
Enum API

- **`Following are the methods declared in java.lang.Enum class:`**

String	name()Returns the name of this enum constant, exactly as declared in its enum declaration.
int	ordinal()Returns the ordinal of this enumeration constant (its position in its enum declaration, where the initial constant is assigned an ordinal of zero).
String	toString()Returns the name of this enum constant, as contained in the declaration.
static <T extends Enum<T>> T	valueOf(Class<T> enumType, String name)Returns the enum constant of the specified enum type with the specified name.

Sole constructor : Programmers cannot invoke this constructor. It is for use by code emitted by the compiler in response to enum type declarations.

Enum for the compiler



Java Source Code

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

Compiled Code

```
final class Color extends Enum<Color> {
    public static final Color RED;

    public static final Color GREEN;

    public static final Color BLUE;

    public static Color[] values();

    public static Color valueOf(String name);
}
```

Properties of enum

1. Similar to a class, an enum can have objects and methods. The only difference is that enum constants are public, static and final by default. Since it is final, we can’t extend enums
2. It cannot extend other classes since it already extends the java.lang.Enum class.
3. It can implement interfaces.
4. The enum objects cannot be created explicitly and hence the enum constructor cannot be invoked directly.
5. It can only contain concrete methods and no abstract methods.

Thank you.

Rohan.paramane@sunbeaminfo.com