Main.java

*ENUM*

**enum** Color {

***RED***(255, 0, 0),

***GREEN***(0, 255, 0),

***BLUE***(0, 0, 255),

***WHITE***(255, 255, 255),

***BLACK***(0, 0, 0),

***GRAY***(128, 128, 128),

***CYAN***(0, 255, 255),

***MAGENTA***(255, 0, 255),

***YELLO***(255, 255, 0);

**private** **final** **int** r;

**private** **final** **int** g;

**private** **final** **int** b;

**private** Color(**int** r, **int** g, **int** b) {

**this**.r = r;

**this**.g = g;

**this**.b = b;

}

**private** **int** r() {**return** r;}

**private** **int** g() {**return** g;}

**private** **int** b() {**return** b;}

**public** String hexValue() {

String rString = Integer.*toHexString*(r);

**if**(rString.length()<=1) rString = "0"+rString;

String gString = Integer.*toHexString*(g);

**if**(gString.length()<=1) gString = "0"+gString;

String bString = Integer.*toHexString*(b);

**if**(bString.length()<=1) bString = "0"+bString;

**return** "#"+ rString + gString + bString;

}

}

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

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

**for**(Color colors : Color.*values*()) {

System.***out***.print(colors.ordinal() +": ");

System.***out***.print(colors); //same output colors = colors.toString() = colors.name()

System.***out***.print("/"+colors.toString()); //PrintStream calls colors.toString() when it is provided with colors

System.***out***.print("/"+colors.name()); //toString() of ENUM calls the name() method which outputs the ENUM values

System.***out***.println(" = " + colors.hexValue());

} //sample output below:

//0: RED/RED/RED = #ff0000

//1: GREEN/GREEN/GREEN = #00ff00

//2: BLUE/BLUE/BLUE = #0000ff

//....

Color color;

color = Color.***RED***;

System.***out***.println(color + " = " + color.hexValue()); //Prints: RED = #ff0000

System.***out***.println(color); //Prints: RED

System.***out***.println(color.name()); //Prints: RED

System.***out***.println(color.ordinal()); //Prints: 0

System.***out***.println(color == Color.***RED***); //true

System.***out***.println(Color.***RED***.toString().equals("RED")); //true

color = Color.*valueOf*("GREEN");

System.***out***.println(color.name()); //Prints: GREEN

//color = Color.valueOf("Blue"); //Throws java.lang.IllegalArgumentException: No enum constant Color.Blue

//System.out.println(color.name());

color = Color.***GRAY***;

**switch**(color) {

**case** ***WHITE***:

System.***out***.println("Too bright.");

**break**;

**case** ***GRAY***:

System.***out***.println("Just ok.");

**break**;

**case** ***BLACK***:

System.***out***.println("Too dark;");

**break**;

//case SOMETHING: //Compilation fails. SOMETHING cannot be resolved or is not a field

//case Color.RED: //Compilaiton fails. The qualified case label Color.RED must be replaced with the unqualified enum constant RED

}

**for**(HoursDefault hours : HoursDefault.*values*()) {

hours.getHour();

}

//Prints:

//Winter Hours.

//Summer Hours.

//Default Hours.

}

}

**enum** Sample1 {

***A***, ***B***, ***C*** //Semicolon is optional here as there is nothing but the values

}

**enum** Sample2 {

***A***, ***B***, ***C***; //Semicolon is optional here as there is nothing but the values

}

**enum** HoursAbstract {

***WINTER*** {

@Override

**public** **void** getHour() {

System.***out***.println("Winter Hours.");

}

},

***SUMMER*** {

@Override

**public** **void** getHour() {

System.***out***.println("Summer Hours.");

}

}

//, SPRING //Compilation fails. The enum constant SPRING must implement the abstract method getHour()

;

**public** **abstract** **void** getHour();

}

**enum** HoursDefault {

***WINTER*** {

@Override

**public** **void** getHour() {

System.***out***.println("Winter Hours.");

}

},

***SUMMER*** {

@Override

**public** **void** getHour() {

System.***out***.println("Summer Hours.");

}

},

***SPRING***;

**public** **void** getHour() { System.***out***.println("Default Hours.");}

}