

1、课程名称：枚举的其他应用



2、本课程预计讲解的知识点



3、具体内容

枚举类型可以跟普通的类一样实现一个接口，但是实现接口的时候要求枚举中的每个对象都必须单独编写好接口中的抽象方法。

```
interface Print{  
    public String getColor();  
}  
enum Color implements Print{  
    RED
```

```
        public String getColor(){  
            return "红色";  
        }  
    },GREEN{  
        public String getColor(){  
            return "绿色";  
        }  
    },BLUE{  
        public String getColor(){  
            return "蓝色";  
        }  
    }  
};  
public class InterfaceEnumDemo{  
    public static void main(String args[]){  
        for(Color c:Colorvalues()){  
            System.out.print(c.getColor()+"、");  
        }  
    }  
};
```

还可以直接在枚举中定义抽象方法，但是要求枚举中的每个对象都分别实现此抽象方法。

```
enum Color implements Print{  
    RED{  
        public String getColor(){  
            return "红色";  
        }  
    },GREEN{  
        public String getColor(){  
            return "绿色";  
        }  
    },BLUE{  
        public String getColor(){  
            return "蓝色";  
        }  
    }  
};  
public abstract String getColor();  
}  
public class AbstractMethodEnum{
```

```
        public static void main(String args[]){  
            for(Color c:Colorvalues()){  
                System.out.print(c.getColor()+"、");  
            }  
        }  
    }  
};
```

4、总结

1、枚举的最大作用在于限定取值范围，在随后的 Annotation 中也会有所讲解。

2、如果已经习惯了使用其他语言的枚举操作的话，可能会适应 JAVA 的枚举的开发习惯，而如果没有枚举相关经验的话，那么也不是很建议使用枚举。

5、预习任务

