1. 枚举出现的背景，实现原理
   1. 枚举是jdk1.5后的新特性，jdk1.6后switch支持枚举类型
   2. 枚举类型由关键字enum定义，本质其实也是一个类，只是这个类被限定了其对象的产生范围，构造器被私有化，在类的内部自己创建了所限定的几个常量对象
   3. enmu定义一个枚举类型后，编译器会为我们生成一个相关的类，这个类是一个final类且继承java.lang.Enum类

enum Day { MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY }

上面的枚举类型翻译成如下的类

final class Day extends Enum

{

//编译器为我们添加的静态的values()方法

public static Day[] values()

{

return (Day[])$VALUES.clone();

}

//编译器为我们添加的静态的valueOf()方法，注意间接调用了Enum也类的valueOf方法

public static Day valueOf(String s)

{

return (Day)Enum.valueOf(com/zejian/enumdemo/Day, s);

}

//私有构造函数

private Day(String s, int i)

{

super(s, i);

}

//前面定义的7种枚举实例

public static final Day MONDAY;

public static final Day TUESDAY;

public static final Day WEDNESDAY;

public static final Day THURSDAY;

public static final Day FRIDAY;

public static final Day SATURDAY;

public static final Day SUNDAY;

private static final Day $VALUES[];

static

{

//实例化枚举实例

MONDAY = new Day("MONDAY", 0);

TUESDAY = new Day("TUESDAY", 1);

WEDNESDAY = new Day("WEDNESDAY", 2);

THURSDAY = new Day("THURSDAY", 3);

FRIDAY = new Day("FRIDAY", 4);

SATURDAY = new Day("SATURDAY", 5);

SUNDAY = new Day("SUNDAY", 6);

$VALUES = (new Day[] {

MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY

});

}

}

1. 枚举能做什么，解决什么问题
2. 枚举知识点清单
   1. 向enum类添加方法和自定义构造函数
   2. enum类中定义抽象方法
   3. enum类与接口（enum已继承Enum类，故不能继承其他类，但不妨碍它实现接口）
   4. EnumMap的基本用法
   5. EnumSet的基本用法
3. 练习、输出

参考：

<https://blog.csdn.net/qq_27093465/article/details/52180865>

https://blog.csdn.net/javazejian/article/details/71333103