Java Day 13

泛型

参数化数据类型，将数据类型作为参数进行传递

泛型好处：

1）提高安全性：将运行期的 ClassCastException 转换到编译期报错

2）避免了类型强转的麻烦

3）增加可读性

定义和使用含有泛型的类

定义格式：修饰符 class 类名<代表泛型的变量> { }

例如，API 中的 ArrayList 集合：

```
class ArrayList<E>{
	public boolean add(E e){ }

	public E get(int index){ }
	....
}
```

使用泛型： 即什么时候确定泛型，在创建对象的时候确定泛型

例如，ArrayList<String> list = new ArrayList<String>();

此时，变量 E 的值就是 String 类型,那么我们的类型就可以理解为：

```
class ArrayList<String>{
       public boolean add(String e){ }
       public String get(int index){  }
        ...
}
```

注意：

泛型不支持子类型，=号左右两边泛型类型保持一致

JDK7 以后，=号右边的泛型类型可以省略

```
List<String> list = new ArrayList<>();
```

通配符

<?>通配符，表示泛型类型未知，可以传递任何数据类型-->WildcardsTest.java

<? extends 数据类型>限定通配符，可以传递“数据类型”及其子类型

<? super 数据类型>限定通配符，可以传递“数据类型”及其父类型

自定义泛型类

1）在类定义中加入类型参数，一般用大写字母表示

2）类中类型用大写字母来代替

裸类型：泛型类型没有使用类型参数，类型不安全

类型擦除：编译以后的字节码文件中，所有泛型信息将被删除

泛型仅在编译阶段，为编译器提供类型检查信息

枚举：自定义枚举类型

---

enum

和 class、interface 同级，用来定义枚举类型，取值在指定范围内

枚举和 switch-->EnumTest1.java 、 EnumTest2.java 、 EnumTest3.java

enum 使用注意：

1）可以定义在包中、类中，但不能定义在方法中

2）枚举类型不能被继承

3）枚举类型默认继承 java.lang.Enum

4）枚举类型可以有属性、方法、构造器(不能用 public)

5）枚举类型的每个取值都是自己的一个实例对象

6）枚举类型可以实现接口

7）枚举类型有静态方法 values()获取所有取值

8）枚举类型中可以有抽象方法，必须在每个取值中重写

9）java.util.EnumSet 集合辅助使用 enum 类型：allOf()/range()/of()...-->EnumSetTest.java

10）java.util.EnumMap 集合辅助使用 enum 类型-->EnumMapTest.java

反射

概念

镜像：类被加载到内存以后，通过类镜像来表，java.lang.Class

反射：在运行时，通过类镜像进行类操作（获取类信息、构建类对象、调用方法等），使用 java.lang.reflect 包下的 API（Field、Method、Constructor 等）

反射使用场景

当类的信息在编译时无法确定，运行时才能确定下来时，必须使用反射进行相关操作

反射编程步骤

1）获取类镜像：Class = java.lang.Class.forName(...)

2）如需要，创建类实例对象：Class-->newInstance()

3）调用反射相关 API：

```
java.lang.reflect.Field
java.lang.reflect.Constructor
java.lang.reflect.Method
```

类加载器（了解）

1）类加载器的概述

负责将.class 文件加载到内存中，并为之生成对应的 Class 对象。

2）类加载器的分类

Bootstrap ClassLoader 根类加载器

Extension ClassLoader 扩展类加载器

Sysetm ClassLoader 系统类加载器

3）类加载器的作用

Bootstrap ClassLoader 根类加载器

也被称为引导类加载器，负责 Java 核心类的加载

比如 System,String 等。在 JDK 中 JRE 的 lib 目录下 rt.jar 文件中

Extension ClassLoader 扩展类加载器

负责 JRE 的扩展目录中 jar 包的加载。

在 JDK 中 JRE 的 lib 目录下 ext 目录

Sysetm ClassLoader 系统类加载器

负责在 JVM 启动时加载来自 java 命令的 class 文件，以及 classpath 环境变量所指定的 jar 包和类路径