

用了 `toUpperCase()` 方法, 当调用 `get()` 方法后可以实现大写转换操作。

范例 8-57: 断言型接口。断言型接口主要是进行判断操作, 本身需要接收一个参数, 同时会返回一个判断结果 (boolean 型)。本次将引用 `String` 类中的 `equalsIgnoreCase()` 方法 (public boolean equalsIgnoreCase(String str))。

```
package com.yootk.demo;
import java.util.function.Predicate;
public class TestDemo {
    public static void main(String[] args) {
        Predicate<String> pre = "yootk"::equalsIgnoreCase;
        System.out.println(pre.test("YOOTK"));
    }
}
```

程序执行结果: true

本程序直接将 `String` 类的 `equalsIgnoreCase()` 普通方法利用断言型接口进行引用, 然后进行忽略大小写比较。

本章小结

1. Java 新特性中提供了可变参数, 这样在传递参数时就可以不用受到参数的个数限制, 全部的参数将以数组的形式保存下来。
2. `foreach` 是 Java 中的新特性, 主要目的是方便地输出数或集合组中的内容。
3. 泛型可以使程序的操作更加安全, 避免发生类转换异常。
4. 在程序中如果使用类时没有指定泛型, 泛型将被擦除掉, 将使用 `Object` 接收参数。
5. 可以使用通配符 “?” 接收全部的泛型类型对象。
6. 通过 `<? extends 类>` 可以设置泛型的上限, 通过 `<? super 类>` 可以设置泛型的下限。
7. 泛型方法可以定义在泛型类中, 也可以定义在普通类中。
8. 泛型也可以在接口中定义, 实现泛型接口的子类要指明具体的泛型类型。
9. 在程序中可以使用一个枚举来指定对象的取值范围, 枚举就相当于一种简化版的多例设计模式。
10. 在 Java 中使用 `enum` 关键字定义一个枚举类, 每一个枚举类都继承 `Enum` 类。
11. 在枚举中可以通过 `values()` 方法取得枚举中的全部内容。
12. 在枚举类中可以定义构造方法, 则在设置枚举范围时必须显式地调用构造方法。
13. 一个枚举类可以实现一个接口或者直接定义一个抽象方法, 但是每个枚举对象都必须分别实现全部的抽象方法。
14. `Annotation` 是 JDK 1.5 之后新增的功能, 主要是使用注释的形式进行程序的开发。
15. 在系统中提供了 3 个内建的 `Annotation`: `@Override`、`@Deprecated`、`@SuppressWarnings`。
16. `Lambda` 表达式可以有效解决匿名内部类定义复杂的问题, 使用更简单的函数式语句即可实现函数式接口 (函数式接口可以使用 “`@FunctionalInterface`” 注解定义)。
17. 利用方法引用与函数式接口的概念, 可以为方法定义别名, 也可以简化 `Lambda` 表达式编程的重复性。

18. 从 JDK 1.8 开始, 提供了 `java.util.function` 开发包, 其中提供了 4 个核心的内建函数式接口: `Function`、`Consumer`、`Supplier`、`Predicate` 供用户直接使用。

课后习题

一、填空题

1. 在使用泛型类时, 没有指定泛型的类型, 则编译会出现_____信息, 程序在使用时会使用_____类型进行接收。
2. 通过_____指定泛型的上限, 通过_____指定泛型的下限。
3. 使用_____通配符可以接收全部的泛型类型实例, 但却不可修改泛型属性内容。
4. Java 通过_____关键字定义一个枚举, 使用此关键字实际上就相当于一个类继承_____。
5. 枚举中通过_____方法取得枚举的全部内容。
6. Java 提供的三个内建的 Annotation 是_____、_____和_____。
7. 从 JDK 1.8 之中, 接口内可以定义三类方法: _____、_____和_____。
8. Java 提供了四个函数式接口, 分别为: _____、_____、_____和_____。

二、判断题

1. 在枚举类中可以定义抽象方法, 而抽象方法只需要实现一次即可。 ()
2. 枚举中可以定义构造方法, 但要求每个枚举对象都必须调用此构造方法。 ()
3. 枚举中定义的构造方法可以使用 `public` 权限声明。 ()
4. 作为函数式接口, 里面可以定义任意多个抽象方法。 ()

三、简答题

1. 简述泛型的作用。
2. 简述枚举的作用及实现特点。
3. 简述 Java SE 中三个内建 Annotation 的作用。

四、编程题

定义一个品牌计算机的枚举类, 里面只有固定的几个计算机品牌, 例如: `Lenovo`、`HP`、`Dell`、`Apple`、`Acer`。