# 基础题

### 练习一：函数式接口

1. 定义一个函数式接口CurrentTimePrinter,其中抽象方法void printCurrentTime()，使用注解@FunctionalInterface
2. 在测试类中定义static void showLongTime(CurrentTimePrinter timePrinter)，该方法的预期行为是使用timePrinter打印系统当前毫秒值
3. 测试showLongTime(),通过lambda表达式完成需求

答案

TimePrinter接口：

@FunctionalInterface  
public interface CurrentTimePrinter

{  
 void printCurrenTime();  
}

测试类：

public class Test01 {  
 public static void main(String[] args) {  
 showLongTime(()->System.out.println(System.currentTimeMillis()));  
 }  
  
 public static void showLongTime(CurrentTimePrinter timePrinter){  
 timePrinter.printCurrentTime();  
 }  
}

### 练习二：函数式接口

1. 定义一个函数式接口IntCalc,其中抽象方法int calc(int a , int b)，使用注解@FunctionalInterface
2. 在测试类中定义static void getProduct(int a , int b ,IntCalc calc), 该方法的预期行为是使用calc得到a和b的乘积并打印结果
3. 测试getProduct(),通过lambda表达式完成需求

答案

IntCalc接口：

@FunctionalInterface  
public interface IntCalc {  
 int calc(int a, int b);  
}

测试类：

public class Test02 {  
 public static void main(String[] args) {  
 getProduct(2,3,(a,b)->a\*b);  
 }  
 public static void getProduct(int a, int b, IntCalc intCalc){  
 int product = intCalc.calc(a,b);  
 System.out.println(product);  
  
 }  
}

### 练习三：静态方法引用

1. 定义一个函数式接口NumberToString,其中抽象方法String convert(int num)，使用注解@FunctionalInterface
2. 在测试类中定义static void decToHex(int num ,NumberToString nts), 该方法的预期行为是使用nts将一个十进制整数转换成十六进制表示的字符串，***tips:已知该行为与Integer类中的toHexString方法一致***
3. 测试decToHex (),使用方法引用完成需求

答案

NumberToString接口：

@FunctionalInterface  
public interface NumberToString {  
 String convert(int num);  
}

测试类：

public class Test03 {  
 public static void main(String[] args) {  
 decToHex(999, Integer::toHexString);  
 }  
 public static void decToHex(int num ,NumberToString nts){  
 String convert = nts.convert(num);  
 System.out.println(convert);  
 }  
}