

北京圣思园科技有限公司第一期 学员编程竞赛



竞赛时间：4 小时（14:00—18:00）

1. 通过命令行输入一串字符，首先判断这些字符是否都为数字。如果其中有字符不是数字，那么直接在命令行上输出“用户输入的字符不都为数字，无法转换”，程序结束；如果用户输入的字符都为数字，那么将该数字转换为中文的大写表示（用户每次最多输入 8 个字符）。

（本题目占据总成绩的 50 分）

输入输出示例：

输入：1234567

输出：一百二十三万四千五百六十七

输入：0001

输出：一

输入：1001

输出：一千零一

输入：1100

输出：一千一百

输入：10100

输出：一万零一百

输入：0010

输出：十

输入：00000001

输出：一

输入：10000001

输出：一千万零一

输入：10000010

输出：一千万零十

输入：10000100

输出：一千万零一百

输入：11000000

输出：一千一百万

输入：10100000

输出：一千零十万

输入：10010000

输出：一千零一万

输入：10001000

输出：一千万零一千

输入：a00045

输出：用户输入的字符不都为数字，无法转换

输入：a

输出：用户输入的字符不都为数字，无法转换

2. 编写一个小型的应用程序框架，该框架会向用户公开一个 `run` 方法，方法声明为：

```
public void run(String className);
```

该方法位于类 `ApplicationRun` 类中，该类的声明为：

```
1 public class ApplicationRun
2 {
3     public void run(String className)
4     {
5         //这里是方法的执行代码
6     }
7 }
```

其中的字符串参数 `className` 为用户提供的的一个类的全名（包名+类名），当用户将类的全名以字符串的形式传递给该 `run` 方法时，`run` 方法会自动执行用户所提供的类中的所有被 **@Test** 注解所修饰的 **public void** 且不带参数的方法。

说明：**@Test** 注解为该小型应用程序框架所定义的，用户可以使用该注解修饰自己的方法，同时该**@Test** 注解只能用于修饰**方法**。

程序示范：

假如用户自己定义的类为 `com.shengsiyuan.MyClass`，且该类的定义如下所示：

```
1 package com.shengsiyuan;
2
3 public class MyClass
4 {
5     public void method1()
6     {
7         System.out.println("method1");
8     }
9
10    @Test
11    public void method2()
12    {
13        System.out.println("method2");
14    }
15
16    @Test
17    public int add(int a, int b)
18    {
19        return a + b;
20    }
21
22    @Test
23    public void doSomething(String str)
24    {
25        System.out.println(str);
26    }
27
28    @Test
29    public void doSomething2()
30    {
31        System.out.println("doSomething2 ()");
32    }
33
34 }
```

那么当用户调用框架提供的 `run` 方法时应该向 `run` 方法提供参数 `com.shengsiyuan.MyClass` 字符串，结果输出如下所示：

method2

doSomething2()

理由为：

1. method1 方法没有被@Test 注解修饰
2. add 方法接受了参数并且有返回值
3. doSomething 方法接受了参数
4. method2 方法为 public void 且不接收参数，同时被@Test 注解修饰
5. doSomething2 方法为 public void 且不接收参数，同时被@Test 注解修饰

（本题目占据总成绩的 50 分）

注意：编写的代码应有良好的代码风格，在重要的类、方法或语句处添加注释说明