# 实验三：适配器模式

## 实验内容

被适配的类可以把一个Calendar类型的日期按照类似于“2020年11月25日”格式输出。

现输入类似“2020.11.2”的内容，使用对象适配器和类适配器完成按照格式的输出。

参考代码下载：[https://gitee.com/liangsan/sa-adapter.git](https://gitee.com/liangsan/sa-factory.git)，内容可能一直有变化，请注意更新。

类名、包名、项目名等请参考，可以适当调整，但不要随意乱命名。

格式等参照默认的规范，细节这里就不在仔细描述了。

## 对象适配器模式

### UML设计



图1 对象适配器类图

### 核心代码

Target.java核心代码

void printDate(String dateString);

ObjectAdapter.java核心代码

public void printDate(String calendarString) {

//将string转成三个int

String[] calendarStrings = calendarString.split("\\.");

int year = Integer.parseInt(calendarStrings[0]);

int month = Integer.parseInt(calendarStrings[1]);

int day = Integer.parseInt(calendarStrings[2]);

//成为对象

Adaptee adaptee = new Adaptee();

System.out.println("对象适配器：");

adaptee.printCalendar(year,month,day);

}

Adaptee.java核心代码

public void printCalendar(int year,int month,int day) {

System.out.println("你好，我是被适配的类，你输入的日期是："

+ year + "年"

+month +"月"

+ day +"日");

}

### 实现效果

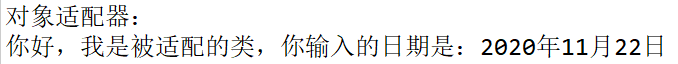


图2 对象适配器实现效果图

## 类适配器模式

### UML设计



图3 类适配器类图

### 核心代码

ClassAdapter.java核心代码

public void printDate(String calendarString) {

// 将calendarString转换为三个int

String[] calendarStrings = calendarString.split("\\.");

int year = Integer.parseInt(calendarStrings[0]);

int month = Integer.parseInt(calendarStrings[1]);

int day = Integer.parseInt(calendarStrings[2]);

//继承

System.out.println("类适配器：");

printCalendar(year,month,day);

}

Adaptee.java核心代码

public void printCalendar(int year,int month,int day) {

System.out.println("你好，我是被适配的类，你输入的日期是："

+ year + "年"

+month +"月"

+ day +"日");

}

Target.java核心代码

void printDate(String dateString);

### 实现效果

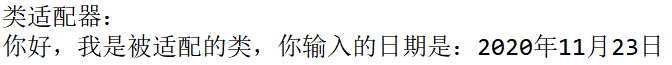


图4 类适配器实现效果图

## 实验体会