```
package com.interpreter.extral;
import java.util.*;
/*文法规则
  <expression> ::= <city>的<person>
  <city> ::= 韶关|广州
  <person> ::= 老人|妇女|儿童
*/
public class InterpreterPatternDemo
{
    public static void main(String[] args)
    {
        Context bus=new Context();
        bus.freeRide("韶关的老人");
        bus.freeRide("韶关的年轻人");
        bus.freeRide("广州的妇女");
        bus.freeRide("广州的儿童");
        bus.freeRide("山东的儿童");
    }
}
//抽象表达式类
interface Expression
{
    public boolean interpret(String info);
}
//终结符表达式类
class TerminalExpression implements Expression
{
    private Set<String> set= new HashSet<String>();
    public TerminalExpression(String[] data)
    {
        for(int i=0;i<data.length;i++)set.add(data[i]);
    }
    public boolean interpret(String info)
    {
        if(set.contains(info))
        {
            return true;
        }
        return false;
    }
}
//非终结符表达式类
class AndExpression implements Expression
{
    private Expression city=null;
    private Expression person=null;
    public AndExpression(Expression city,Expression person)
    {
        this.city=city;
        this.person=person;
    }
    public boolean interpret(String info)
    {
        String s[]=info.split("的");
        return city.interpret(s[0])&&person.interpret(s[1]);
    }
}
//环境类
class Context
{
    private String[] citys={"韶关","广州"};
    private String[] persons={"老人","妇女","儿童"};
    private Expression cityPerson;
    public Context()
    {
        Expression city=new TerminalExpression(citys);
        Expression person=new TerminalExpression(persons);
        cityPerson=new AndExpression(city,person);
    }
    public void freeRide(String info)
    {
        boolean ok=cityPerson.interpret(info);
        if(ok) System.out.println("您是"+info+"，您本次乘车免费！");
        else System.out.println(info+"，您不是免费人员，本次乘车扣费2元！");
    }
}
```