

## Java 程序设计文档

### LSP 原则论证

新增的特殊出租车 `TraceableTaxi` 类在 `Taxi` 类的基础上添加了如下数据：

```
private ArrayList<Record> records;
```

```
private int preIndex;
```

```
private int nextIndex;
```

这三个数据只是保存了可追踪出租车的历史接单信息，普通出租车不会用到，显然不会对父类对象的行为造成影响。

继承的 `TraceableTaxi` 类重写了 `repOK()` 方法和 `recordServeInfo()` 方法，其它写出的方法均为新增的方法，`Taxi` 类中不存在。

**repOK():** 在保证子类数据有效时，也就是当子类的各项属性合法时，`repOK` 的有效性只和父类数据有关，所以与父类 `repOK` 的值是一致的。

**recordServeInfo():** 父类的这个方法是空方法，子类中该方法会将完成的订单记录在 `records` 数组中，如果父类也执行该方法，不会产生对出租车现有的运行和接单调度产生任何影响。

其余新增的五个方法均为对子类中 `records` 数组的访问操作，只涉及子类属性的更改，同样不会产生对出租车现有的运行和接单调度等产生任何影响。

所以由上分析，任何父类出现的地方都可以使用子类来代替，子类型能够完全替换父类型，而不会让调用父类型的客户程序从行为上有任何改变。