类和对象对于初学者来说是难点，主要难在概念多，知识点抽象。对于这部分的学习会让很多同学们怀疑人生，有这种感觉是正常的，不要灰心。

类和对象原本就不是一朝一夕就能学懂的，一定是在长期的学习和工作中领悟出来的。

为了让同学们学起来更轻松，本文从**应用**的角度阐述类和对象及其关系。

下表是一份简化了的商品订单，订单中购买了3种商品，每个商品包含商品名称，单价，数量。

商品名称	单价	数量
LED灯带	25	2
扩展坞	39	1
电子表	39	1

在电商系统开发中，这些订单是要保存到数据库中的。因此需要在数据库中创建订单表，来存储订单信息。我们以MySQL数据库为例，创建订单表如下

```
CREATE TABLE `order_details` (
  `product_name` VARCHAR(50) DEFAULT NULL,
  `price` DOUBLE DEFAULT NULL,
  `number` INT DEFAULT NULL
) ;
```

然后插入订单信息

```
INSERT INTO order_details(product_name,price,number) VALUES ('电子表',39,1);
INSERT INTO order_details(product_name,price,number) VALUES ('扩展坞',39,1);
INSERT INTO order_details(product_name,price,number) VALUES ('LED灯带',25,2);
```

现在查询订单表，结果如下：

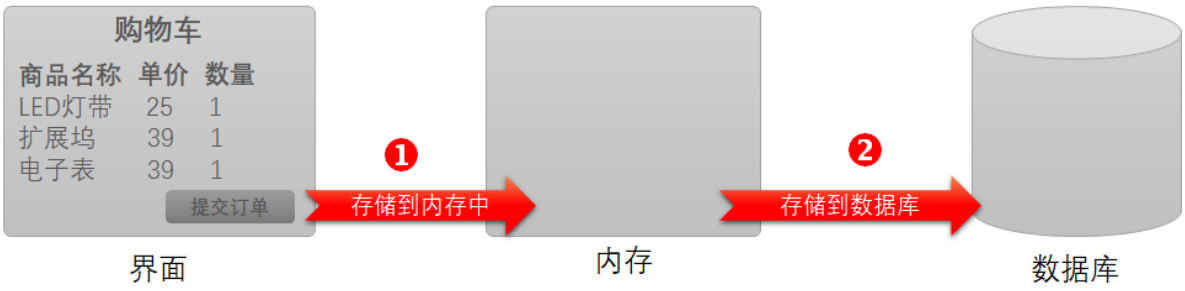
product_name	price	number
电子表	39	1
扩展坞	39	1
LED灯带	25	2

你或许有疑问，本文不是讲类和对象吗？干嘛从数据表开始讲起呢？

数据表与类和对象当然是有联系的，你接着往下看！

下订单的功能，实际上就是将顾客在订单页面中选择的商品信息保存到数据库中，下单就完成了。比如，用户在商品页面中选择了本例中的电子表、扩展坞、LED灯带，然后提交订单。

从开发的角度来说，提交订单的流程是这样的：



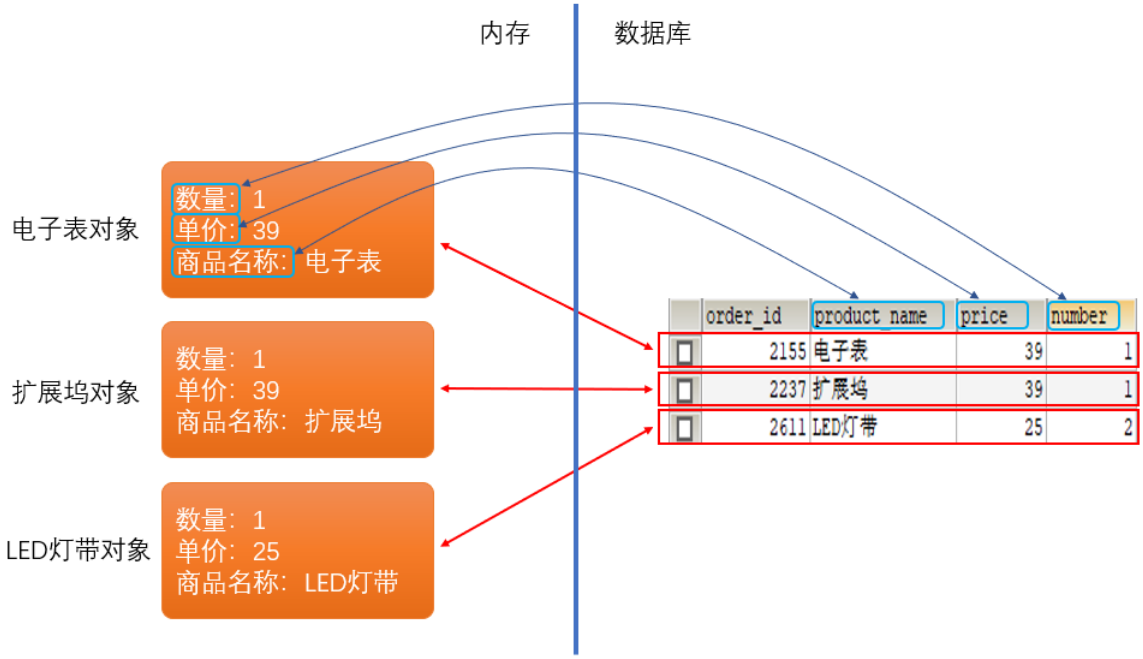
- 流程1：先将订单信息保存到内存中
- 流程2：再将内存中的订单信息保存到数据库中

你会发现内存起到了临时存储订单信息的作用，内存建立起了界面和数据库的桥梁

我们知道数据库中是用 **表** 来存储订单信息的，那么**内存中如何存储订单信息呢**？

重点来了！

内存中是用对象存储订单信息的，订单中的每一个商品就是一个对象，每一个对象对应订单表中的一条记录，如下图：



对象和表有两种对应关系，专业术语叫做对象关系映射（ORM：Object Realtionship Mapping）

- 对象 映射为 记录，例如 **电子表对象** 映射为数据表中的第一条 **记录**
- 属性 映射为 字段，例如对象的 **商品名称** 属性映射为表中的 product_name 列

接下来你需要思考一个问题，内存中的三个对象是如何创建的呢?

答案是通过类创建的对象。

要想创建对象，必须先创建类，然后通过类实例化出对象。

如何创建类呢?

此时，你想一下类的概念是什么!

类是具有相同状态和相同行为的一组对象的集合。

- 状态就是属性，用变量表示
- 行为就是方法

所以，类是由属性和方法构成的。

```java
// 类是由属性和方法构成的
public class ClassName{
    // 类的属性
    int age;

    // 类的方法
    void method(){

    }
}
```

本例中的三个商品都有 商品名称，单价，数量 三个属性，因此属于同一类。

分析到这里，订单类就出现了

```java
public class OrderDetails {
    /******** 类的属性 ********/
    String productName; //商品名称属性
    double price;       //商品单价属性
    int number;         //商品数量属性
}
```

类和表的对应关系

- 类 映射为 表

如何通过类创建对象呢?

创建对象的方法是**实例化**，通过实例化产生一个**实例**，**实例就是对象**

现在我们通过实例化来创建电子表、扩展坞、LED灯带三个对象

```java
public class OrderDetails {
    /******** 类的属性 ********/
    String productName; //商品名称属性
    double price;       //商品单价属性
    int number;         //商品数量属性

    /******** 类的方法 ********/
    public static void main(String[] args) {
        //创建电子表对象
        OrderDetails dianzibiao = new OrderDetails();
        dianzibiao.productName = "电子表";
```

```
        dianzibiao.price= 39;
        dianzibiao.number = 1;

        //创建扩展坞对象
        OrderDetails kuozhanwu = new OrderDetails();
        kuozhanwu.productName="扩展坞";
        kuozhanwu.price = 39;
        kuozhanwu.number = 1;

        //创建灯带对象
        OrderDetails dengdai = new OrderDetails();
        dengdai.productName="LED灯带";
        dengdai.price = 25;
        dengdai.number = 1;
    }
}
```

- new 关键字的作用就是实例化
- new 的本质的分配内存

下面一行代码是实例化，看看是如何分配内存的

```
OrderDetails dianzibiao = new OrderDetails();
```

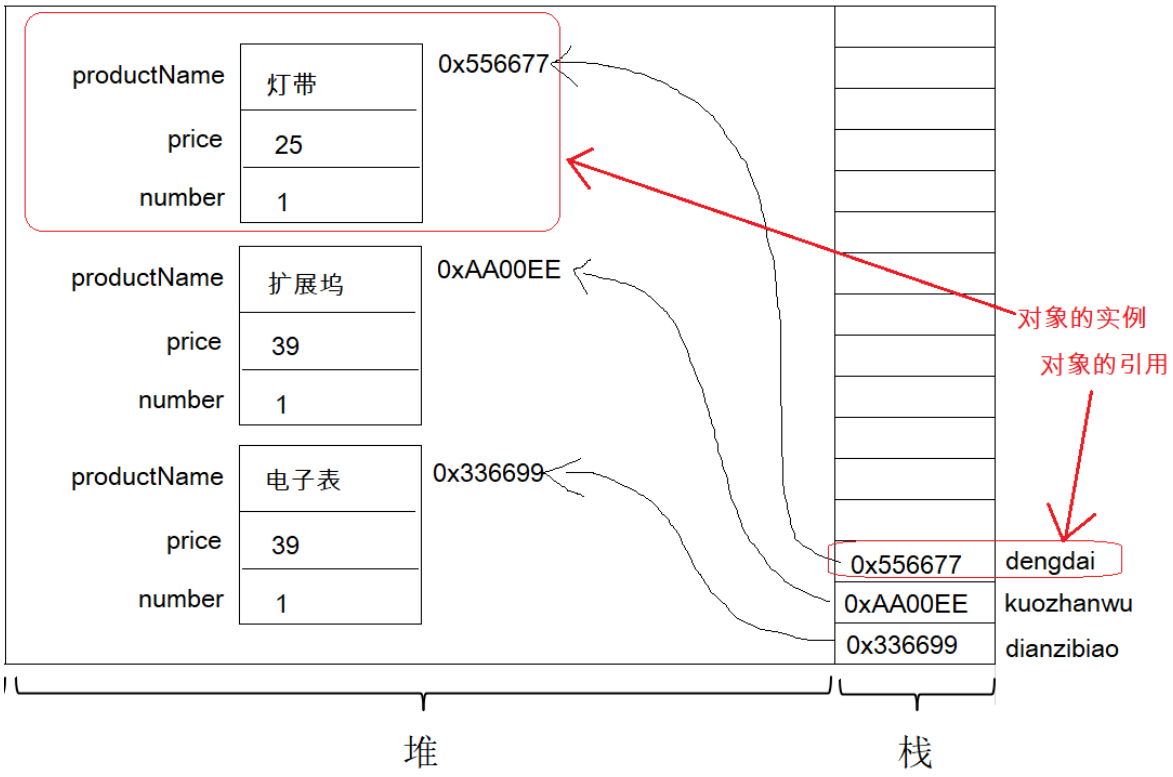
首先要知道**对象**由**对象的引用**和**对象的实例**构成

- = 左边的 变量名`dianzibiao` 就是对象的引用，对象的引用分配在栈中
- = 右边的 通过new 分配的内存就是对象的实例，对象的实例分配在堆中

其次要知道new是在给谁分配内存，分配多大内存

- new就是给实例的属性分配内存
- 实例有多少属性，就分配多少内存
- OrderDetails类中定义了3个属性，（productName，price，number）所以new就是给这三个属性分配内存

三个商品实例化的内存图如下：



这幅图展示了创建三个对象，对象的引用分配在栈中，对象的实例分配在堆中，对象的引用指向对象的实例。

当执行 dianzibiao.price = 39 时，就是将39赋值到 dianzibiao指向的实例中的price属性中。

总结一下：

- 什么是类：类是具有相同状态和相同行为的一组对象的集合
- 什么是对象：对象是类的实例，通过实例化类产生对象
- 类和对象的关系：
  - 类是对象的模板。对象是类的实例
  - 类表示抽象概念，不是具体的个体
  - 对象表示具体概念，表示具体的个体