**الگوی Visitor یا**

**عملکرد : یک Operation یا عملیات مشخص را از درون اشیا مختلف استخراج کند و در یک کلاس قرار دهد و در آن کلاس بتوان عملیات هایی که هر کدام مرتبط با یک شی هستند را یکجا تجمیع و آن را پردازش کنیم.**

**برای مثال:**

**فرض کنیم میخواهیم یک سوپر اپلیکیشن را توسعه بدهیم(مثل اسنپ که دارای سرویس های مختلفی از قبیل تاکسی، اسنپ تریپ،اسنپ مارکت، اسنپ فود،اسنپ باکس و ...)**

**همه سرویس ها دارای کلاسی بنام Order بصورت مجزا هستند.**

**یرای مثال بیشتر کلاس های زیر را در نظر بگیرید:**

**public class TaxiOrder**

**{**

**public void Operation(){}**

**}**

**public class MarketOrder**

**{**

**public void Operation(){}**

**}**

**public class FoodOrder**

**{**

**public void Operation(){}**

**}**

**مشکل این روش این است که وقتی تعداد کلاس های ما زیادتر میشود ما عمل Operation را به ازای هر شی انجام میدهیم و چون تعداد اشیا خیلی زیاد میشود و نمیتوانیم بین همه اشیا پیمایش کنیم.**

**راه حل دیگر این است که از Interface استفاده کنیم:**

**public interface IOrder {**

**void operation();**

**}**

**public class TaxiOrder:IOrder**

**{**

**public void Operation(){}**

**}**

**public class MarketOrder:IOrder**

**{**

**public void Operation(){}**

**}**

**public class FoodOrder:IOrder**

**{**

**public void Operation(){}**

**}**

**مشکل این روش : توسعه و نگهداری را مشکل ساز میکند،مثال تغییر خروجی operation این تغییر باید در همه کلاس ها اعمال شود.**