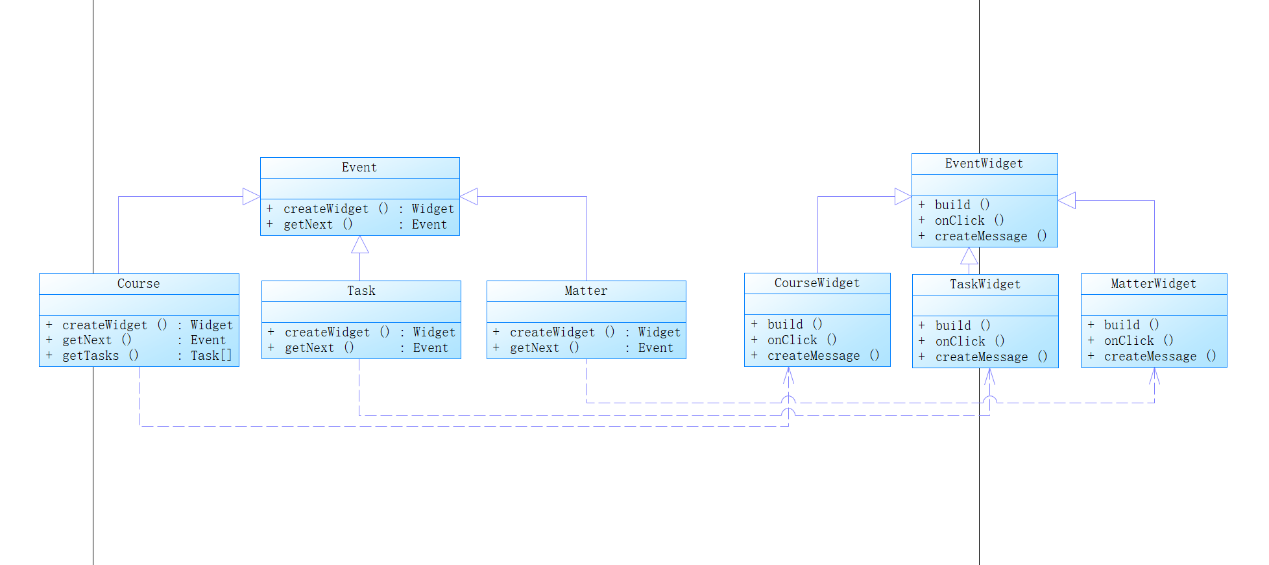
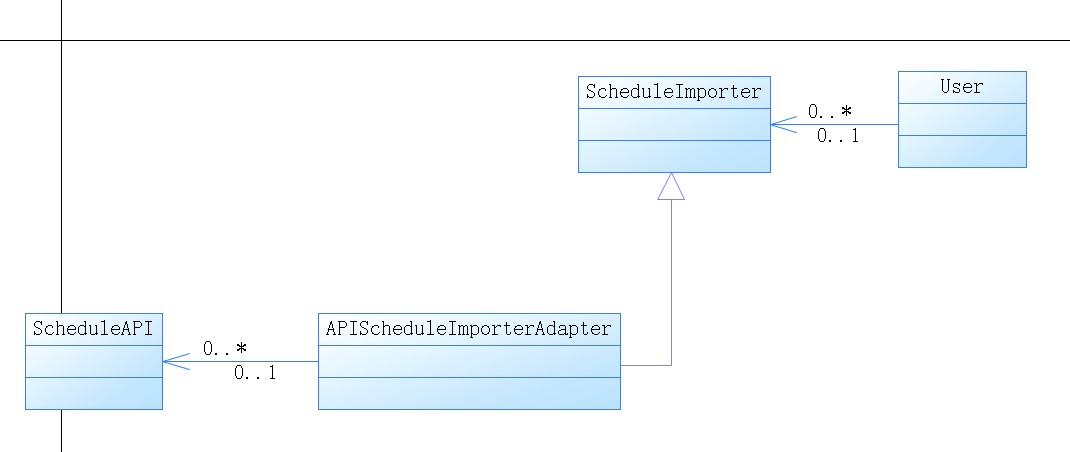
1. 工厂方法Factory Method



在课程表管理软件中，需要构造很多事件类来组织计划表。除了课程类Course以外，还有其他的事件类如任务类Task、事务类Matter，而这些类都是抽象类Event的子类。在前端框架中，最重要的就是利用事件类及其子类的工厂方法创建出具体的Widget视图。Event作为抽象工厂，仅需提供一些必须实现的接口，而不需要参与创建具体产品的过程；Course等子类为具体工厂，需要实现父类接口，并能够创建出对应的具体产品。EventWidget为抽象产品，提供一些发挥具体作用的接口；其子类的各种视图为具体产品，需要实现与对应具体工厂相关的能力与效果。

使用工厂方法的优点：多态性，管理子类时无需关心具体类，可统一管理父类，适用性更强；封装并继承或实现父类的方法，或者还可以增加独有方法，体现灵活性；扩展性好，在增加一个产品类之前，只需要扩展出对应的工厂类即可。

1. 适配器模式



* **ScheduleImporter（目标接口）**：这是客户端期望使用的接口。在本例中，用户通过它导入课表。
* **ScheduleAPI（被适配者）**：这是教务系统的课表API，它有自己的方法来获取课表数据。
* **APIScheduleImporterAdapter（适配器）**：这是适配器类，它实现了•ScheduleImporter接口，并可以调用**ScheduleAPI**，将API返回的数据转换为客户端可以理解的Timetable对象格式。

通过使用适配器模式，现有系统可以无缝地使用教务系统的课表API，而无需修改现有代码。