**كلية الهندسة المعلوماتية**



**السنة الخامسة**

**قسم البرمجيات**

**وظيفة هندسة البرمجيات -3-**

**Design Pattern**

**المهندس**

**روان قرعوني**

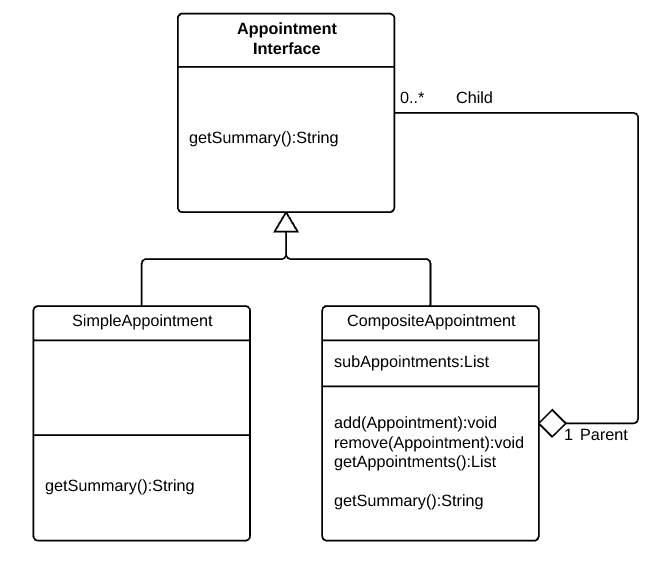
**الطلاب**

**الطلب الأول:**

الموعد يكون لفعالية رئيسية وضمن هذه الفعالية الرئيسية يوجد مواعيد لفعاليات جزئية، وبالتالي نحنا بحاجة إلى تمثيل الموعد ببنية هرمية (Hierarchical representations) بحيث يمكن للموعد الواحد يحوي مواعيد أخرى.

موعد الفعالية الرئيسية لا يختلف من حيث طريقة المعالجة والتعامل عن موعد الفعالية الجزئية وبالتالي نحنا بحاجة إلى بنية موحدة للتعامل مع المواعيد الجزئية والمواعيد الرئيسية.

بحسب ما سبق يكون مخطط الصفوف الذي يعبر عن الموعد كما يلي:



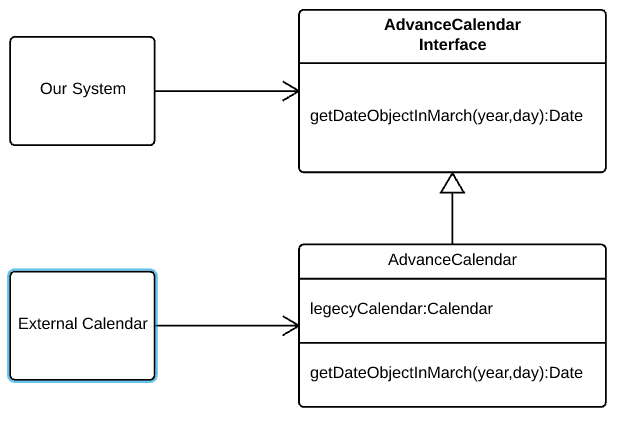
الصف المجرد Appointment بحيث يحوي الموعد على توقيت وتاريخ وإجرائية طباعة تفاصيل الموعد.، الصف CompositeAppointment يرث الصف المجرد ويحوي قائمة مواعيد، وهي التي تعبر عن المواعيد الجزئية، الصف SimpleAppointment يرث الصف المجرد.

مخطط الصفوف هذا يطابق نموذج التصميم Composite حيث أن الصف Appointment هو Compound والصف CompositeAppointment هو Composite والصف SimpeAppointment هو Leaf.

**الطلب الثاني:**

النمط التصميم المستخدم هو Adapter لأن المكتبة الخارجية تقدم لنا الوظيفة (Functionally) التي نريدها ولكن الواجهة (Interface) التي تقدمها لا تناسب احتياجاتنا. اخترنا هذا النمط لأنه مرن وعمله هو الربط وتسهيل التواصل مع الأنظمة وليس التعامل مع النظام السابق بطريقة تنفيذ جديدة.

بحسب ما سبق يكون مخطط الصفوف كما يلي:



الواجهة AdvanceCalendarInterface تعرف الطريقة التي تناسب احتياجاتنا، الصف AdvanceCalendear طبق الواجهة السابقة وهو Adapter، الواجهة External Calendar هو واجهة المكتبة الخارجية وهي Adaptee.

**الطلب الثالث:**

النمط المستخدم في هذا الطلب هو من نوع Singleton حيث عندما يريد التطبيق ان يتعامل مع التقويم نتأكد من وجود نسخة واحد وواحد فقط من التقويم من الصف حيث يحتوي عل باني private ودوماً يرد نفس النسخة عن طريق Static Method وبذلك نضمن وجود نسخة واحدة من التقويم طول فترة حياة البرنامج.

