**题目：**

请设计一个电子相册自动生成程序，要求该程序能够将一组图片自动生成到一个以树形结构保存的电子相册中，相册的目录采用树形结构，一级目录为年，二级目录为月，三级目录为日。每张照片可以添加音效和花边等特效。请写出你所选择的设计模式，画出类图，并给出核心代码。

解答：

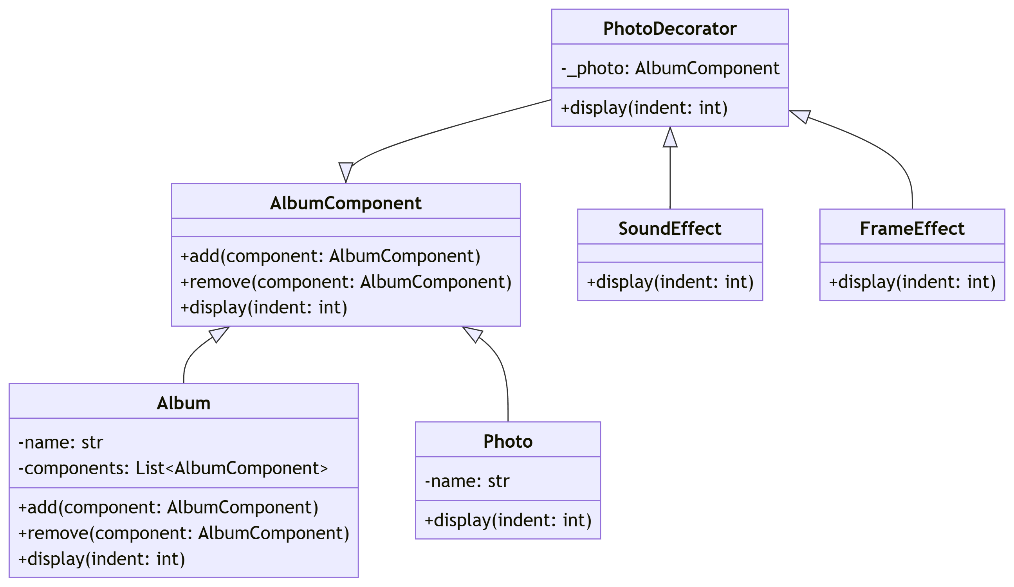
* **设计思路：**

1. 可以结合使用组合模式和装饰模式。
2. 将目录结构抽象为一个树形结构，每个目录可以包含子目录或照片。一级目录表示年，二级目录表示月，三级目录表示日。照片作为树的叶子节点。
3. 为每张照片动态添加音效、花边等特效。装饰者模式可以允许为照片对象添加多个不同的特效，而无需修改照片类。

* **模式：**

**组合模式**和**装饰模式**

* **类图：**



|  |
| --- |
| * **核心代码：** |
| from abc import ABC, abstractmethod  *# 相册组件的抽象基类*  class AlbumComponent(ABC):      def add(self, component):          raise NotImplementedError      def remove(self, component):          raise NotImplementedError      @abstractmethod      def display(self, indent: int):          pass  *# 具体的相册目录类*  class Album(AlbumComponent):      def \_\_init\_\_(self, name):          self.name = name          self.components = []      def add(self, component: AlbumComponent):          self.components.append(component)      def remove(self, component: AlbumComponent):          self.components.remove(component)      def display(self, indent: int = 0):          print(" " \* indent + f"Album: {self.name}")          for component in self.components:              component.display(indent + 2)  *# 照片类（叶子节点）*  class Photo(AlbumComponent):      def \_\_init\_\_(self, name):          self.name = name      def display(self, indent: int = 0):          print(" " \* indent + f"Photo: {self.name}")  *# 照片装饰者抽象类*  class PhotoDecorator(AlbumComponent):      def \_\_init\_\_(self, photo: AlbumComponent):          self.\_photo = photo      def display(self, indent: int = 0):          self.\_photo.display(indent)  *# 音效特效*  class SoundEffect(PhotoDecorator):      def display(self, indent: int = 0):          super().display(indent)          print(" " \* indent + "-- with Sound Effect")  *# 花边特效*  class FrameEffect(PhotoDecorator):      def display(self, indent: int = 0):          super().display(indent)          print(" " \* indent + "-- with Frame Effect") |