

الف) کلاسی **Abstract** برای کار با فایل‌ها تعریف کنید که فقط دو متد **open** و **close** دارد (فعلا این کلاس را کلاس 1 می‌نامیم. شما زمان پیاده‌سازی اسامی مناسب را برای هر کلاس انتخاب کنید.) و این دو متد پیاده‌سازی ندارند (از نظر مفهومی کلاس 1 از نوع **interface** است).

ب) مدهای مختلف بازکردن فایل را مرور کنید.

- Read Only ('r')
- Read and Write ('r+')
- Write Only ('w')
- Write and Read ('w+')
- Append Only ('a')
- Append and Read ('a+')

برای هر مد یک کلاس بنویسید که همگی از کلاس 1 ارث‌بری می‌کنند. ساختار ارث‌بری به این شکل است: کلاس مد ۲ مستقیماً از کلاس 1 ارث‌بری می‌کند. کلاس مد **r+** از مد ۲ مستقیماً ارث‌بری می‌کند. کلاس مد **w** مستقیماً از کلاس 1 ارث‌بری می‌کند. کلاس مد **w+** مستقیماً از کلاس مد **w** ارث‌بری می‌کند. کلاس مد **a** مستقیماً از کلاس 1 ارث‌بری می‌کند و کلاس مد **a+** مستقیماً از کلاس مد **a** ارث‌بری می‌کند.

فایل، ویژگی سطح **instance** این کلاس‌ها است که در هر کلاس در مد موردنظر تعریف و باز می‌شود.

نام فایل هم ویژگی سطح **instance** این کلاس‌ها است و در زمان تعریف شیء باید به‌عنوان ورودی داده شود.

ج) متد + را به‌نحوی برای این کلاس‌ها بنویسید که اگر دو شیء از یک کلاس را با هم جمع کنیم، فایل سوم شامل محتوای هر دو فایل باشد که به دنبال هم در فایل سوم نوشته شده‌اند.

د) علاوه بر امکان نوشتن در فایل به وسیله متد نوشتن که برای کلاس‌های موردنظر تعریف کرده‌اید، کلاس‌های مدهای با قابلیت نوشتن را به‌صورت **callable** نیز تعریف کنید تا متنی که قرار است در فایل بنویسند یا اضافه کنند را در پرانتز جلوی اسم شیء به آن بدهید.