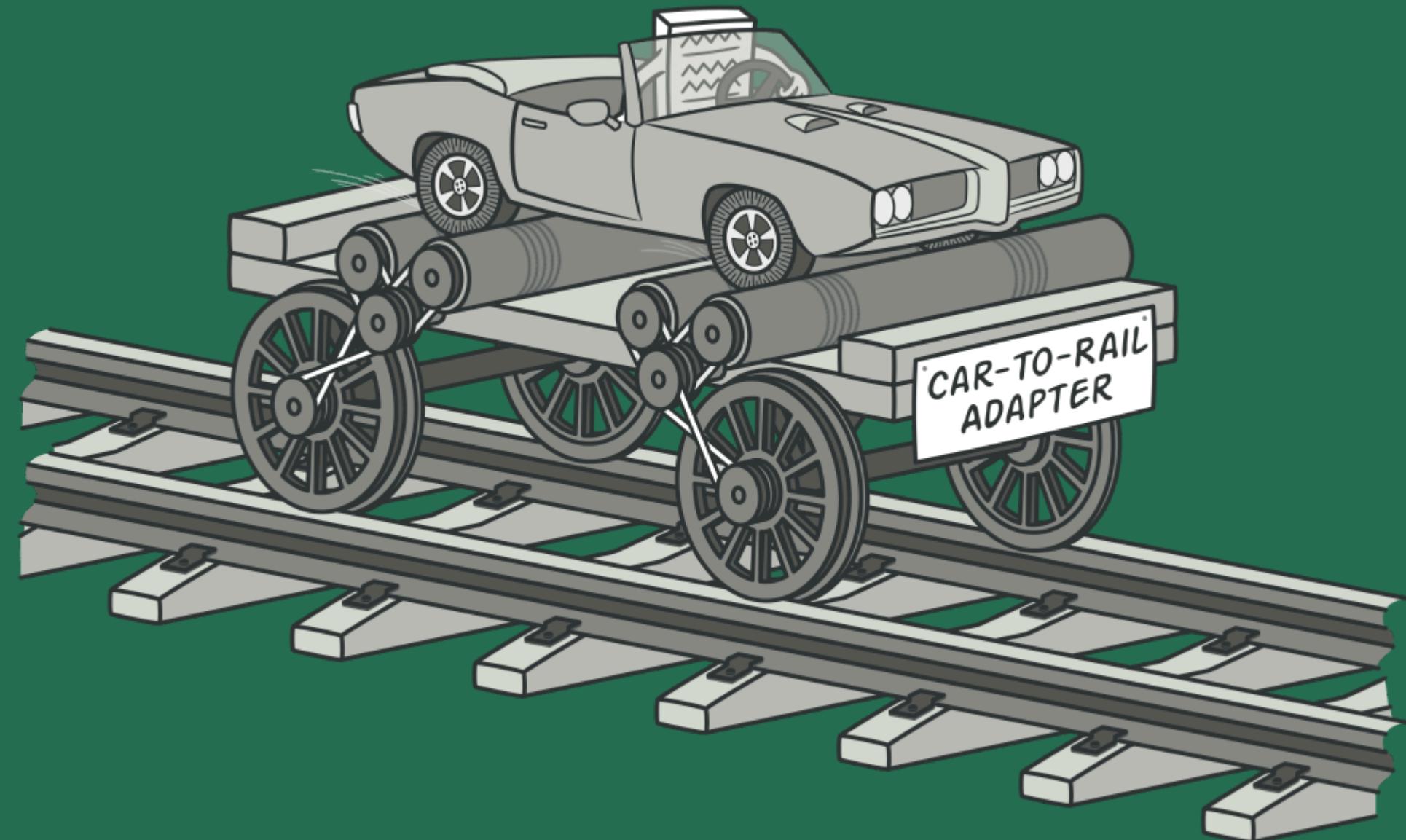


الگو آدپتريا (ADAPTER PATTERNS)



PUBLISHED BY : FATEMEH AHMADZADEH

آدابر یعنی چی

تا حالا شده دو تا وسیله داشته باشی که به هم نمی خورن؟
مثلاً شارژر لپ تاپ سه شاخه است ولی پریز اتاق دو تاست
یه تبدیل (Adapter) می زنی و همه چیز درست میشه.

در برنامه نویسی هم همین طوره!
گاهی دو کلاس یا سیستم داریم که رابطشون یکی نیست، ولی می خوای با هم کار کن.
اینجاست که Adapter Pattern وارد عمل میشه.

تعريف الگو ADAPTER

الگوی Adapter یکی از الگوهای ساختاری (Structural) هست. هدفش اینه که دو کلاس ناسازگار رو به هم وصل کنه بدون اینکه کدهاشونو تغییر بدی.

در واقع، آدپتر نقش مترجم بین دو کلاس رو بازی میکنه  یک طرف یه زبان صحبت میکنه، طرف دیگه یه زبان دیگه – بینشون ترجمه میکنه.

مثال واقعی

فرض کن یه سیستم پرداخت جدید داری که فقط تابع `processPayment(amount)` را می‌شناسه.

ولی کتابخونه قدیمی شرکت `makeTransaction(value)` داره.

راه حل

به جای بازنویسی کل برنامه، یه Adapter می‌سازی که تماس‌های makeTransaction() را به processPayment()

نتیجه: هر دو سیستم (قدیمی و جدید) بدون دردسر با هم کار می‌کنن!

مثال واقعی تر: وقتی دادههای حرف هم نمی‌فهمن

یه اپ داری که دادههای بورس رو از چند منبع به صورت XML می‌گیره.

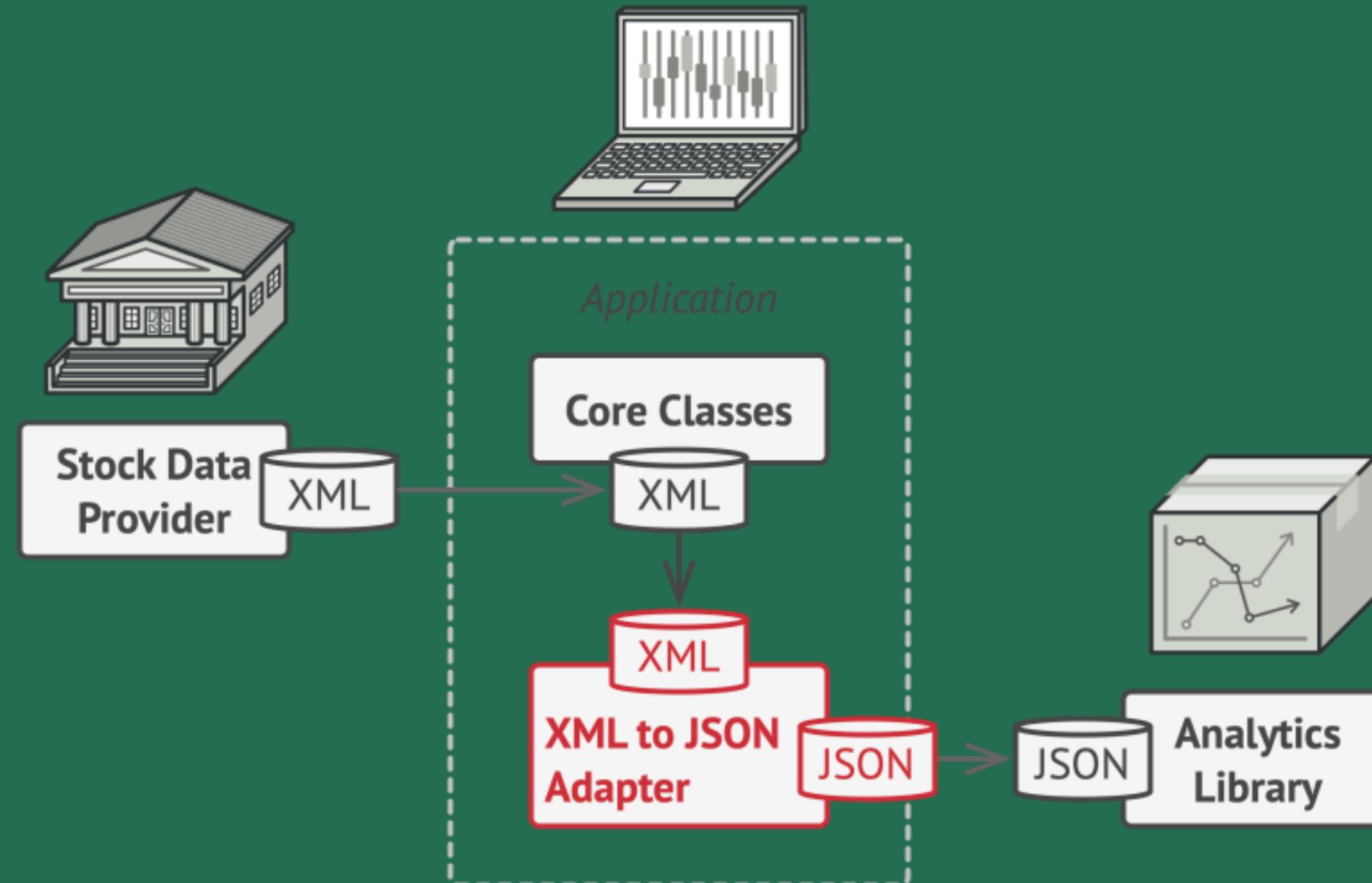
حالا من خوای یه کتابخونه آنالیز جدید اضافه کنی،
ولی اون فقط با JSON کار می‌کنه 😬

مثال واقعی تر: چه کار می کنی؟؟

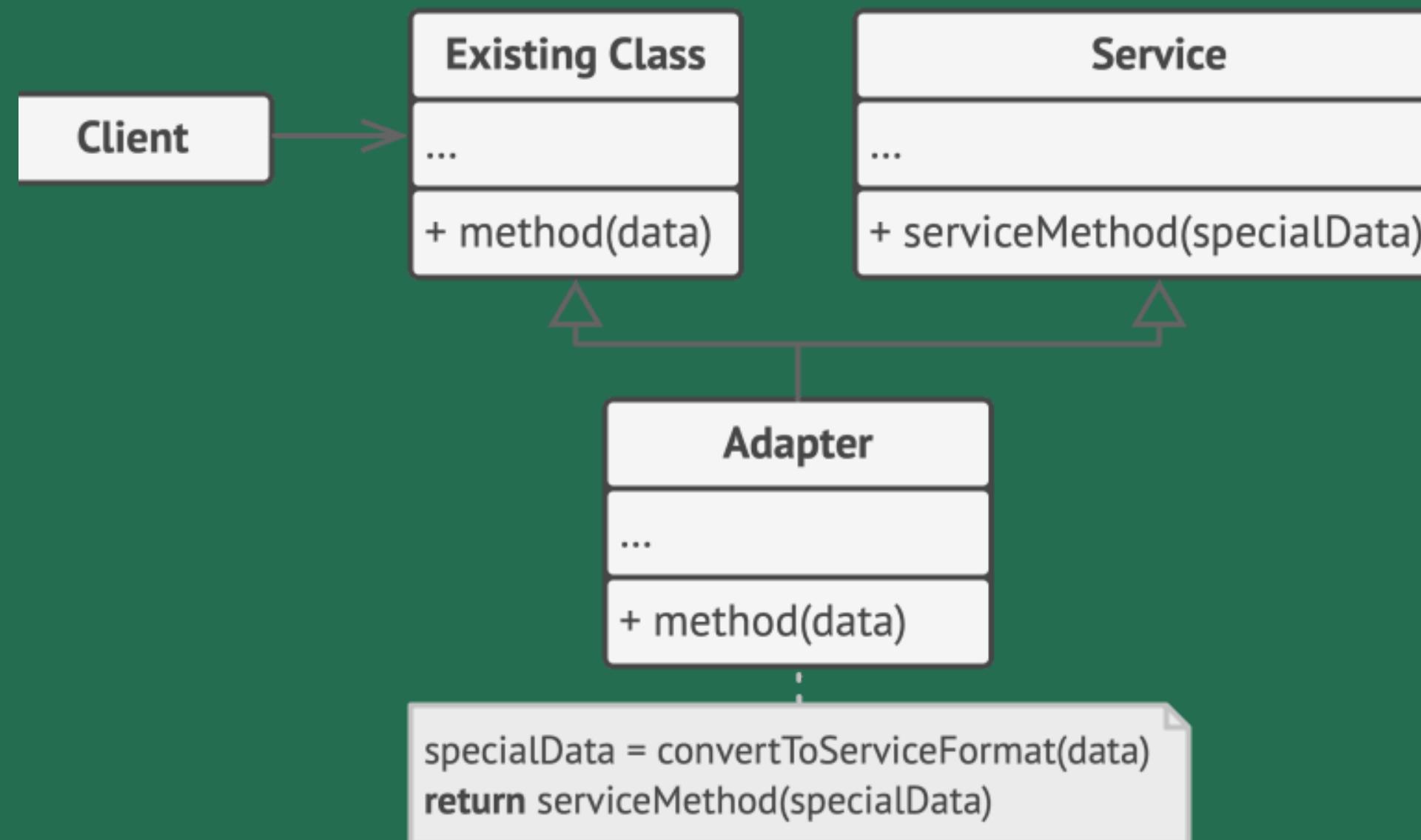
به جای تغییر کتابخونه (که حتی ممکنه کدش دست تو نباشه)،
یه Adapter می سازی که XML رو به JSON تبدیل کنه
و بین اپ و کتابخونه قرار می گیره.

نتیجه: بدون دست زدن به کدهای قدیمی، هر دو بخش با هم سازگار
می شن.

مثال واقعی تر: چه کار می کنی؟؟



ساختار کلی ADAPTER



ساختار کلی ADAPTER

✿ نسخهی کلاس محور (Class Adapter) 

از هر دو کلاس زیر ارث بری می کنی: Adapter

→ رابطی که بقیهی برنامه باهاش کار می کنی 

→ کلاسی که کار اصلی رو انجام می ده (ولی رابط متفاوتی دارد) 

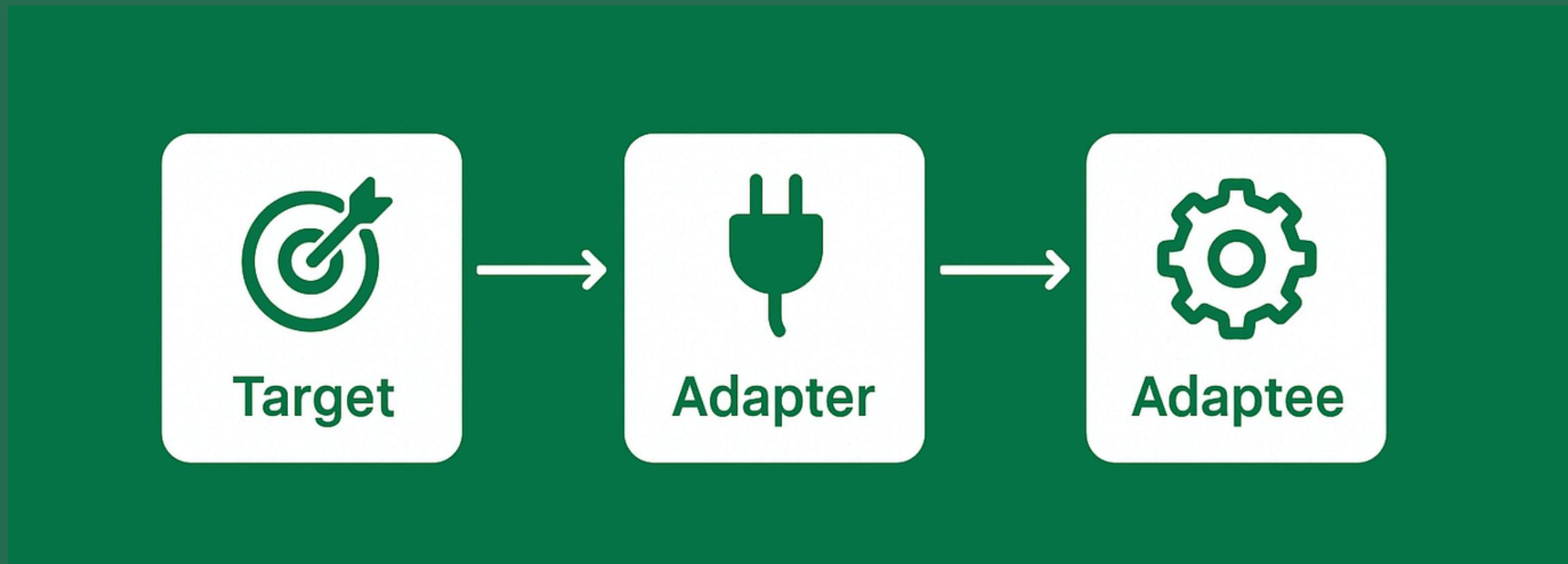
ساختار کلی ADAPTER: چرا از هر دو ارثبری میکنی؟

چون با این کار:

میتوانه مثل Target رفتار کنه (از دید بیرونی همون رابط رو
داره)

میتوانه از Adaptee کمک بگیره تا کار واقعی رو انجام بده

ارتباط بین ADAPTEE و TARGET، ADAPTER



کی باید از ADAPTER استفاده کنیم؟

✓ وقتی یه کلاس داری که من خوای استفاده اش کنی
ولی رابطه ش با بقیه هی برنامه هت سازگار نیست.

✓ وقتی چند تا زیر کلاس داری که یه قابلیت مشترک ندارند و نمی خواهی کد رو
تکرار کنی. می تونی یه Adapter بسازی که اون قابلیت رو بهشون اضافه کنه بدون
تغییر در ساختار اصلی.

خلاصه اینکه

آدایپتر یعنی "آشتی دهنده‌ی کلاس‌ها"!
😊

به جای اینکه کدهای قدیمی رو نابود کنی، یه مترجم پیشون می‌ذاری تا با هم کنار بیان.