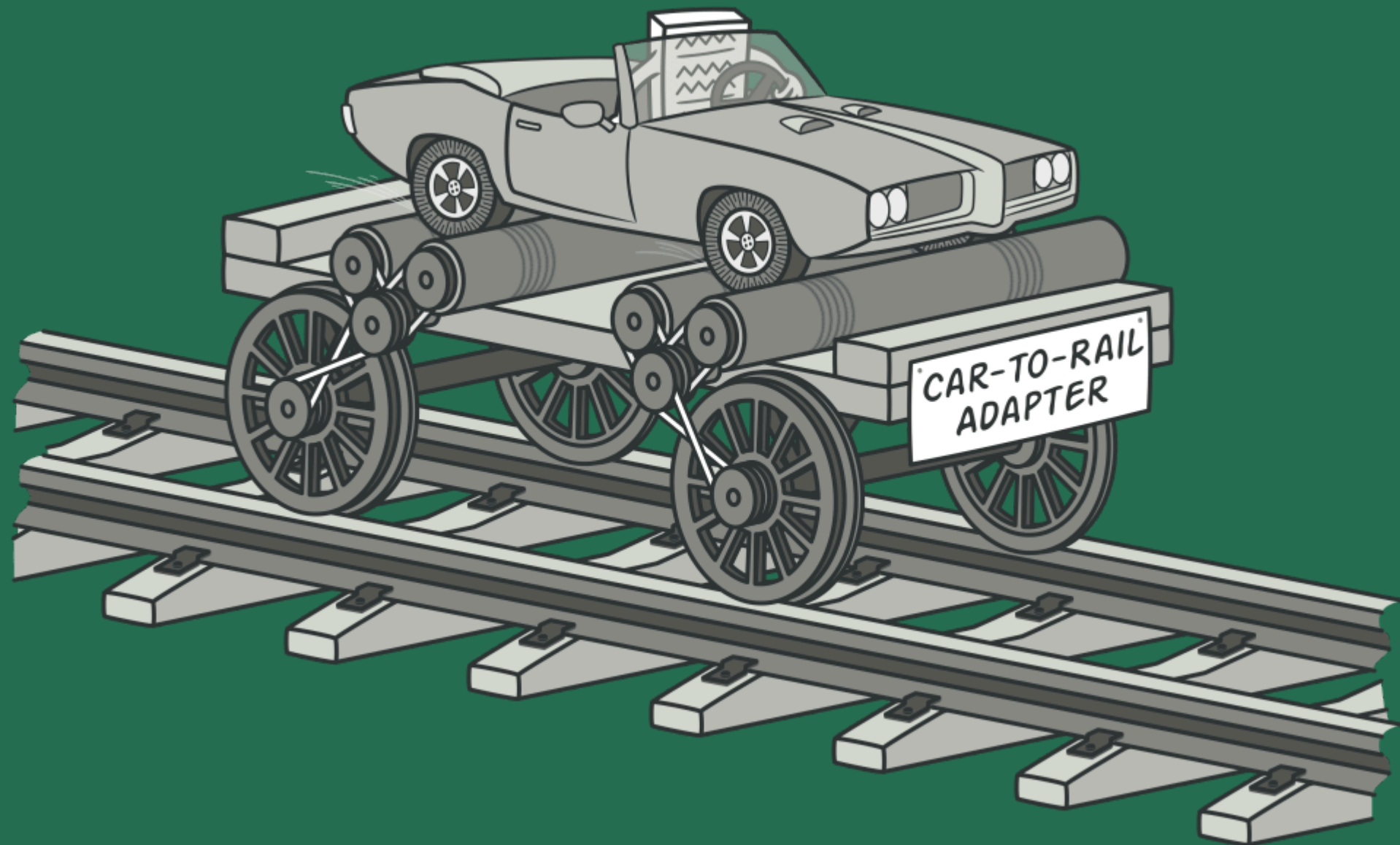


الگو آداپتري يا (ADAPTER PATTERNS)



PUBLISHED BY : FATEMEH AHMADZADEH

آداپتر یعنی چی

تا حالا شده دو تا وسیله داشته باشی که به هم نمی‌خورن؟
مثلاً شارژر لپ‌تاپت سه شاخه‌ست ولی پریز اتاقت دوتا‌ست
یه تبدیل (Adapter) می‌زنی و همه‌چیز درست می‌شه.

در برنامه‌نویسی هم همین‌طوره!
گاهی دو کلاس یا سیستم داریم که رابطشون یکی نیست، ولی می‌خوای با هم کار کنی.
اینجاست که Adapter Pattern وارد عمل می‌شه.

تعریف 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، آداپتر نقش مترجم بین دو کلاس رو بازی می‌کنه 📌
یک طرف به زبان صحبت می‌کنه، طرف دیگه به زبان دیگه – Adapter
بینشون ترجمه می‌کنه.

مثال واقعی

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

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

راه حل

به جای بازنویسی کل برنامه، به Adapter می‌سازی که تماس‌های processPayment() رو به makeTransaction() تبدیل کنه.

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

مثال واقعی تر: وقتی داده‌ها حرف همو نمی‌فهمن

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

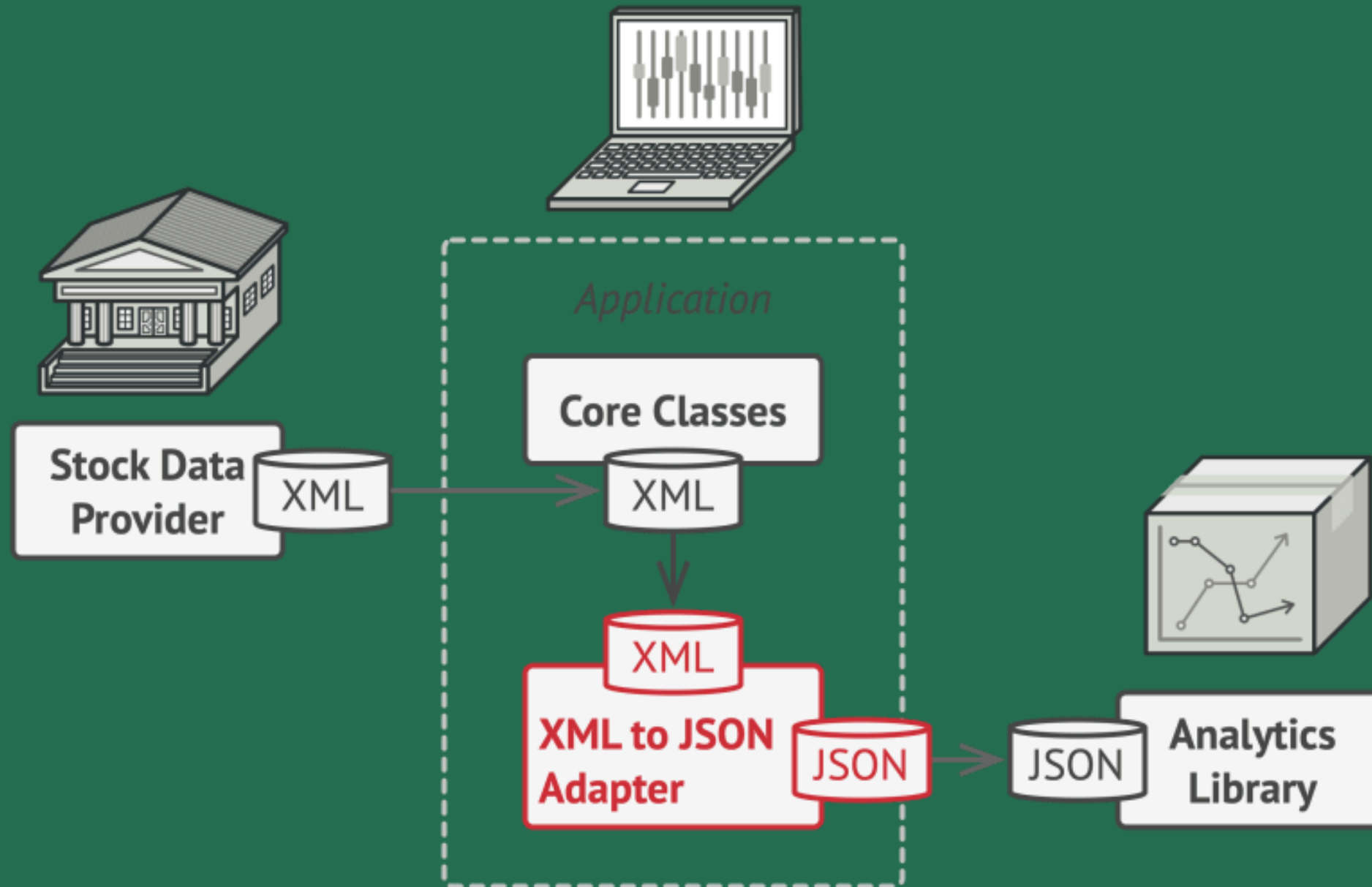
حالا می‌خواهی یه کتابخونه آنالیز جدید اضافه کنی،
ولی اون فقط با JSON کار می‌کنه 🤪

مثال واقعی تر: چه کار می کنی؟؟

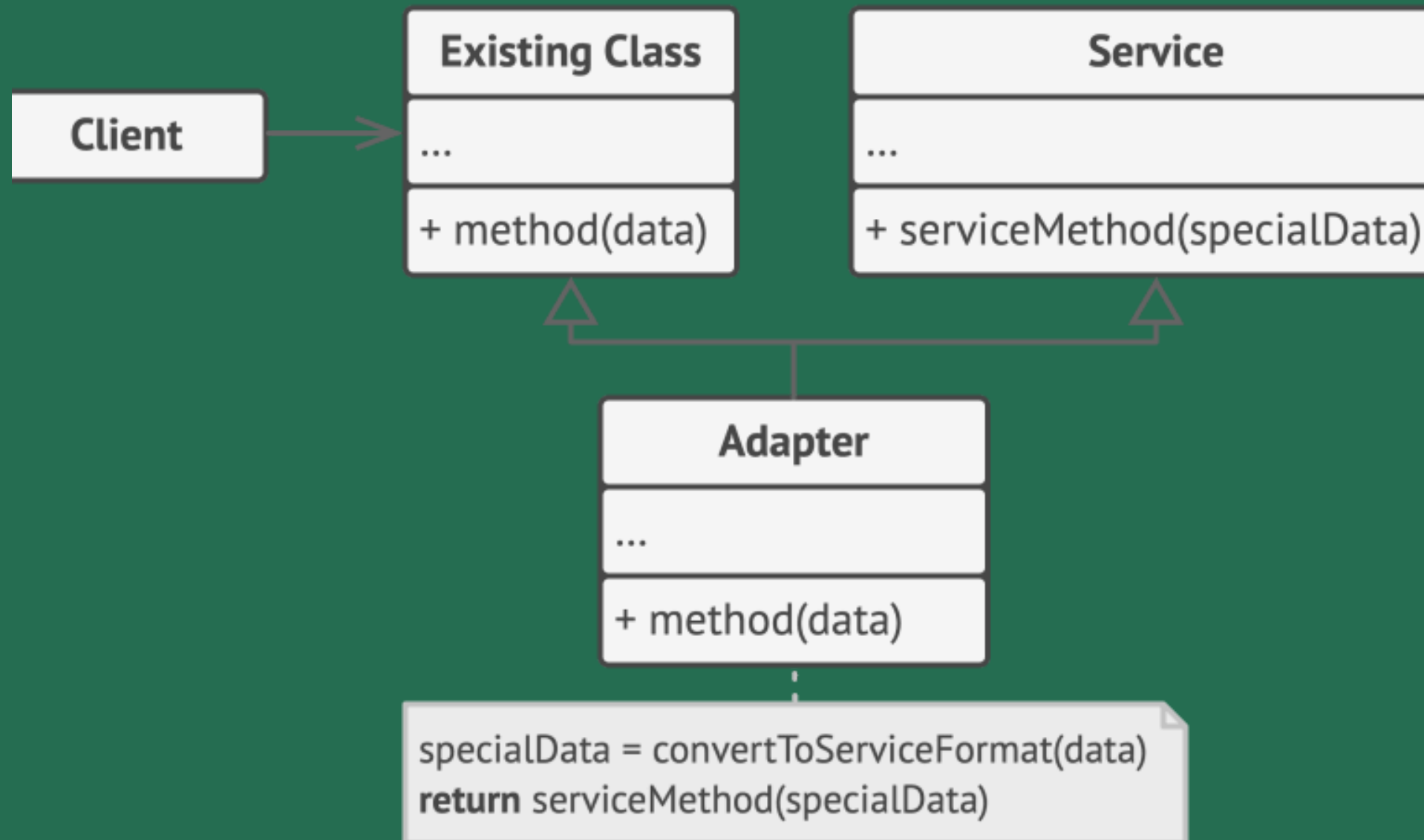
به جای تغییر کتابخونه (که حتی ممکنه کدش دست تو نباشه)،
یه **Adapter** می سازی که **XML** رو به **JSON** تبدیل کنه
و بین اپ و کتابخونه قرار می گیره.

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): 

Adapter از هر دو کلاس زیر ارث‌بری می‌کند:

Target  → رابطی که بقیه‌ی برنامه باهاش کار می‌کنن

Adaptee  → کلاسی که کار اصلی رو انجام می‌ده (ولی رابط متفاوتی داره)

ساختار کلی ADAPTER: چرا از هر دو ارث‌بری می‌کنه؟

چون با این کار:

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

Adapter می‌تونه از Adaptee کمک بگیره تا کار واقعی رو انجام بده

ارتباط بین ADAPTEE و TARGET، ADAPTER



کی باید از ADAPTER استفاده کنیم؟

✓ وقتی به کلاس داری که می‌خواهی استفاده‌اش کنی ولی رابطش با بقیه‌ی برنامه‌ت سازگار نیست.

✓ وقتی چند تا زیرکلاس داری که به قابلیت مشترک ندارند و نمی‌خواهی کد رو تکرار کنی. می‌تونی به Adapter بسازی که اون قابلیت رو بهشون اضافه کنه بدون تغییر در ساختار اصلی.

خلاصه اینکه

آداپتر یعنی "آشتی دهنده‌ی کلاس‌ها"! 😊

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