## *■ Learning new style of code*

لما نيجي نتعلم ستايل جديد في كتابة الكود، الهدف الأساسي بيكون إن الكود يبقى:

* مقروء لأي مبرمج تاني غير اللي كتبه.
* منظم وسهل يتوسع أو يتعدل بعدين.
* نضيف من التكرار والأخطاء اللي بتصعّب الصيانة.

الطرق اللي بتساعد في كده:

* اتباع الـ ***Best Practices*** زي مبدأ ***DRY (Don’t Repeat Yourself) و SOLID.***
* استخدام الـ ***Style Guides*** الخاصة بكل لغة (زي PSR في PHP أو PEP8 في Python).
* الاعتماد على أدوات زي ***Linters & Formatters*** (بتكشف الغلطات وتظبط التنسيق تلقائي).
* إعادة هيكلة الكود ***(Refactoring***): ناخد كود قديم ونظبطه عشان يبقى أنضف.
* قراءة كود لمبرمجين تانيين خصوصًا المشاريع المفتوحة المصدر عشان نتعلم طرق تفكير مختلفة.

الخلاصة: تعلم ستايل جديد مش مجرد تغيير في شكل الكود، هو تغيير في طريقة التفكير وإدارة المشروع بحيث الكود يعيش فترة طويلة من غير ما يبقى صعب الصيانة.

## *■ Learning about OOA & OOD*

***OOA (Object-Oriented Analysis):*** دي المرحلة اللي بنعمل فيها تحليل للمشكلة أو النظام. بنحدد الـ objects اللي موجودة في الواقع (زي طالب، مقرر، محاضر) والـ علاقات اللي بينهم. الهدف هنا نفهم إيه المطلوب من غير ما ندخل في تفاصيل التنفيذ.

***OOD (Object-Oriented Design):*** بعد ما نخلص التحليل ونحدد العناصر، نبدأ نصمم إزاي الكائنات دي هتشتغل مع بعض. يعني بنرسم الـ class diagrams، نحدد الـ methods، ونقسم المسؤوليات بينهم. هنا بنركز على إزاي هننفذ المتطلبات بشكل عملي.

**الفرق بينهم:**

* ***OOA*** = بفكر في الواقع وأحلله.
* ***OOD*** = بفكر في الحل وأصممه.