

## مبادئ الكود النظيف (CLEAN CODE PRINCIPLES)

### • أولاً : المبادئ الأساسية من كتاب (Clean Code) للعلم بوب (روبرت سيسيل مارتن)

الوصف	الاسم بالإنجليزية	الاسم بالعربية
الكود يجب أن يكون واضحًا وسهل الفهم.	Readability	سهولة القراءة
اختيار أسماء تعبر بدقة عن الوظيفة.	Meaningful Names	استخدام أسماء ذات معنى
كتابة دوال صغيرة تؤدي مهمة واحدة فقط.	Small and Clear Functions	دوال قصيرة وواضحة
كل وحدة في الكود يجب أن تؤدي مهمة واحدة فقط.	Single Responsibility Principle	المسؤولية الواحدة
لا تكرر الكود، استخدم الدوال وإعادة الاستخدام.	Don't Repeat Yourself (DRY)	تجنب التكرار
تنظيم الكود بشكل يسهل قراءته ومراجعته.	Proper Formatting	تنسيق الكود
إدارة الاستثناءات بطريقة واضحة ومنظمة.	Effective Error Handling	التعامل مع الأخطاء بفعالية
التأكد من أن الكود يعمل كما هو متوقع من خلال اختبارات.	Unit Testing	كتابة اختبارات

### • ثانيًا: مبادئ تصميم البرمجيات - SOLID

الوصف	الاسم بالإنجليزية	الاسم بالعربية
كل كلاس يجب أن يكون له مسؤولية واحدة.	Single Responsibility Principle (SRP)	مبدأ المسؤولية الواحدة
الكود يجب أن يكون مفتوحًا للإضافة، مغلقًا للتعديل.	Open/Closed Principle (OCP)	مبدأ الفتح/الإغلاق
الكائنات الفرعية يجب أن تستبدل الأصلية بدون مشاكل.	Liskov Substitution Principle (LSP)	مبدأ الاستبدال ليسكوف
تجنب الواجهات الكبيرة غير المتخصصة.	Interface Segregation Principle (ISP)	مبدأ تقسيم الواجهات
الاعتماد على التجريد وليس على التفاصيل.	Dependency Inversion Principle (DIP)	مبدأ عكس الاعتمادية

• ثالثاً: مبادئ برمجية مساندة من منهجيات Agile و XP

الوصف	الاسم بالإنجليزية	الاسم بالعربية
.لا تعقّد الكود بدون داع	Keep It Simple, Stupid (KISS)	اجعلها بسيطة
لا تكتب ميزات قبل الحاجة الحقيقية لها	You Aren't Gonna Need It (YAGNI)	لا تضيف ما لا تحتاجه الآن
كتابة الاختبارات أولاً قبل كتابة الكود	Test-Driven Development (TDD)	التطوير بواسطة الاختبار
تنظيم الكود ليفصل بين المجالات المختلفة ويسهل التوسع	Clean Architecture	الهندسة النظيفة