

تعدد الواجهه : Polymorphism

Describe pattern in oop in which classes have different functionality while sharing common interface

The beauty of Polymorphism is that code working with different classes does not need to which class is using since they are all used in same way

مثال :

كلنا عارفين ازاي نستخدم ال button

بس مش كل ال الازرار بتعمل نفس الحاجه او ليها نفس ال functionality

أنواع: Polymorphism

1. Polymorphism عبر الوراثة: (Inheritance)

- يتم تحقيقه باستخدام الكلاسات الأم (parent class) والكلاسات الفرعية (child classes).
- يمكن للكلاسات الفرعية إعادة تعريف (override) الدوال الموروثة من الكلاس الأم لتقديم سلوك مختلف.

2. Polymorphism عبر الواجهات: (Interfaces)

- يتم تحقيقه باستخدام الواجهات (Interfaces) التي تُلزم الكلاسات بتنفيذ دوال معينة.
- يسمح لفئات مختلفة بتقديم تنفيذات مختلفة لنفس الواجهة.

2. Polymorphism عبر الكلاسات المجردة: (Abstract Classes)

- يشبه الواجهات، ولكنه يسمح بتعريف دوال مجردة (abstract methods) ودوال عادية.
- الكلاسات الفرعية ملزمة بتقديم تنفيذ للدوال المجردة.

- المرونة: يسمح بكتابة كود أكثر مرونة وقابلية للتوسع.
- إعادة الاستخدام: يمكن استخدام نفس الواجهة أو الدالة مع كائنات مختلفة.
- سهولة الصيانة: يجعل الكود أكثر تنظيمًا وسهولة في التعديل.