Programação Orientada a Objetos (POO)

1. **Representação de algo do mundo real. Exemplo: alunos.**
2. **Classe: Modelo para criar vários objetos.**
3. **Encapsulamento: Protege dados.**
4. **Herança: Reutiliza códigos de outra classe.**
5. **Objeto Polimorfismo: Mesmo método com comportamento diferente.**
6. **Abstração: Cria um modelo base**

**Pilares: Abstração, encapsulamento, herança, polimorfismo:**

**Abstração: Permite criar classes que não podem ser instanciadas diretamente, servindo apenas como *modelo base* por outras classes. Evitando criação de objetos genéticos.**

**Encapsulamento: ele protege informações privadas evitando mudanças acidentais. Exemplo: um aluno tem uma nota secreta**

**Herança (reutilizar): Herança permite que uma classe aproveite características de outra, evitando repetição Exemplo: cria a classe pessoa que contém nome e idade, depois criar aluno e professor que herdam as características de pessoa.**

**Polimorfismo: O polimorfismo permite que diferentes classes exibam o mesmo método, mas funcionando de formas diferentes**

**Explicação:**

* **A classe Pessoa não pode ser instanciada diretamente.**
* **O If no Contructor impede que objetos iguais sejam criados.**
* **O Aluno herda de pessoa, então pode ser instanciado normalmente.**
* **Isso garante que Pessoa funcione como modelo para outras classes.**