**Orientação a objeto**

**Abstração:** pensa em uma empresa, abstração é como se fosse delegar uma função e ela pode ser construída por você (você ensinou um funcionário a executar essa função) mas você também pode contratar isso de uma forma externa, então na programação você faz isso para que pequenos blocos são responsáveis por algumas tarefas e que possamos reutilizar isso.

**Encapsulamento:** forma segura de trabalhar, formas privadas de fazer algo

**Herança:** resolver mais problemas, se torna produtivo, se você passar pelos 2 primeiros passos, a herança começa fazer sentido, você para de repetir.

**Polimorfismo:** completa todo o sistema. Eu tenho uma regra de negócio de um funcionário e em um determinado momento essa regra muda (em um momento você verifica só o e-mail depois precisa verificar o CPF)

**Classes:** design da rotina, se você tem um usuário ele tem certas ações e todas elas são o design da rotina e o objeto utiliza esse design para executar essas ações. Assim nos temos 2 camadas, o modelo da aplicação (classes) e a segunda das os objetos que interagem com o sistema e nos dão acesso aos recursos dessas classes.

Conjunto de propriedades que refletem em um modelo ao mesmo tempo que tem um conjunto de rotinas que são os métodos que são funções que se responsabilizam por tratar todas essas propriedades

Nome de uma classe:

- Sempre tem o mesmo nome do arquivo

- Sempre utiliza schudlecase (User class -> User.php)

- Classe deve estar sozinha no próprio arquivo, somente a classe

Functional descreve a função da pessoa

Organograma departamental coloca departamento

Suporte 7/4

Qual o horariow comercercial