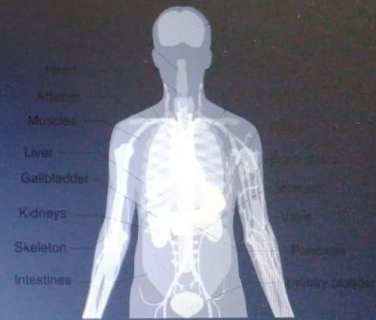


Abstração

Permite isolar de um objeto **somente os conceitos que são necessários** para o funcionamento do programa.

```
export class Pessoa {  
  private nome: string;  
  private sobrenome: string;  
  
  constructor(nome: string, sobrenome: string) {  
    this.nome = nome;  
    this.sobrenome = sobrenome;  
  }  
}
```



Encapsulamento

Visa **ocultar partes internas de um objeto** e exibir apenas o necessário para uso externo.

```
export class RemoteControl {  
  constructor(private powerStatus = false) {}  
  
  public turnOn(): void {  
    this.powerStatus = true;  
  }  
  
  public turnOff(): void {  
    this.powerStatus = false;  
  }  
  
  public getStatus(): boolean {  
    return this.powerStatus;  
  }  
}
```



Herança

Visa **passar características de um objeto para outro.**

```
export abstract class Animal {
  constructor(public name: string) {}
  abstract makeNoise(): void;
}
export class Dog extends Animal {
  makeNoise(): void {
    console.log(`${this.name} está fazendo AU AU...`);
  }
}
export class Cat extends Animal {
  makeNoise(): void {
    console.log(`${this.name} está fazendo MIAU...`);
  }
}
```



Polimorfismo

Algo que é polimorfo tem a **habilidade de assumir diferentes formas.**

```
class AnimalSounds {
  public playSound(animal: Animal): void {
    animal.makeNoise();
  }
}

const dog = new Dog('Tina');
const cat = new Cat('Suzy');
const animalSounds = new AnimalSounds();
animalSounds.playSound(dog); // Tina está fazendo AU AU...
animalSounds.playSound(cat); // Suzy está fazendo MIAU...
```

