

# Principais Relações entre Classes

## 1. Associação

- Quando uma classe **usa** ou **se relaciona** com a outra.
- Exemplo: quando um **Professor** leciona uma **Disciplina**.

```
class Professor:
    def __init__(self, nome):
        self.nome = nome

class Disciplina:
    def __init__(self, nome, professor: Professor):
        self.nome = nome
        self.professor = professor # Referência -> Associação
```

## 2. Agregação (uma forma especial de associação)

- Quando uma classe **possui** outra, mas **podem existir separadamente**. Portanto, se o objeto principal for destruído, o outro continua existindo.
- Exemplo: os **Alunos** fazem parte da **Turma**, mas continuam existindo fora dela.

```
class Aluno:
    def __init__(self, nome):
        self.nome = nome

class Turma:
    def __init__(self):
        self.alunos = [] # agregação

    def adicionar_aluno(self, aluno):
        self.alunos.append(aluno)
```

### 3. Composição

- Quando uma classe **compõe** outra. Logo, se o objeto principal for destruído, os objetos internos também são.
- Exemplo: Um **Carro** tem um **Motor**. Se o carro deixar de existir, o motor também.

```
class Motor:
    def __init__(self, tipo):
        self.tipo = tipo

class Carro:
    def __init__(self):
        self.motor = Motor("V8") # composição
```

### 4. Herança

- Quando uma classe **herda** atributos e métodos de outra. É uma relação "é um". Usamos para reutilizar código.
- Exemplo: Um **Aluno** é uma **Pessoa**.

```
class Pessoa:
    def __init__(self, nome):
        self.nome = nome

class Estudante(Pessoa): # herança
    def __init__(self, nome, curso):
        super().__init__(nome)
        self.curso = curso
```