



# Composição

- Composição é uma forma de combinar objetos simples com o objetivo de formar objetos mais complexos
  - Implica em uma relação de **ter** (ou é composto), ao invés de uma relação de **ser** (obtida via herança)



# Composição

- Novos objetos podem ser criados **combinando** objetos já existentes (princípio da reutilização)
- Um objeto (A) pode ter uma **relação** com outro objeto (B) de diversas formas (multiplicidade)
  - **0..1**: A tem **no máximo um** B
  - **1..1**: A tem **um e somente um** B
  - **0..\***: A **tem muitos** B's
  - **1..\***: A **tem um ou mais** B's
  - **3..5**: A tem **de três a cinco** B's (valores específicos)

# Composição

- A multiplicidade é implementada usando objetos como atributos
- Um objeto pode ter um objeto como atributo
  - Ex.: Um relógio possui data e hora
- Um objeto pode ter uma coleção de objetos como atributo
  - Ex.: Uma turma possui vários alunos (implementado como *arrays*, coleções ou outros)

# Exemplo em Java

- Exemplo de coleções de objetos como atributos

```
class Turma {  
    Aluno[] alunos;  
  
    Turma(int n) {  
        alunos = new Aluno [n];  
    }  
  
    void matricular(Aluno a) {  
        for (int i = 0; i < 4; i++)  
            if (alunos[i] != null) {  
                alunos[i] = a;  
                break;  
            }  
    }  
}
```

Ou

```
import java.util.*;  
class Turma {  
    List<Aluno> alunos = new  
        ArrayList<Aluno>();  
  
    void matricular(Aluno a) {  
        alunos.add(a);  
    }  
}
```

# Herança vs. Composição

- **Herança e composição** são dois mecanismos para reutilização de funcionalidades
  - **Herança**: estende atributos e métodos de uma classe
  - **Composição**: estende através de delegação
- **Herança** sempre foi considerada como a ferramenta básica para extensão e reuso, mas **composição** é muito superior na maioria dos casos
  - Herança define um relacionamento estático, enquanto composição um relacionamento dinâmico
- **Composição e herança** não são mutuamente exclusivas; as técnicas podem ser combinadas para obter melhores resultados

# Herança vs. Composição

- Quando usar **Herança**?
  - Somente deve ser usada quando estiver construindo uma família de tipos (relacionados entre si)
  - Somente se puder comparar um objeto A com outro objeto B dizendo que, A "**É UM tipo de...**" B
- Quando usar **Composição**?
  - Em todos os outros casos!

**Dica:** sempre favoreça  
composição sobre herança