Programação OO

5ª aula

Prof. Douglas Oliveira
douglas.oliveira@prof.infnet.edu.br

Principais Características de OO

- Classificação ✔
- Abstração ✔
- Encapsulamento ✔
- Relacionamentos
- Herança
- Polimorfismo

Relacionamentos

- Relacionamentos

Mecanismo pelo qual um objeto se relaciona com outro com o objetivo de utilizar recursos providos por este.

Relacionamentos

- Existem vários tipos de relacionamentos entre classes:
  - ✔ Associação
  - ✔ Composição
  - ✔ Agregação
  - ✔ Hierarquia

Relacionamentos

- Exemplos:
  - ✔ Professor leciona disciplinas
  - ✔ Aluno pertence a curso
  - ✔ Aluno cursa disciplinas
  - ✔ Transportadora entrega produtos
  - ✔ Ônibus transporta passageiros
  - ✔ Carro possui motor
  - ✔ Jogador faz parte de time
  - ✔ Livro possui capítulos

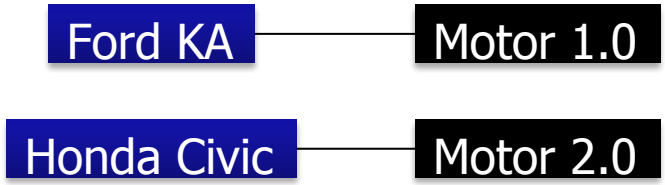
Relacionamentos

- Quando identificamos um relacionamento é preciso também identificar a cardinalidade ou multiplicidade desse relacionamento.

- Existem 3 tipos de cardinalidade:

1) 1 para 1: um objeto de uma classe está relacionado a um objeto de outra classe.

   - Exemplo:
     - ✔ Um carro possui um motor

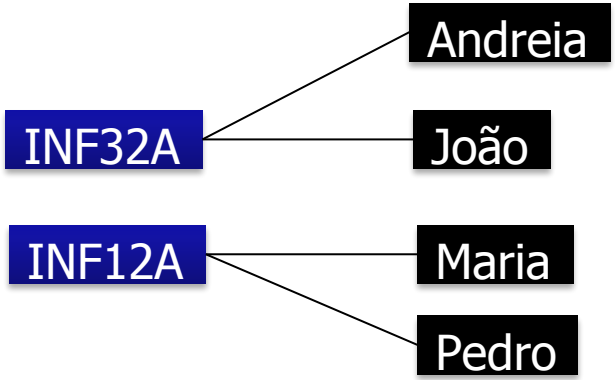


Relacionamentos

- Existem 3 tipos de cardinalidade:

2) 1 para muitos: um objeto de uma classe está relacionado a vários objetos de outra classe.

- Exemplos:
  - ✔ Um time possui vários jogadores
  - ✔ Uma turma possui vários alunos



Relacionamentos

- Existem 3 tipos de cardinalidade:

3) muitos para muitos: vários objetos de uma classe estão relacionados a vários objetos de outra classe.

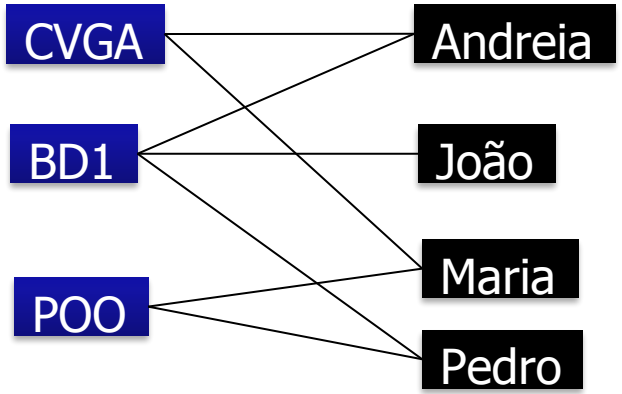
- Exemplo:
  - ✔ Um aluno está inscrito em várias turmas;
  - ✔ Uma turma possui vários alunos.

Relacionamentos

- Existem 3 tipos de cardinalidade:

3) muitos para muitos: vários objetos de uma classe estão relacionados a vários objetos de outra classe.

- Exemplo:



Relacionamentos

- Como implementar essas cardinalidades?

1) 1 para 1: declaramos na classe origem um atributo do mesmo tipo da classe destino.
   - Exemplos:
     - ✔ Um carro possui um motor
     - ✔ Um aluno pertence a um curso

```
public class Carro {
  private String modelo;
  private int ano;
  private Motor motor;
}
```

```
public class Aluno {
  private int matricula;
  private String nome;
  private Curso curso;
}
```

Exercícios

- Exercício 28: crie as classes Carro e Motor, onde o carro possui modelo (string) e um motor e o motor possui uma cilindrada (float). Em seguida implemente na classe Carro o método:

```
float velocidadeMaxima()
```

que calcula a velocidade máxima do carro baseada na cilindrada do motor:

- Até 1.0: 140Km/k
- Acima de 1.0 até 1.6: 180Km/h
- Acima de 1.6 até 2.0: 220Km/h
- Acima de 2.0: 260Km/h

Por fim, crie instâncias da classe Carro com motores de cilindradas iguais e diferentes.

Exercícios

- Exercício 29: Crie a classe Ponto, que representa um ponto (x, y) no plano cartesiano. Em seguida, implemente a classe SegmentoReta, que é formada por dois pontos. Na classe SegmentoReta defina os métodos:

```
public SegmentoReta(Ponto p1, Ponto p2)
public SegmentoReta(float x1, float y1,
                    float x2, float y2)
public float tamanho()
```

Em seguida, leia dados para criar dois pontos e, usando esses dois pontos, crie um segmento de reta. Imprima o tamanho desse segmento.

Dica: repare que o tamanho de um segmento de reta pode ser definido coma distância entre os pontos que o formam.