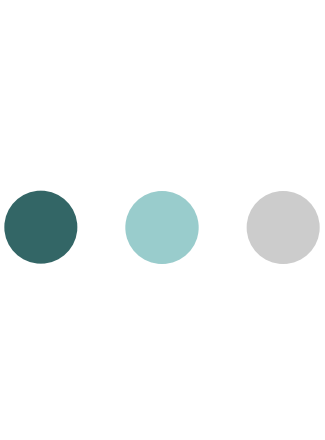
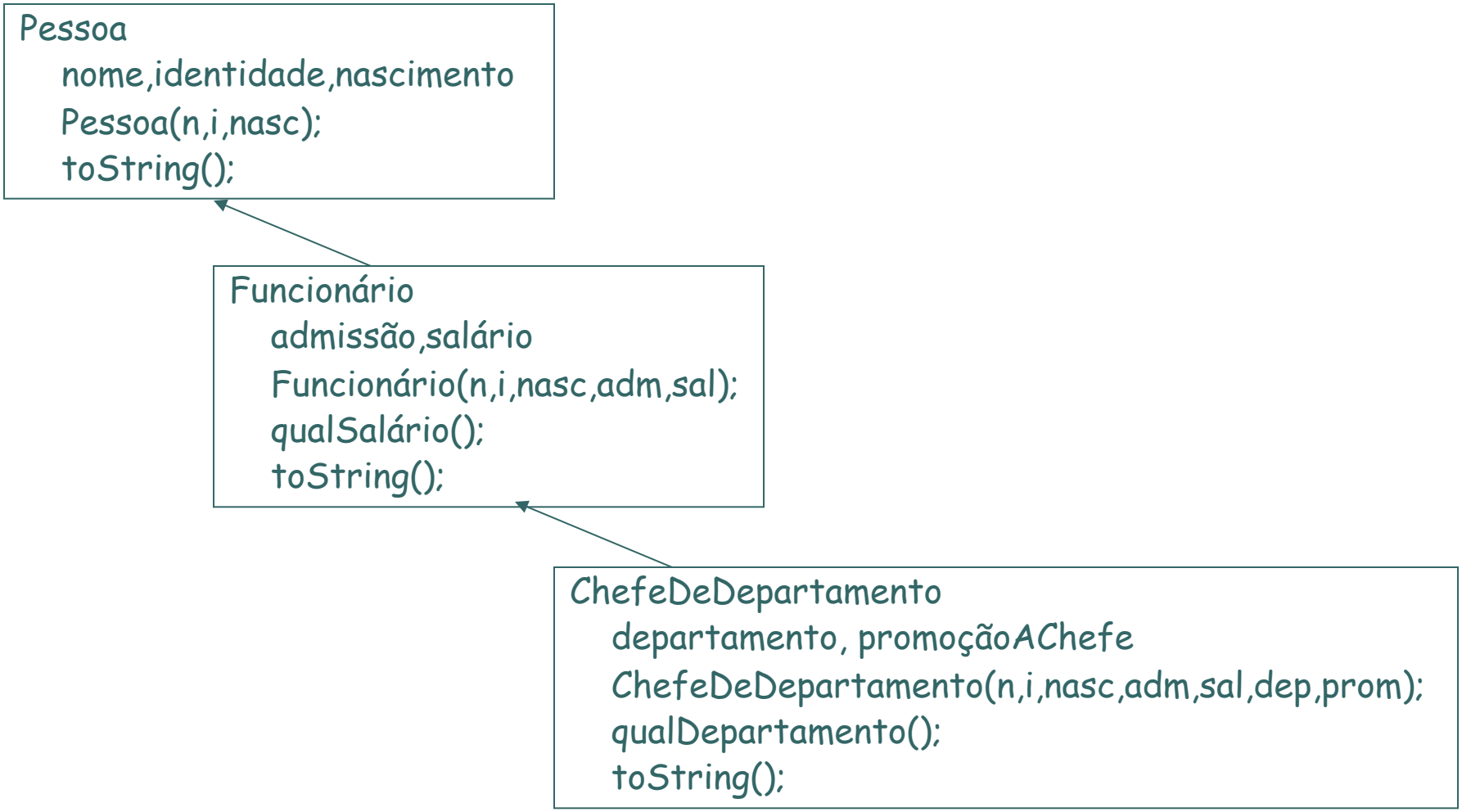


Polimorfismo

MATA 55

Rita Suzana

Herança


Pessoa
nome,identidade,nascimento
Pessoa(n,i,nasc);
toString();
Funcionário
admissão,salário
Funcionário(n,i,nasc,adm,sal);
qualSalário();
toString();
ChefeDeDepartamento
departamento, promoçãoAChefe
ChefeDeDepartamento(n,i,nasc,adm,sal,dep,prom);
qualDepartamento();
toString();

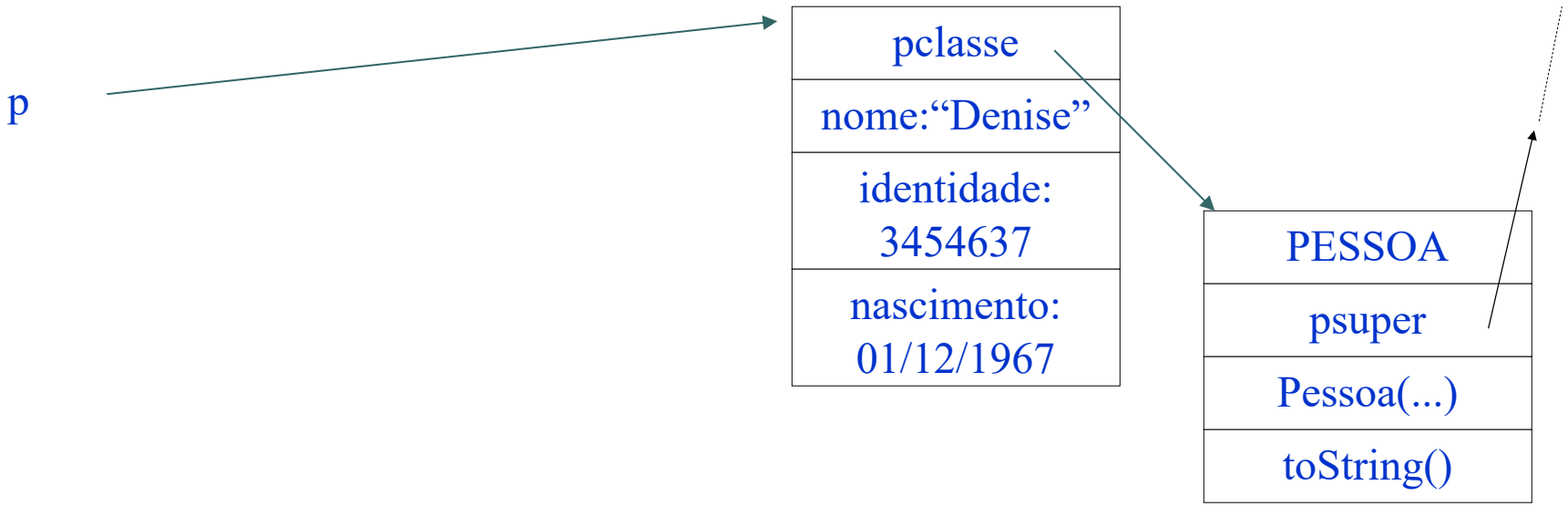
Amarração Tardia de Tipos

- Exemplo

```
public class Empresa {
        public static void main(String[] args) {
            float s;  int i;
            DataCons d1 = new DataCons((byte)12,(byte)12,(short)1967);
            Pessoa p = new Pessoa ("Denise", 3454637, d1);
            DataCons d2 = new DataCons((byte)1,(byte)12,(short)1972);
            DataCons d3 = new DataCons((byte)1,(byte)12,(short)2002);
            i = p.qualIdentidade();
            Funcionario f1 =
                  new Funcionario ("Rogerio", 93452128 ,d2 ,d3 ,(float)1000.00);
            s = f1.qualSalario();
            i = f1.qualIdentidade();
           System.out.println(f1);
        }
    }
```

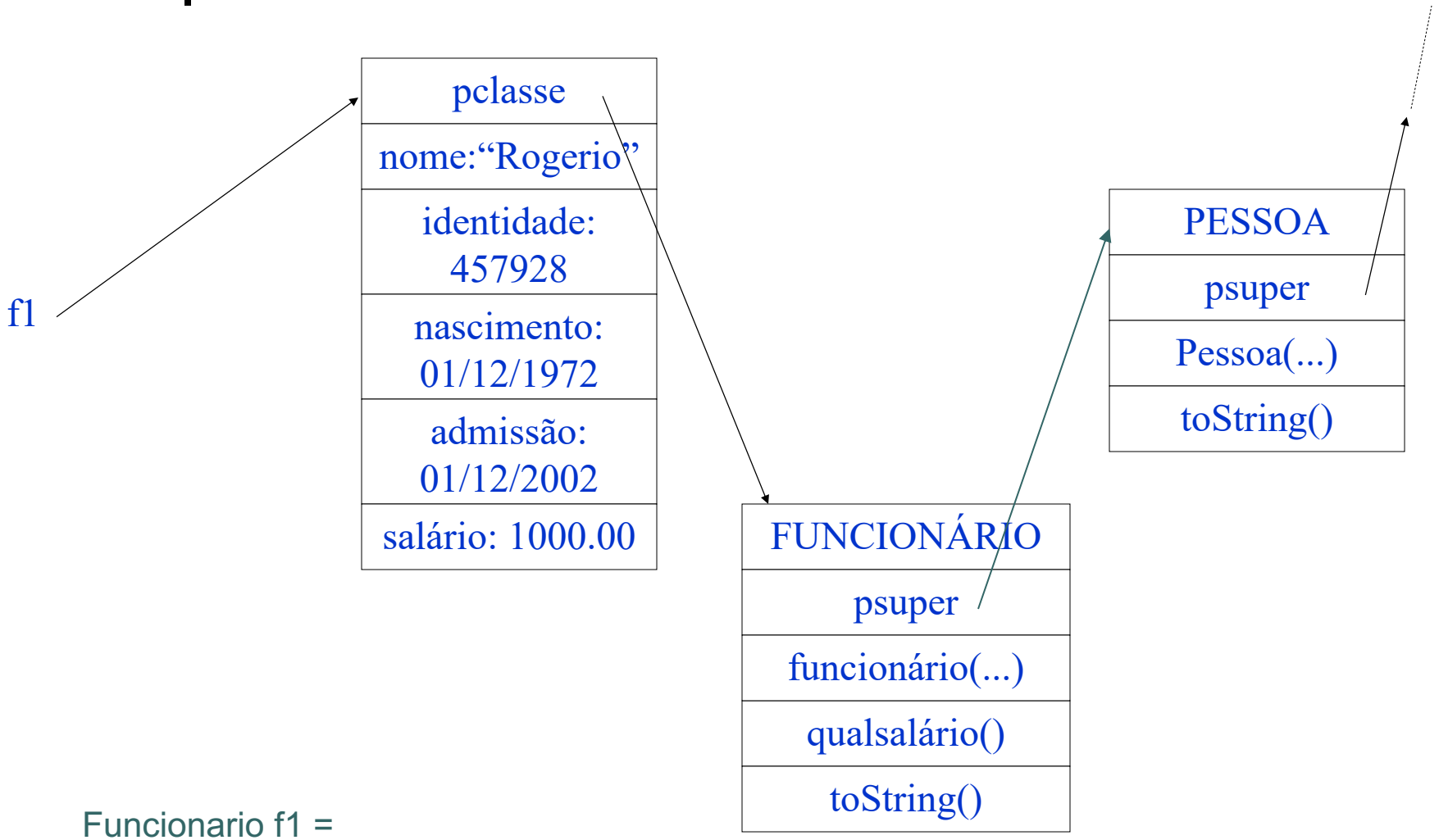
Empresa.java

Amarração Tardia de Tipos




```
Pessoa p = new Pessoa ("Denise", 3454637, d1);
```

Amarração Tardia de Tipos




```
Funcionario f1 =
        new Funcionario ("Rogerio", 93452128 ,d2 ,d3 ,(float)1000.00);
```

Amarração Tardia de Tipos

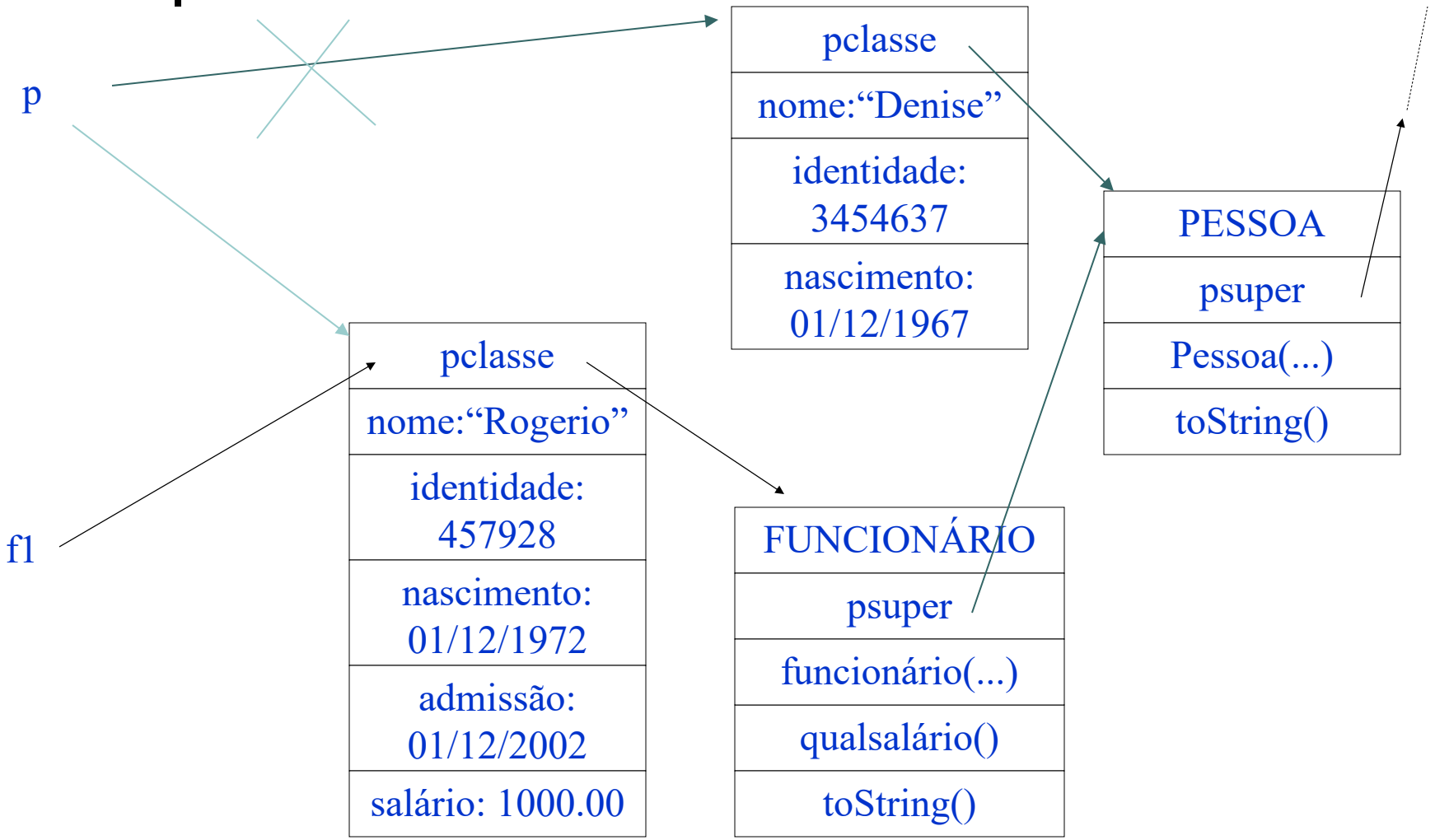
```
public class EmpresaDin {
      public static void main(String[] args) {
          float s;  int i;
          DataCons d1 = new DataCons((byte)12,(byte)12,(short)1967);
          Pessoa p = new Pessoa ("Denise", 3454637, d1);
          DataCons d2 = new DataCons((byte)1,(byte)12,(short)1972);
          DataCons d3 = new DataCons((byte)1,(byte)12,(short)2002);
          Funcionario f1 =
               new Funcionario ("Rogerio", 457928,d2,d3,(float)1000.00)

          p = f1;
          System.out.println(p);
    }
}
```

referência de Pessoa passa a apontar para instância de Funcionário

invoca toString de Pessoa? de Funcionário?

Amarração Tardia de Tipos



Amarração Tardia de Tipos

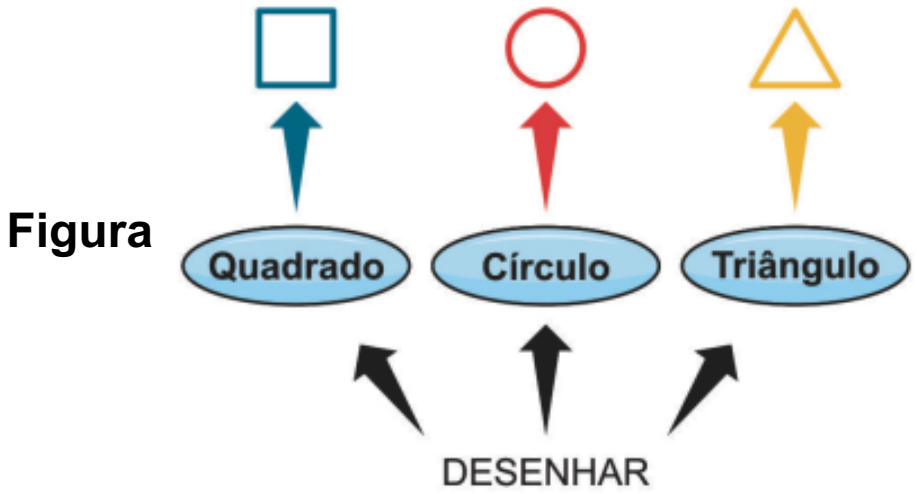
- O Java chama isso de Ligação Dinâmica ou Dynamic Binding
- A maquina virtual sabe qual o tipo atual do objeto. Assim faz uma ligaçao dinâmica e quando o objeto vem de uma determinada classe, os métodos também também vem da classe.
- A maquina virtual cria uma tabela de métodos, onde para cada classe ha as assinaturas correspondentes.
- Quando executamos um exemplo como os já mostrados a JVM simplesmente acessa esta tabela.

Polimorfismo

- A relação ***é-um-tipo-de*** entre classes permite a existência de outra característica fundamental de linguagens OO, o **polimorfismo**
- "Muitas formas"
  - Permite a manipulação de instâncias de classes que herdem de uma mesma classe ancestral de forma unificada.
- Um mesmo tipo de objeto, sob certas condições, pode realizar ações diferentes ao receber uma mesma mensagem

Polimorfismo

- Permite o envio de uma mesma mensagem a objetos distintos
  - cada objeto responde da maneira mais apropriada
    - Java -> método sobrescrito



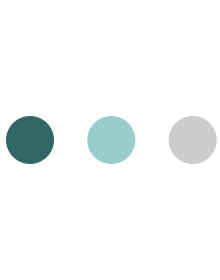
Polimorfismo

- Algumas definições
- Mecanismo que permite que uma operação receba argumentos de diferentes tipos

  Um parâmetro declarado como valor de um tipo...
  ... pode receber valores do subtipo.

- Mecanismo que permite que um método receba argumentos de diferentes classes:

  Um parâmetro declarado como instância da superclasse...
  ... pode receber instâncias da subclasse.



Polimorfismo

- Exercício Automóvel, BlueJ

Exemplo

```
public class ConcessionariaDeAutomoveis
{ public static void main(String args[])
  {  Automovel a1 = new Automovel("Fiat","bege",
                                        Automovel.movidoAAlcool);
     AutomovelBasico a2 = new AutomovelBasico("Corsa","cinza",
                                              Automovel.movidoAGasolina);
     imprime(a1);
     imprime(a2);
  }

  public static void imprime(Automovel a)
  { System.out.println(“Dados do automóvel escolhido : ");
    System.out.println(a.toString());
    System.out.println("Valor : "+a.quantoCusta());
    System.out.println(a.quantasPrestacoes()+" prestações de "+
                       (a.quantoCusta()/a.quantasPrestacoes()));
  }
}
```

O método **imprime** recebe um objeto da classe Automovel como parâmetro. Observe que nesse exemplo, chamamos o método passando a1 e a2, ou seja, objetos de classes diferentes mas da mesma hierarquia de classes, caracterizando dessa forma a utilização de polimorfismo.

Polimorfismo

- **Vantagens do uso do Polimorfismo:**
  - Permite o envio de mensagens a um objeto sem a determinação exata do seu tipo;
  - Permite a extensibilidade do sistema com pouco ou nenhum impacto nas classes existentes;
  - Permite a construção de classes genéricas para efetuar tarefas gerais comuns aos softwares, como as estruturas de dados.