ODMG - *Object Database Management Group*

- Padrão para SGBDOO
- Consórcio de pesquisadores e fabricantes
- Componentes principais do padrão
  - modelo de objetos
  - linguagem de definição de dados (ODL)
  - linguagem de consulta (OQL)

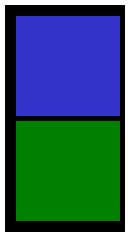
Modelo de Objetos

- Características
  - Formas de definição de dados (ODL)
    - Interfaces
      - define apenas comportamento (assinatura)
      - não possui instâncias
    - Classes
      - define atributos, relacionamentos e comportamento
      - possui instâncias (com OID – chamados objetos)
      - implementação do comportamento em alguma LPOO
  - Herança
  - OID e chave
- Exemplo de definição de interface em ODL

```
interface Pessoa {
   short idade();
   boolean ehMenor();
   void casou(Pessoa p) raises(jahEhCasado);
   ...                  };
```

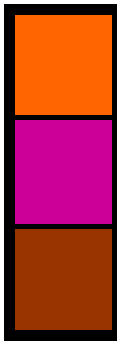
Definição de Classe em ODL

```
class Departamento (extent Departamentos) {
  attribute string nome;
  attribute short código;
  attribute struct Endereço{
     string rua, short número,
     string cidade} localização;
  attribute struct atendimento{
   time horaInício, time horaTérmino} horário;
  attribute Empregado chefe;
  relationship set<Empregado> empregados
     inverse Empregado:: depto;
  void adicionaEmp(short RG) raises
  (jahTrabalha, RGInexistente);
  ...
  };
```



conjunto de instâncias

atributo atômico



atributo estruturado

atributo de referência a objeto

relacionamento

Herança

- Duas formas
  - herança *IS-A*
  - herança *extends*
- Herança *IS-A*
  - herança de interface
  - pode ser *interface*←*interface* ou *interface*←*classe*
  - permite herança múltipla
- Herança *extends*
  - herança de propriedades
  - ocorre somente entre classes (*classe*←*classe*)
  - não permite herança múltipla

Herança em ODL

```
Classe Empregado (extent Empregados) {
   attribute short RG;
   attribute string nome;
   attribute enum gênero{M,F} sexo;
   attribute Date DN;
   attribute float salário;
   relationship Departamento depto
       inverse Departamento:: empregados;
   ...
};

Classe Professor (extent Professores)
   extends Empregado : Pessoa {
   attribute string titulação;
   attribute string areaAtuação;
   ...
   };
```

herança extends

herança IS-A

OID e Chave

- OID
  - identificador do objeto
- Chave
  - uma ou mais propriedades cujos valores devem ser únicos

```
class Departamentos
(extent todosDeptos
 key código)
{
attribute string nome;
attribute short código;
...
};
```

```
class Cidades
(key (estado,nome))
{
 attribute string estado;
 attribute string nome;
 ...
}
```

Consultas em BDOOs

- Duas abordagens
  - navegacional
    - navegação entre objetos através de suas referências
      - controlada pelo programa da aplicação ou SGBDOO
  - declarativa
    - uso de uma linguagem de consulta do SGBDOO
    - violação de encapsulamento
      - maior flexibilidade para formulação de consultas
- Não há DML, apenas linguagem de consulta
  - métodos implementam operações de atualização
- Linguagem de consulta do padrão ODMG
  - OQL (*Object Query Language*)

OQL

- Linguagem de consulta declarativa
- Dialeto SQL com suporte ao tratamento de
  - objetos complexos
  - junções por valor ou por OID
  - invocação de métodos
  - buscas em hierarquias de herança

Consultas e Resultados

- Ponto de partida de uma consulta
  - extensão de uma classe (*extent*)

```
select e.*
from e in Empregados
```

variável de iteração

- Resultados de consultas
  - atributos de objetos e/ou novas estruturas

```
select struct (
 nome:         d.nome
 empsRicos:    (select e.*
                from e in d.empregados
                where e.salário > 5000))
from d in Departamentos
```

Expressões de Caminho

- Permitem a navegação em estruturas complexas e objetos associados
  - objetos associados
    - atributos de referência e relacionamentos
  - utiliza-se a notação de ponto (“.”)
- Exemplo

```
select p.nome, p.titulação
from p in Professores
where p.depto.código = 'INE'
```

Expressões de Caminho

- Variáveis de iteração são definidas para a navegação em coleções de objetos referenciados (referências 1:N)
  - a variável de iteração associa-se com cada elemento da coleção referenciada
- Exemplo

```
select e.nome
from d in Departamentos, e in d.empregados
where d.código = 'INE'
and e.salário > 5000
```

Junções

- Junções entre conjuntos de objetos são permitidas, como em BDRs
- Junções tanto por valor quanto por OID são permitidas
- Exemplo

```
select e2.nome
from e1 in Empregados, e2 in Empregados
where e1.nome = 'Pedro Campos Souza'
and e2.salário = e1.salário
and e2.depto = e1.depto
```

junção por valor

junção por OID

Invocação de Métodos

- Métodos podem ser declarados em consultas da mesma forma que atributos
- Exemplos

```
select e.nome
from e in Empregados
where e.idade > 50

select d.código, d.nroHorasAtendimento
from d in Departamentos
```

Consultas em Hierarquias de Classes

- Consultas aplicadas a uma classe processam automaticamente objetos da classe e de suas subclasses
- Restrições sobre subclasses alvo podem ser especificadas
- Exemplo

```
select (Professores, Pesquisadores)e.nome
from e in Empregados
where e.salário > 3000
```

Funções de Agregação

- Aplicadas sobre qualquer coleção de dados
- Exemplos

```
avg(select p.salário
from p in Professores
where p.depto.código = 'INE')

select d.código, d.nome
from d in Departamentos
where count (d.empregados) > 30
```