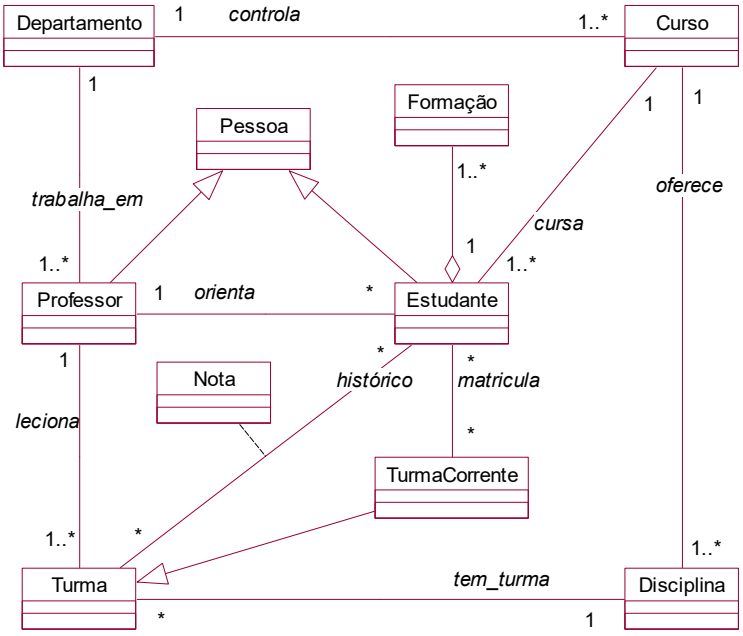
UFOP / ICEB / DECOM
Banco de Dados II (BCC441) – Prof. Guilherme Tavares de Assis

Lista de Exercícios VII – BDOO

Considere o seguinte diagrama de classes para um sistema de controle acadêmico, conforme apresentado em sala de aula:



Considere a seguinte definição de classes em ODL (*Object Definition Language*) referente ao diagrama de classes apresentado:

```
class Pessoa
(  extent pessoas  key identidade  )
{
   attribute  string  nome;
   attribute  string  identidade;
   attribute  date    dataNascimento;
   attribute  enum  Gênero {M,F}  sexo;
   attribute  struct  Endereço  {short número, string rua, string compl, string cidade,
                                 string estado, string cep}  endereço;
   short  idade();
};

class Professor extends Pessoa
(  extent professores  )
{
   attribute  string  cargo;
   attribute  float    salário;
   attribute  string   sala;
   attribute  string   fone;
   relationship  Departamento  trabalha_em  inverse  Departamento::tem_professores;
   relationship  set<Estudante>  orienta  inverse  Estudante::orientador;
   relationship  set<Turma>  leciona  inverse  Turma::professor;
   void  reajustaSalario (in float percentual);
   void  promove (in string novo_cargo);
};
```

```
class Estudante extends Pessoa
( extent estudantes )
{
  attribute string matrícula;
  relationship set<Formação> formação;
  relationship Curso cursa inverse Curso::tem_estudantes;
  relationship set<Nota> turmas_completadas inverse Nota::estudante;
  relationship set<TurmaCorrente> matriculado_em inverse
                                    TurmaCorrente::estudantes_matriculados;
  relationship Professor orientador inverse Professor::orienta;
  float mediaGeral();
  void matricula (in Turma t);
  void atribuiNota (in Turma t, in double nota) raises (nota_inválida);
  void atribuiOrientador (in string nome) raises (nome_inválido);
};

class Formação
{
  attribute string escola;
  attribute string grau;
  attribute short ano;
};

class Disciplina
( extent disciplinas key codDisc )
{
  attribute string nomeDisc;
  attribute string codDisc;
  attribute string ementa;
  attribute short numCreditos;
  attribute short cargaHorária;
  relationship set<Turma> tem_turmas inverse Turma::disciplina;
  relationship Curso oferecida_por inverse Curso::oferece;
};

class Turma
( extent turmas )
{
  attribute string codTurma;
  attribute short ano;
  attribute short semestre;
  relationship Disciplina disciplina inverse Disciplina::tem_turmas;
  relationship set<Nota> estudantes inverse Nota::turma;
  relationship Professor professor inverse Professor::leciona;
};

class TurmaCorrente extends Turma
( extent turmas_corrente )
{
  relationship set<Estudante> estudantes_matriculados inverse Estudante::matriculado_em;
  void matricula (in string matrícula) raises (matrícula_inválida, turma_cheia);
};

class Nota
( extent notas )
{
  attribute double nota;
  relationship Turma turma inverse Turma::estudantes;
  relationship Estudante estudante inverse Estudante::turmas_completadas;
};
```

UFOP / ICEB / DECOM
Banco de Dados II (BCC441) – Prof. Guilherme Tavares de Assis

class Departamento
```
( extent departamentos key codDepto )
{
  attribute string codDepto;
  attribute string nomeDepto;
  relationship set<Professor> tem_professores inverse Professor::trabalha_em;
  relationship set<Curso> controla inverse Curso::pertence_a;
};
```

class Curso
```
( extent cursos key nomeCurso )
{
  attribute string nomeCurso;
  relationship Departamento pertence_a inverse Departamento::controla;
  relationship set<Estudade> tem_estudantes inverse Estudante::cursa;
  relationship set<Disciplina> oferece inverse Disciplina::oferecida_por;
};
```

⇨ Faça cinco consultas em OQL (*Object Query Language*) dentre as seguintes:

1. Recuperar o nome de todas as pessoas.
2. Recuperar o nome dos professores que recebem salário superior a R$5000,00.
3. Recuperar o nome dos estudantes do curso de 'Ciência da Computação'.
4. Recuperar o nome dos estudantes do curso de 'Ciência da Computação' que são orientandos do professor 'José da Silva'.
5. Criar uma visão que recupere uma disciplina cujo nome é fornecido como parâmetro.
6. Utilizando a visão do item "5", recuperar o curso que oferece a disciplina 'Banco de Dados II'.
7. Utilizando a visão do item "5", recuperar o ano, o semestre e o nome do professor de toda turma já oferecida da disciplina 'Banco de Dados II'.
8. Recuperar o nome, a formação e a média geral de todos os alunos do curso de 'Ciência da Computação'.
9. Recuperar o histórico da aluna 'Ana de Souza', ou seja, nome da disciplina, número de créditos, ano, semestre e nota obtida de todas as disciplinas já cursadas pela aluna, ordenado pelo ano e semestre das disciplinas.
10. Recuperar o maior salário dentre os salários dos professores do curso de 'Ciência da Computação'.
11. Recuperar o código das turmas correntes, juntamente com os nomes das disciplinas correspondentes, que possuem mais de 50 estudantes matriculados.
12. Responda: existe algum aluno do curso de 'Ciência da Computação' que nunca foi reprovado em alguma disciplina (nota ≥ 6)?
13. Responda: já foram concluídas turmas para todas as disciplinas do curso de 'Ciência da Computação'?
14. Recuperar o nome do último aluno orientando do professor 'José da Silva', em ordem alfabética do seu nome, que possui média geral superior a 8.
15. Para cada professor que possui mais de 3 orientandos, recuperar seu nome, o nome do seu departamento e a média das médias gerais dos seus orientandos.

1) Select p.nome from p in Pessoa;

2) Select p.nome
from p in professores
where p.salario ≥ 5000;

3) Select e.nome
from e in estudantes
where e.curso.nomeCurso = 'Ciência da Computação';

4) Select e.nome
from e in estudantes
where e.curso.nomeCurso = 'Ciência da Computação'
and e.orientador.nome = 'José da Silva';

5) define disciplina_view(nome_disciplina) as
element (select d
from d in disciplinas
where d.nome like nome_disciplina);

6) disciplina_view("Banco de Dados II").oferecida_por

7) Select t.ano, t.semestre, t.professor.nome
from t in disciplina_view("Banco de Dados II").tem_turmas;

nome, formação e média geral de todos os alunos de ciências da computação

8) Select Struct(e.nome,
formação: (select Struct (grau: f.grau, ano: f.ano, esc: f.escola)
from f in formação),
avg(select n.nota
from n in notas
where n.estudante = e))
from e in estudantes
where e.cursa.nomeCurso = "Ciências da Computação";

ou

Select Struct(e.nome,
formação: (select Struct (grau: f.grau, ano: f.ano, esc: f.escola)
from f in formação),
avg(select n.nota from n in e.turmas_completadas.nota)
from e in estudantes
where e.cursa.nomeCurso = "Ciências da Computação";