



Curso de Java

Classes de Dados e Classes de Controle



©Todos os direitos reservados Klais®





O Papel das Classes

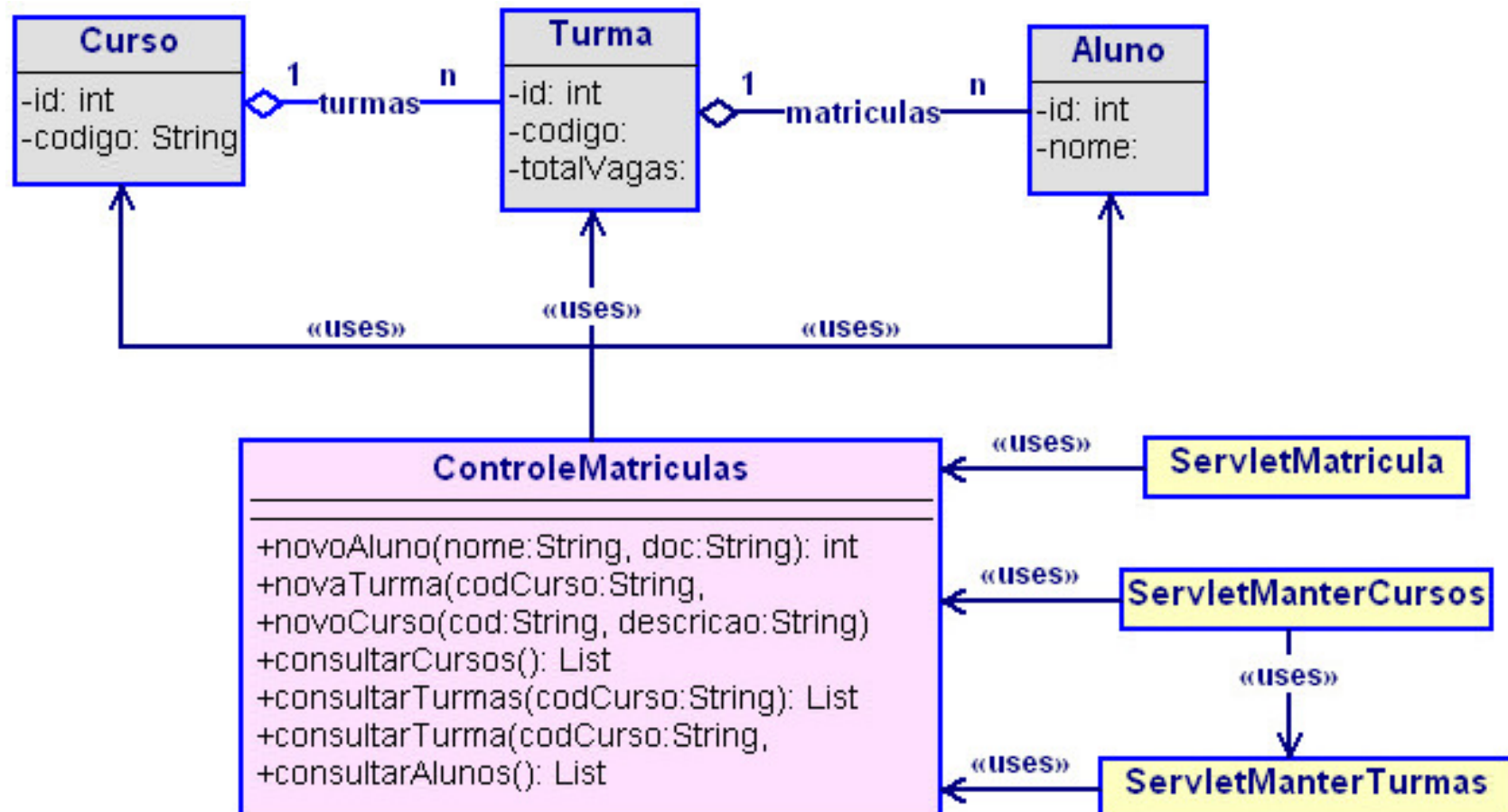
- Numa aplicação, as classes podem ter diferentes papéis em função da forma como são utilizadas.
- Ao se planejar a aplicação é importante caracterizar o papel das classes que a compõem.
- A prática tem mostrado uma clara tendência quanto aos papéis ou categoria das classes típicos:
 - classes de dados
 - classes de controle
 - classes de interface



O Papel das Classes

- Classes de Dados ou Classes de Entidade:
 - São utilizadas para ‘carregar’ os dados da aplicação.
 - Os métodos se restringem à manipulação dos atributos do próprio objeto (não têm a visão ‘do todo’).
- Classes de Controle:
 - Implementam as regras de negócio da aplicação.
 - Seus métodos conseguem tratar conjuntos de objetos de outras classes.
- Classes de Interface:
 - Compõem as ‘fronteiras’ da aplicação com o mundo externo.
 - Interface de usuário, outras aplicações, etc.

Um Exemplo





Um exemplo

- Nesse exemplo
 - A classe Aluno é uma classe de dados ou classe de entidade.
 - ControleMatricula é uma classe de Controle
 - ServletMatricula, ServletManterCursos, ServletManterTurmas são classes de interface, responsáveis pela interação dos usuários através de páginas web.
 - As classes Curso e Turma têm funcionalidades relativas a dados e relativas a controle, portanto não se encaixam nas categorias acima.
 - As classes ControleMatricula, Aluno, Turma e Curso podem ser agrupadas para constituir um componente onde ControleMatricula seria responsável por disponibilizar os serviços à aplicação.



Um exemplo: a *classe Aluno*

```
package matricula;
public class Aluno {
    private int id;
    private String nome;
    private String doc;
    private int idade;
    private static int counter = 0;

    public Aluno(String n, String d, int i){
        id = counter++; nome = n; doc = d; idade = i;
    }

    public String getNome() { return nome; }
    public String getDoc() { return doc; }
    public int getIdade() { return idade; }
    public int getId() { return id; }

    public void setNome(String n) { nome = n; }
    public void setDoc(String d) { doc = d; }
    public void setIdade(int i) { idade = i; }
    public static int getCounter() { return counter; }
}
```



Um exemplo: a *classe Aluno*

- Nesse exemplo
 - Todos os atributos são protegidos e são acessados apenas através de métodos `get*()` e `set*()`.
 - Essa disciplina pode ser usada para garantir a coerência entre os atributos.
 - O atributo `counter`, que indica o número de objetos criados, é um atributo da classe:

```
static int counter = 0;
```
 - O método `getCounter()`, que retorna o valor do atributo `counter` também é associado à classe:

```
static int getCounter();
```



Um exemplo: a *classe Turma*

```
package matricula;
import java.util.*;
public class Turma {
    private int id;
    private String cod;
    private int totalVagas;
    private HashMap matriculas;
    private static int counter = 0;

    public Turma(String c, int tot){ ... }
    ...
    public int getDisp() {
        return totalVagas - matriculas.size();
    }
    public int getNALunos() { return matriculas.size(); }

    public void setTotalVagas(int tot) throws Exception{
        if(tot < matriculas.size()) throw new Exception(...);
        totalVagas = tot;
    }
    public void incluiAluno(Aluno aluno) throws Exception{ ... }
    public void excluiAluno(int id) throws Exception{...}
    public List getMatriculas(){... }
}
```




Um exemplo: a *classe Turma*

- Nesse exemplo
 - Os atributos também são acessados apenas através de métodos `get*()` e `set*()`.
 - O método `getNAlunos()` retorna o número de alunos matriculados, obtido a partir do atributo **`matriculas`**.
 - O método `setTotalVagas()` gera uma exceção se o novo valor de `totalVagas` for incompatível com o número de alunos matriculados.



Classes de Dados

- Numa classe de dados “pura”
 - Têm como objetivo manter dados da aplicação em seus atributos.
 - Na maioria das vezes os atributos são persistidos em banco de dados ou arquivos.
 - Os métodos são voltados à manipulação dos atributos.
 - Os métodos não tratam regras de negócio.
 - Eventualmente os métodos se preocupam em manter os valores dos atributos coerentes entre si.



Classes de Controle

- Numa classe de controle “pura”
 - Têm como objetivo principal a implementação dos métodos relativos às funcionalidades e regras de negócio de uma dada área da aplicação.
 - Na maioria das vezes os atributos são aqueles necessários para a execução dos métodos.
 - Os valores dos atributos em geral se referem a dados de configuração da aplicação.
 - Na maioria dos casos, não têm um construtor para a criação de objetos e todos os métodos são ‘estáticos’.



Um exemplo: a *classe ControleMatricula*

```
package matricula;

import java.util.*;
public class ControleMatricula {
    private static LinkedHashMap alunos = new LinkedHashMap();
    private static LinkedHashMap turmas = new LinkedHashMap();
    private static LinkedHashMap cursos = new LinkedHashMap();

    public static Aluno novoAluno(String nome, String doc, int idade){...}
    public static Turma novaTurma(Curso c, String cod, int nVagas) {...}
    public static Curso novoCurso(String cod, String descr){ ... }
    public static void matricula(Aluno a, Curso c, Turma t){...}
    public static List consultaAlunos(){ ... }
    public static List consultaAlunos(Curso c){ ... }
    public static List consultaAlunos(Curso c, Turma t) { ... }
    public static List consultaCursos(){ ... }
    public static List consultaTurmas(Curso c) { ...}

}
```