Centro Federal de Educação Tecnológica de Minas Gerais
Laboratório de Programação de Computadores II
Prof. Eduardo Cunha Campos

Prática 2

Objetivos:

- Introdução aos conceitos de classe, objeto, método e atributo.

1. Introdução aos conceitos de classe, objeto, método e atributo.

Java é uma linguagem orientada por objetos. Esta seção apresenta os dois principais conceitos deste paradigma: classe, objeto, atributos e métodos.

Classe: "é um *modelo* a partir do qual podem ser instanciados objetos." (Staa, 2001).

Objeto: é uma instância de uma classe.

Os conceitos de classes e objetos permitem a construção de software a partir de modelos computacionais mais próximos do mundo real, o que é referenciado na literatura como redução de *gap* semântico (Meyer, 1988).

Exemplo: a construção de um sistema de gestão acadêmica.

Quais são as entidades envolvidas em um sistema deste tipo?
Professores, alunos, disciplinas, etc.

Como implementar esse sistema no paradigma estruturado?

Como implementar esse sistema no paradigma OO?

Primeiro passo: a construção de qualquer sistema, seja em qual paradigma for, requer o estudo do problema a ser solucionado.

Segundo passo: propor uma solução computacional para o problema. Para isso, é preciso:

- identificar as entidades a serem modeladas;
- modelar computacionalmente tais entidades.

Na OO, essas entidades correspondem naturalmente a *classes* de *objetos*.

No sistema exemplo:

- São classes: professor, aluno, disciplina e aluno.

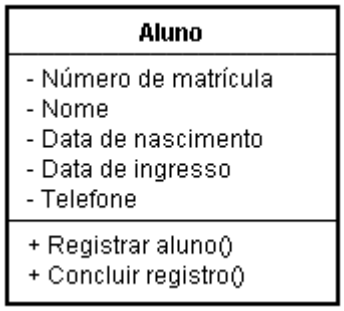
- São objetos da classe professor: Professora Maria, Professor José etc.
- São objetos da classe aluno: Josiane, Dayse, Morvan, Paulo, Ivan, Karina, Antonio, Renato, Lucas, Ricardo, Arlene, Marcos etc.
- São objetos da classe disciplina: Linguagens de Programação, Organização e Gerenciamento de Arquivos, Algoritmos e Técnicas de Programação, Engenharia de Software, etc.

Uma classe é constituída de atributos e métodos:

Atributo: são os membros de dados da classe. Determinam as características de uma classe de objetos.

Método: são as operações, os membros de função da classe. Determinam o comportamento de uma classe de objetos.

Exemplo: a classe Aluno.



- Exercício: identifique atributos e métodos das classes professor e disciplina.

1. Entendendo o método *main*

```
class Programa1{

   /************************************/
   /* todo main deve ter este cabecalho */
   /************************************/
```

```
   public static void main(String[] args){
      System.out.println("Hello, world!");
   }
}
```

Este pequeno exemplo de programa introduz alguns conceitos importantes na linguagem Java, dentre eles:

- A menor unidade de compilação em Java é uma classe.
- O nome do arquivo do programa fonte deve ter a extensão .java e deve ser idêntico ao nome da classe pública que ele contem. (Obs.: o conceito de *público* será discutido em maiores detalhes posteriormente)
- `public static void main(String[] args)`

é parte de todo aplicativo Java. A execução de todo aplicativo Java começa pelo *main*.

`public` : indica que o método é público. Se o método main não for público, ao tentar executar o programa, será exibida a mensagem de erro *Main method not public*.

`static`: indica que o método main existe e pode ser utilizado mesmo se nenhum objeto da classe tenha sido instanciado. Se o método main não for static, ao tentar executar o programa, será exibida a mensagem de erro *Exception in thread "main" java.lang.NoSuchMethodError: main*.

`void`: indica que o método executará as instruções que estão entre as { }e não retornará valor algum.

`String[] args`: define o parâmetro que pode ser passado para o programa principal.

O programa abaixo ilustra o uso de passagem de parâmetros para *main*.

```
class Programa2{
   public static void main(String[] args){

     System.out.println("Recebi " + args.length + "argumentos.");
     for (int i=0; i<args.length; i++)
         System.out.print(args[i] + " ");

     System.out.println();
   }
}
```

2. Programação Orientada por Objetos em Java

Esta seção tem por objetivos que o aluno seja capaz de:

- Entender a estrutura básica de uma classe em Java.
- Criar e utilizar objetos.
- Entender os conceitos de atributos e métodos.

3.1 Classes em Java

Uma classe representa a implementação de um novo tipo. Para entender a estrutura básica de classes em Java, utilizaremos o aplicativo simples a seguir.

```
package meuprojeto;

public class Aluno {

    private String nome;
    private String matricula;
    private String situacao;

    public Aluno (String n, String m){
      nome = n;
      matricula = m;
      situacao = "Nao matriculado";
    }

    public void alterarNome(String n){
      nome = n;
    }

    public void matricular(){
      situacao = "Matriculado";
    }

    public void cancelarMatricula(){
      situacao = "Nao matriculado";
    }

    public String obterNome(){
       return nome;
    }

    public String obterMatricula(){
     return matricula;
    }
```

```
    public String obterSituacao(){
       return situacao;
    }

    public void imprimir(){
      System.out.println("** Dados do Aluno ***");
      System.out.println("Nome: " + nome);
      System.out.println("Matricula: " + matricula);
      System.out.println("Situação: " + situacao);
    }

}
```

A classe abaixo exemplifica o uso da classe Aluno:

```
package meuprojeto;

public class Principal {
  public static void main(String[] args){
    Aluno a, b;

    a = new Aluno("João", "1");
    a.matricular();
    a.imprimir();

    b = new Aluno("Maria", "2");
    b.imprimir();

  }
}
```

- **Exercício 1**: implemente uma classe **Ponto** que representa um ponto de coordenadas x e y. Para esta classe, implemente os métodos que permitem alterar e recuperar as coordenadas do ponto (métodos *get* e *set*). Implemente uma classe principal que crie três objetos do tipo ponto, modifique os dados dos objetos criados e mostre seus dados.

- **Exercício 2**: Implemente uma classe **Contato** que tenha os seguintes dados: nome e telefone (String) e os respectivos métodos *get* e *set*. Os dois atributos são alteráveis. Implemente uma aplicação (uma outra classe que possua um método *main*) que crie três contatos, modifique e exiba os dados dos contatos criados.