Classes

Laboratório de Programação III

Professor Mateus Jung

INSTITUTO FEDERAL
Sul-rio-grandense

Classe
Introdução

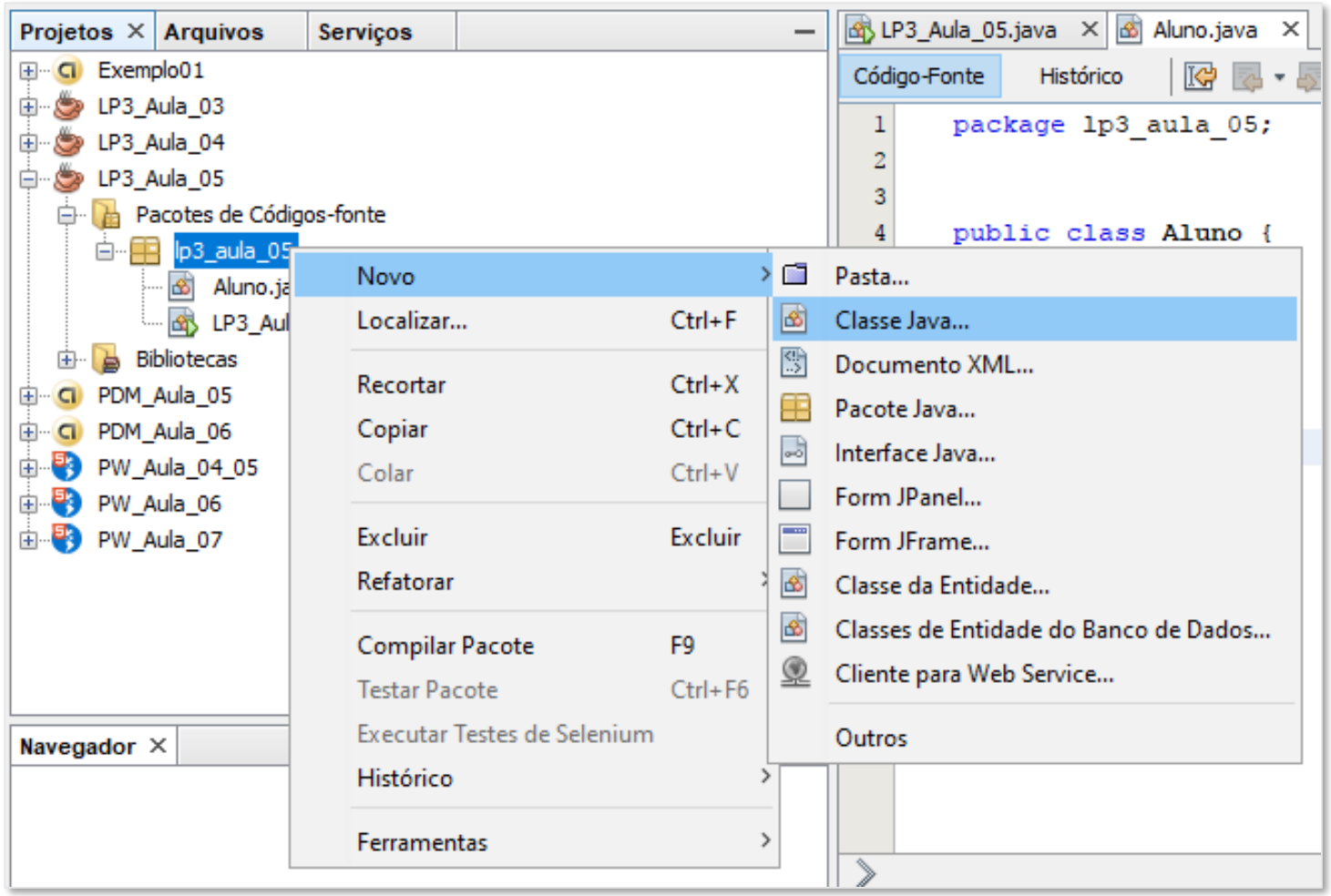
INSTITUTO FEDERAL
Sul-rio-grandense

- Classe é uma definição de um objeto
- Uma classe não pode ser manipulada, ela apenas especifica algo
- Contêm todas as propriedades e métodos de um objeto
- A classe é o molde para a construção de um objeto

Classe

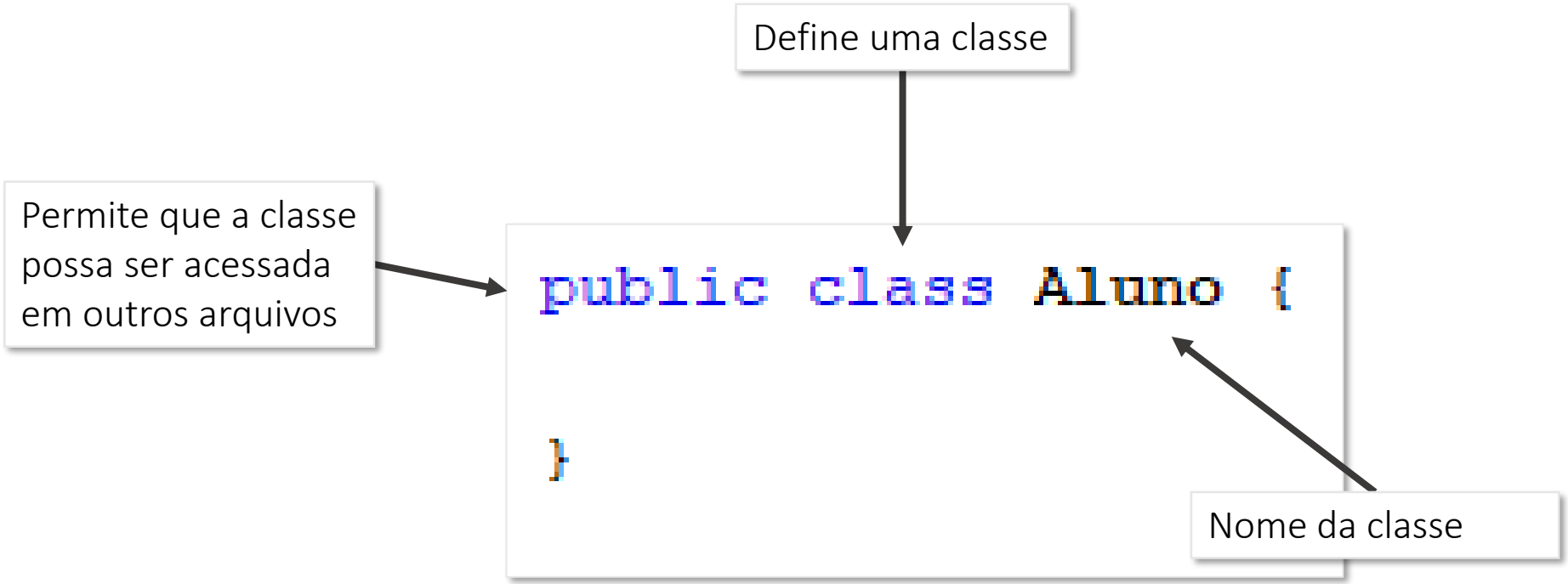
Criação de uma classe Java

INSTITUTO FEDERAL
Sul-rio-grandense



Classe

Construtor



Classe
Propriedades

INSTITUTO FEDERAL
Sul-rio-grandense

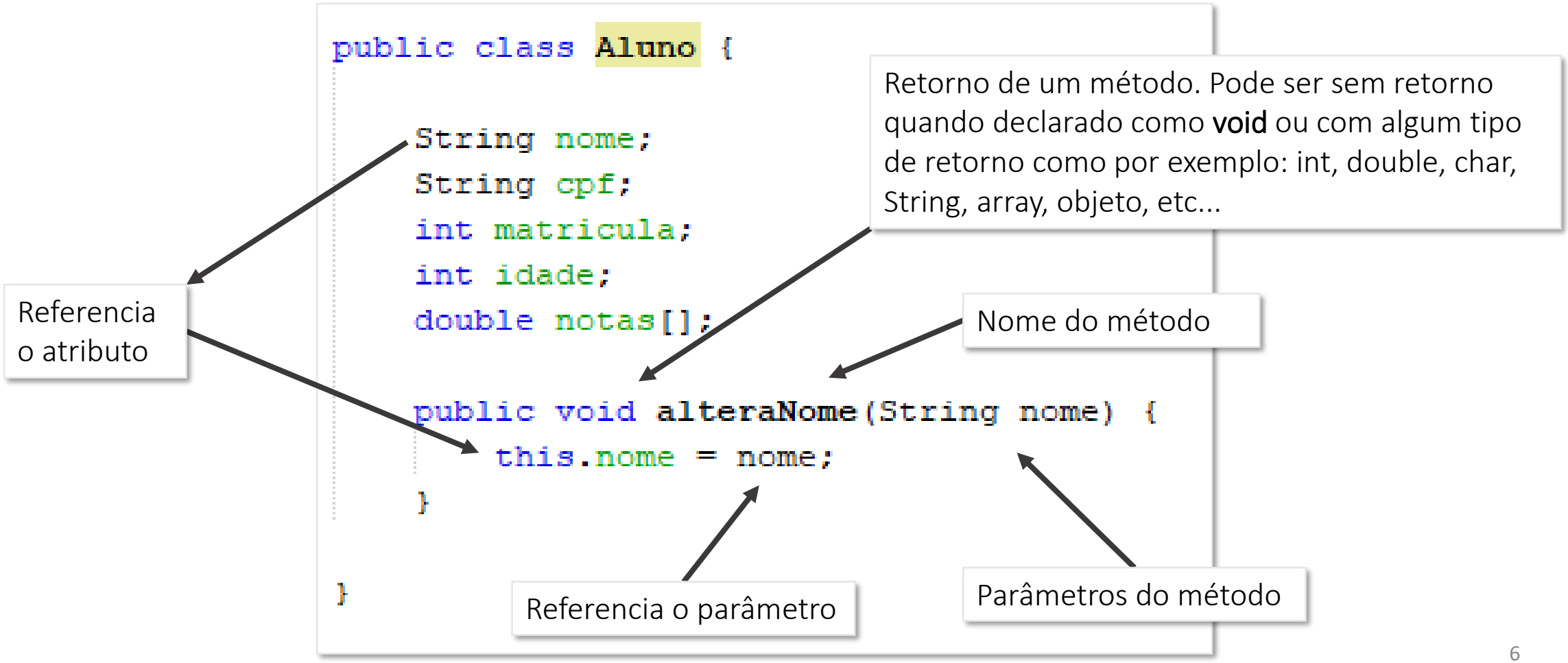
```
public class Aluno {

    private String nome;

    private String cpf;

    private int matricula;

    private LocalDate dataNascimento;

    private int idade;

    private double notas[];
```

Atributos da classe

Um atributo de classe pode ser um objeto de outra classe

LocalDate é uma classe utilizada para armazenar dados do tipo data

Classe
Métodos



Classe
Métodos

INSTITUTO FEDERAL
Sul-rio-grandense

Tipo de dado do retorno

```
public double media() {
    double soma = 0.0;

    for(double nota : this.notas) {
        soma += nota;
    }
    double media = soma / this.notas.length;

    return media;
}
```

Valor a ser retornado

Classe

Construtor

INSTITUTO FEDERAL
Sul-rio-grandense

Declaração de um construtor

Um construtor sempre tem o mesmo nome da classe

```java
public class Aluno {

    String nome;
    String cpf;
    int matricula;
    int idade;
    double notas[];

    public Aluno() {
        this.nome = "";
        this.cpf = "00.000.000-00";
        this.matricula = 0;
        this.idade = 0;
        this.notas = new double[4];
    }

    public void alteraNome(String nome) {
        this.nome = nome;
    }

}
```

Um construtor especifica tudo o que será feito quando um objeto for criado

Exemplo inicializando todos os atributos

Classe

Instanciando um objeto

INSTITUTO FEDERAL
Sul-rio-grandense

```
public class Aplicacao {

    public static void main(String args[]) {
        Aluno primeiroAluno = new Aluno();

        Aluno segundoAluno;
        segundoAluno = new Aluno();

        Aluno terceiroAluno = new Aluno();
        terceiroAluno = new Aluno();

    }
}
```

Antes do construtor, o objeto possui o valor **null**

O objeto tem seus valores apagados toda vez que o construtor é chamado, pois uma nova posição de memória e dada a ele

Classe

Alterando um objeto

INSTITUTO FEDERAL
Sul-rio-grandense

```java
public class Aplicacao {

    public static void main(String args[]) {
        Aluno aluno = new Aluno();

        aluno.nome = "Fulano de Tal";
        aluno.cpf = "123.456.789-01";
        aluno.matricula = 12345;
        aluno.notas[0] = 6.0;
        aluno.notas[1] = 7.5;
        aluno.notas[2] = 9.3;
        aluno.notas[3] = 8.2;
        System.out.println("A média do aluno é " + aluno.media());
    }
}
```

Valores só podem ser atribuídos após a construção do objeto

Invocação do método média que retorna o cálculo da média do aluno

INSTITUTO FEDERAL
Sul-rio-grandense

Encapsulamento

Encapsulamento

Acesso a atributos

INSTITUTO FEDERAL
Sul-rio-grandense

O modificador de acesso **public** permite o acesso direto ao atributo em outra classe

O modificador de acesso **private** nega o acesso direto ao atributo em outra classe

```
public class Aluno {

    public String nome;

    private String cpf;
    private int matricula;
    private int idade;
    private double notas[];
```

Caso não seja definido um modificador de acesso, é atribuído default por padrão

```
public class Aplicacao {

    public static void main(String args[]) {
        Aluno aluno = new Aluno();

        aluno.nome = "Fulano de Tal";
        aluno.cpf = "123.456.789-01";
        aluno.matricula = 12345;
        aluno.notas[0] = 6.0;
        aluno.notas[1] = 7.5;
        aluno.notas[2] = 9.3;
        aluno.notas[3] = 8.2;

    }
}
```

Encapsulamento

Métodos get e set

INSTITUTO FEDERAL
Sul-rio-grandense

Para acesso e modificação de atributos privados, é uma **boa prática** utilizar a convenção **get** e **set**

```
public class Aluno {

    private String nome;
    private String cpf;
    private int matricula;
    private int idade;
    private double notas[];

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }

    public String getCpf() {
        return cpf;
    }
```

Um método nomeado como get, seguido do nome da variável retorna o valor da mesma

Um método nomeado como set, seguido do nome da variável com um parâmetro de mesmo tipo da variável, altera o valor da mesma

Encapsulamento

Métodos get e set

INSTITUTO FEDERAL
Sul-rio-grandense

```
public class Aplicacao {

    public static void main(String args[]) {
        Aluno aluno = new Aluno();

        aluno.setNome("Fulano de Tal");
        aluno.setCpf("123.456.789-01");
        aluno.setMatricula(12345);

        double[] notas = {6.0, 7.5, 8.3, 9.2};
        aluno.setNotas(notas);

        System.out.println("A média do aluno " +
                           aluno.getNome() +
                           " é " + aluno.media());
    }
}
```

INSTITUTO FEDERAL
Sul-rio-grandense

Exercício

Exercício

INSTITUTO FEDERAL
Sul-rio-grandense

1. Implemente a classe Aluno dos exemplos utilizados

2. Levando em consideração que um aluno pode ter várias notas e cada nota está atribuída a uma disciplina que pode ter mais de uma nota, melhore a propriedade “notas” elaborando uma solução que satisfaça essa condição

3. Construa uma classe Turma, que pode conter vários alunos

4. Tendo em vista que a idade de um aluno pode ser obtida a partir de sua data de nascimento, remova a propriedade “idade” e elabore um método que retorne a idade do aluno

5. Crie uma aplicação onde possa ser criada uma turma e, a partir de sua criação, possam ser inseridos alunos na mesma