Alisson RS
www.alissonrs.com.br
moodle.cefetmg.br

Linguagem de programação II

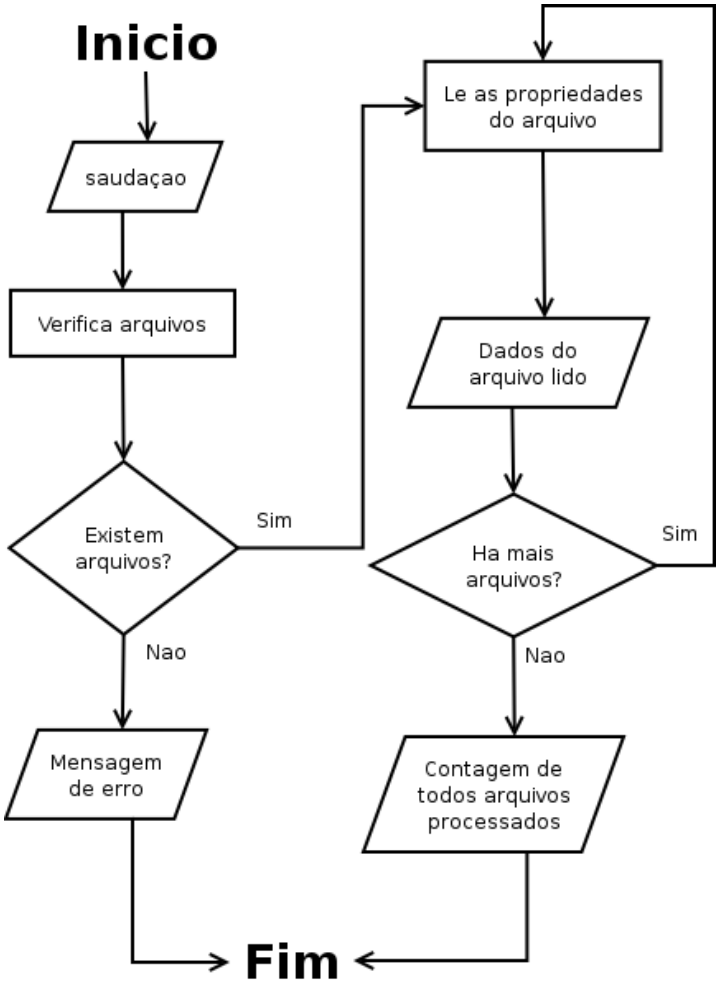
Classes e objetos.

Classes e objetos

- Programação orientada a objetos
  - Uma nova forma de pensar seus programas.
  - Mudança do que deve ser feito por quem faz o quê.
- Facilidade no entendimento do programa.
- Encapsulamento
  - Ocultar partes complexas e de conhecimento desnecessário por parte dos usuários o código.
- Reuso
  - Por meio de instanciações, sobrecargas e herança.

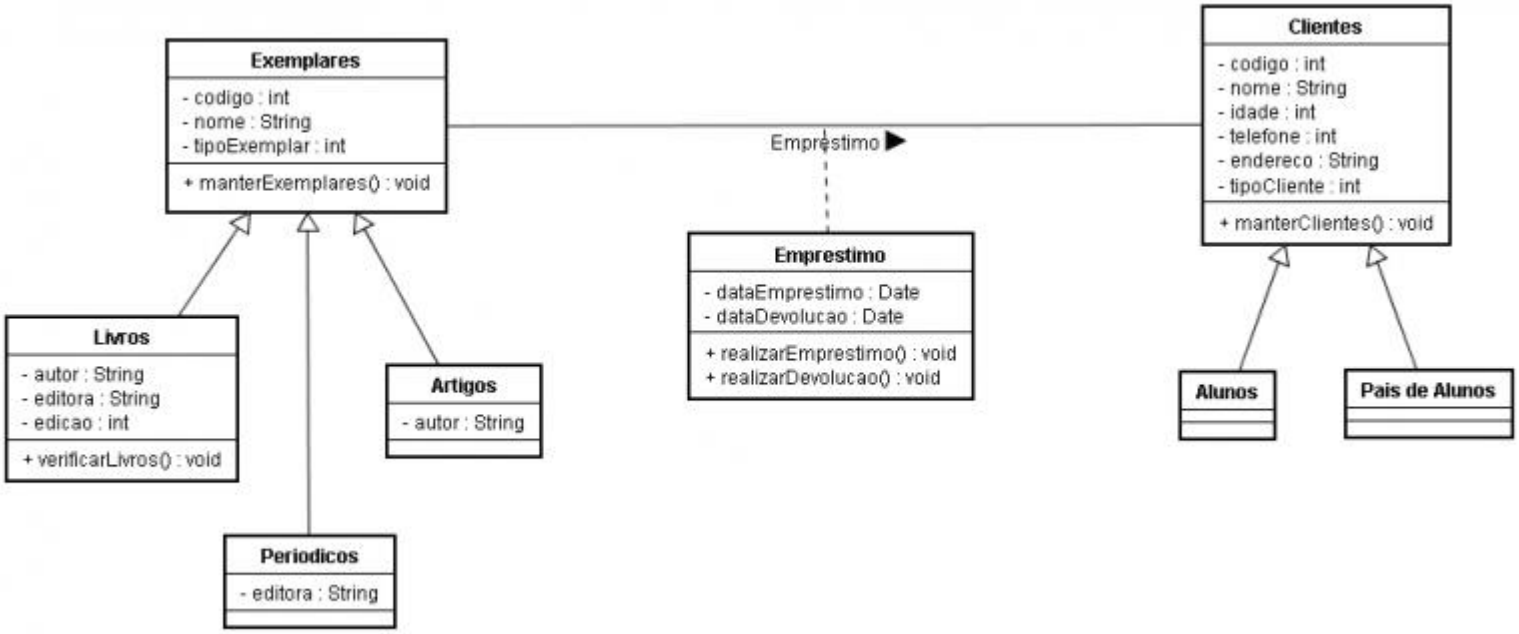
Classes e objetos

- O que fazer?
- Métodos que chamam métodos.



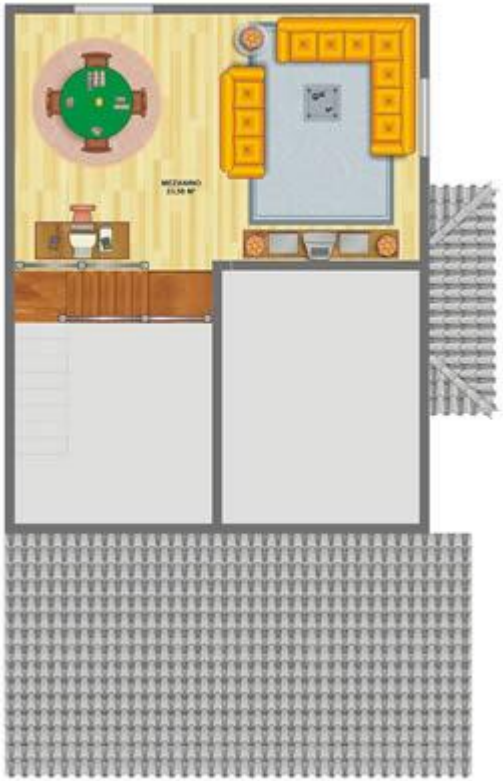
Classes e objetos

- Quem deve fazer?
- Objetos que interagem entre si por meio de métodos.



Classes e objetos

- Classe -> Planta
- Objeto -> Planta executada e pronta para uso.



Classes e obietos

```java
package br.com.alissonrs.aula13;

public class ExemploClasse {

    public static int contador = 0;
    public int numero = 0;
    private String nome = "";

    public ExemploClasse() {
        contador++;
    }

    public ExemploClasse(int novoNumero, String novoNome) {
        this.numero = novoNumero;
        this.nome = novoNome;
        contador++;
    }

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

    public static int contarInstancias(){
        return contador;
    }

}
```

Classes e objetos

- Classe pública – linha 3
  - Nome da classe + .java igual nome do arquivo.
    - Nome do arquivo: ExemploClasse.java
- Package
  - Diretório em que foi colocado o código da classe a partir do diretório raiz do programa.
    - Cada ponto é um subdiretório.
    - Serve para organizar fisicamente e hierarquicamente seu código.

```
1 package br.com.alissonrs.aula13;
2
3 public class ExemploClasse {
4
```

Classes e objetos

```
5     public static int contador = 0;
6     public int numero = 0;
7     private String nome = "";
8
```

- Atributos (variáveis)
  - Armazenam informações dos objetos ou da classe.
- De classe (linha 5)
  - O mesmo para todos os objetos.
  - É criado durante a leitura do programa.
  - Tem o modificador static
  - Pode ser acessado pelo nome da classe
    - Não precisa ter nenhum objeto criado.

(ExemploClasse.contador)

Classes e objetos

- Atributos (variáveis)
  - Armazenam informações dos objetos ou da classe.
- De Instância (linhas 6 e 7)
  - Um diferente para cada objeto criado.
  - É criado durante a instanciação do objeto.
    - A partir da palavra reservada **new**.
  - Pode ser acessada:
    - Pelo nome da variável que recebe o objeto (se pública)
    - Pelo seu nome dentro do código da classe.

```
5     public static int contador = 0;
6     public int numero = 0;
7     private String nome = "";
8
```

Classes e objetos

- Métodos
  - Parte do código em que tarefas são executadas por aquele objeto.
    - Comandos de ação fora dos métodos ou blocos de execução implicarão erro.
  - Possuem:
    - Modificador de visibilidade (privado, protegido ou público)
    - Valor de retorno (void se não retorna nada).
    - Nome
    - Argumentos de entrada.
  - Exemplo de assinatura de um método.

public void setNome(String novoNome)

Classes e objetos

- Métodos construtores
  - Método especial de inicialização do objeto.
    - Mesmo nome da classe.
    - Implementação não obrigatória
    - **Não possui valor de retorno em sua assinatura.**
      - Seu valor de retorno é uma referência ao objeto criado.
    - Pode ser sobrecarregado (o que é isso?)

- Linhas 9 e 13.

```
9     public ExemploClasse() {
10        contador++;
11    }
12
13    public ExemploClasse(int novoNumero, String novoNome) {
14        this.numero = novoNumero;
15        this.nome = novoNome;
16        contador++;
17    }
18
```

Classes e objetos

- Métodos construtores
  - Método especial de inicialização do objeto.
    - Mesmo nome da classe.
    - Implementação não obrigatória
    - **Não possui valor de retorno em sua assinatura.**
      - Seu valor de retorno é uma referência ao objeto criado.
    - Pode ser sobrecarregado (o que é isso?)

- Linhas 9 e 13.

```
9     public ExemploClasse() {
10        contador++;
11    }
12
13    public ExemploClasse(int novoNumero, String novoNome) {
14        this.numero = novoNumero;
15        this.nome = novoNome;
16        contador++;
17    }
18
```

Classes e objetos

- Sobrecarga de Métodos
  - Mais de um método com o mesmo nome em uma classe.
  - Assinaturas devem ser diferentes quanto aos argumentos para:
    - Número
    - Quantidade
    - Ordem
  - Linhas 9 e 13
    - Construtor sobrecarregado.

```
9     public ExemploClasse() {
10        contador++;
11    }
12
13    public ExemploClasse(int novoNumero, String novoNome) {
14        this.numero = novoNumero;
15        this.nome = novoNome;
16        contador++;
17    }
18
```

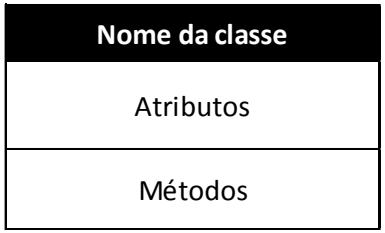
Classes e objetos

- Sobrecarga de Métodos
  - Mais de um método com o mesmo nome em uma classe.
  - Assinaturas devem ser diferentes quanto aos argumentos para:
    - Número
    - Quantidade
    - Ordem
  - Linhas 9 e 13
    - Construtor sobrecarregado.

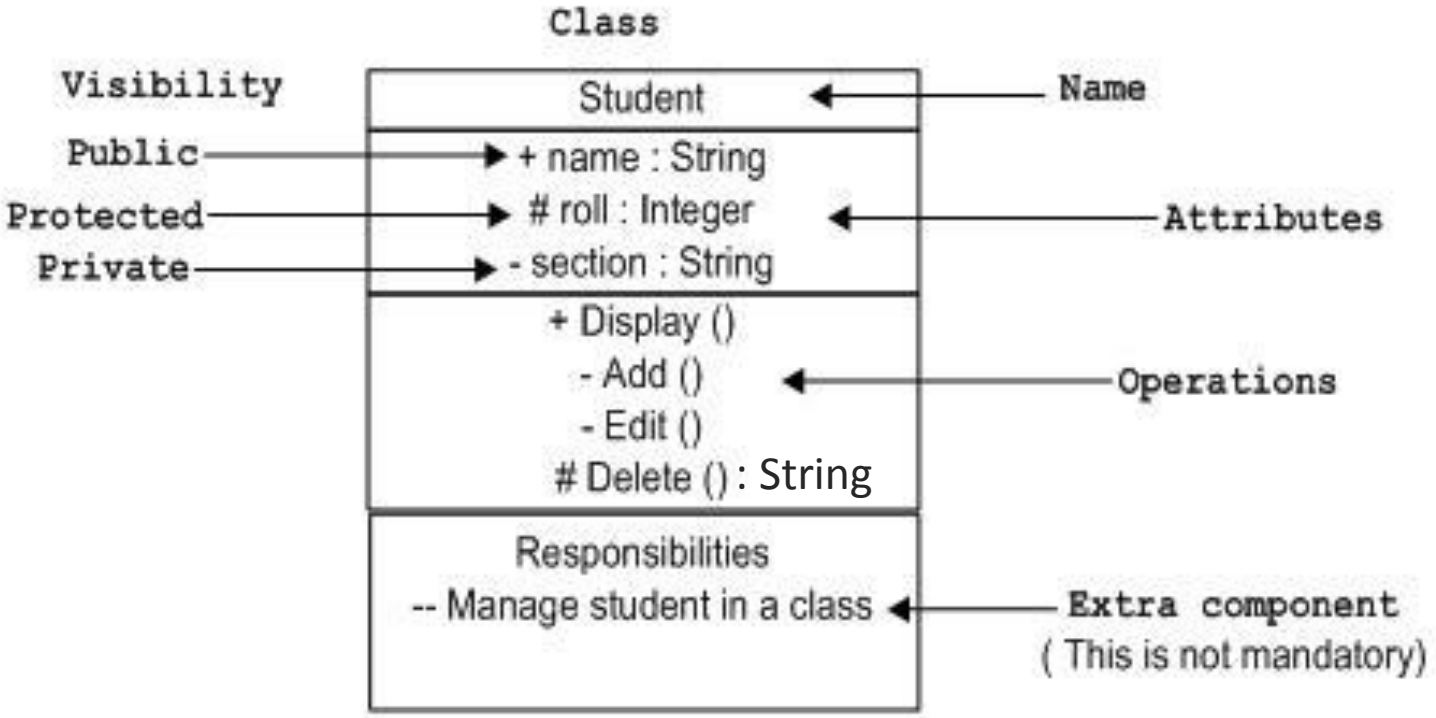
```
9     public ExemploClasse() {
10        contador++;
11    }
12
13    public ExemploClasse(int novoNumero, String novoNome) {
14        this.numero = novoNumero;
15        this.nome = novoNome;
16        contador++;
17    }
18
```

Classes e objetos

- Representação
  - Diagrama UML – Diagrama de classes
    - Classes são representadas por entidades:
    - Nome da classe:
      - Modificador e o nome
    - Atributos:
      - Visibilidade (-privado, # protegido,+ público)
      - Modificador
      - Nome
      - Valor de retorno
    - Métodos:
      - Visibilidade (-privado, # protegido,+ público)
      - Modificador
      - Nome
      - Argumentos
      - Valor de retorno



Classes e objetos



Classes e objetos

- Tabela de modificadores de acesso.

	private	default	protected	public
mesma classe	sim	sim	sim	sim
mesmo pacote	não	sim	sim	sim
pacotes diferentes (subclasses)	não	não	sim	sim
pacotes diferentes (sem subclasses)	não	não	não	sim

www.alissonrs.com.br

Alisson RS