Programação III

Programação III - Aula 01

Prof. Vinicius F. Caridá

UEMG FRUTAL

Motivação e Objetivos

- Analisar, planejar e desenvolver sistemas de computação, utilizando uma ferramenta de programação do tipo RAD (Rapid Application Development) e orientada a objetos.

Ementa da Disciplina

1. Introdução à linguagem Java
2. Tipos de dados, Operadores, Estruturas de controle
3. Classes e objetos
4. Pacotes
5. Atributos e métodos estáticos
6. Delegação
7. Herança
8. Polimorfismo
9. Interface
10. Tratamento de Exceção
11. Classes para tratamento de dados: listas
12. Acesso à BD usando Java
13. Manipulação de Arquivos
14. Interface Gráfica

UEMG FRUTAL

Bibliografia

- **DEITEL**, Harvey; **DEITEL**, Paul. Java Como Programar -- 08ª ed -- São Paulo: Pearson, 2010.
- **SANTOS**, Rafael. Introdução à Programação Orientada a Objetos usando Java. Ed. Campus, 2003.
- **JANDL JR**, Peter. Java – Guia do Programador atualizado para Java 6. São Paulo: Novatec, 2007.
- **HORSTMANN**, Cay S; **CORNELL**, Gary. Core Java 2 Volume I – Fundamentos. Rio de Janeiro: Alta Books, 2005.
- **HORSTMANN**, Cay. Conceitos de Computação com Java. Porto Alegre: Bookman, 2009.
- ARNOLD, Ken; GOSLING, James; HOLMES, David. A Linguagem de Programação Java, 4ª edição. Porto Alegre: Bookman, 2007.

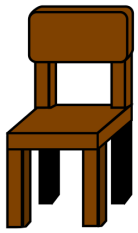
Classes e Objetos

- Classes definem as características de um conjunto de objetos.
  - Usadas para a modelagem de entidades do mundo real. Ex.: Pessoas, animais, produtos, etc.;
  - Define os dados e os procedimentos comuns a todos objetos da classe;
- São organizadas em pacotes;
- Em programação orientada a objetos (POO) as classes permitem a declaração de dois tipos de características:
  - os **atributos** são dados relativos a cada objeto
  - os **métodos**, que implementam as operações que podem atuar sobre os atributos

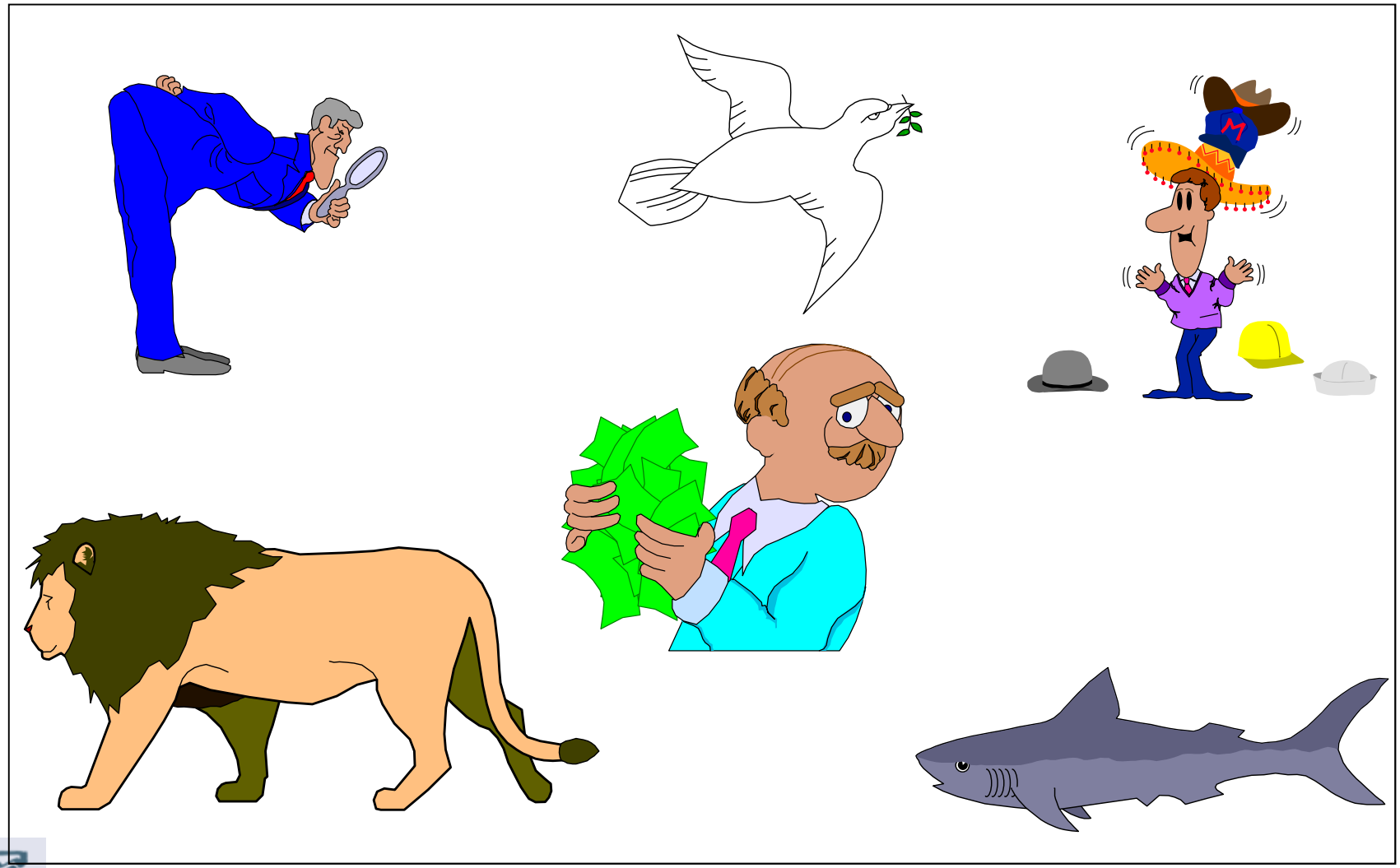
Classes e Objetos

- O que são objetos?

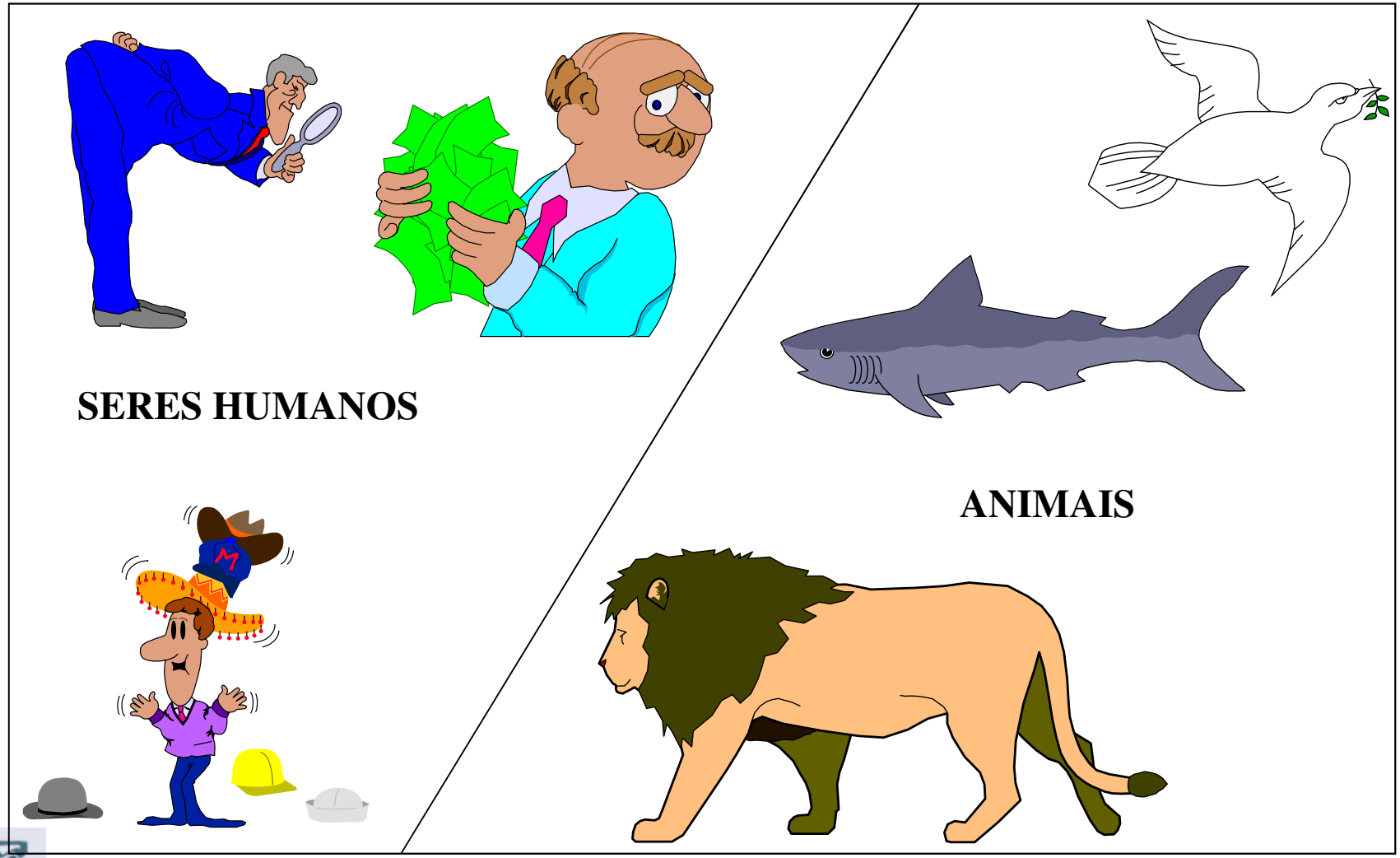
"Nosso mundo é um mundo de objetos, para se convencer disso basta tropeçar neles" (Castanheda)



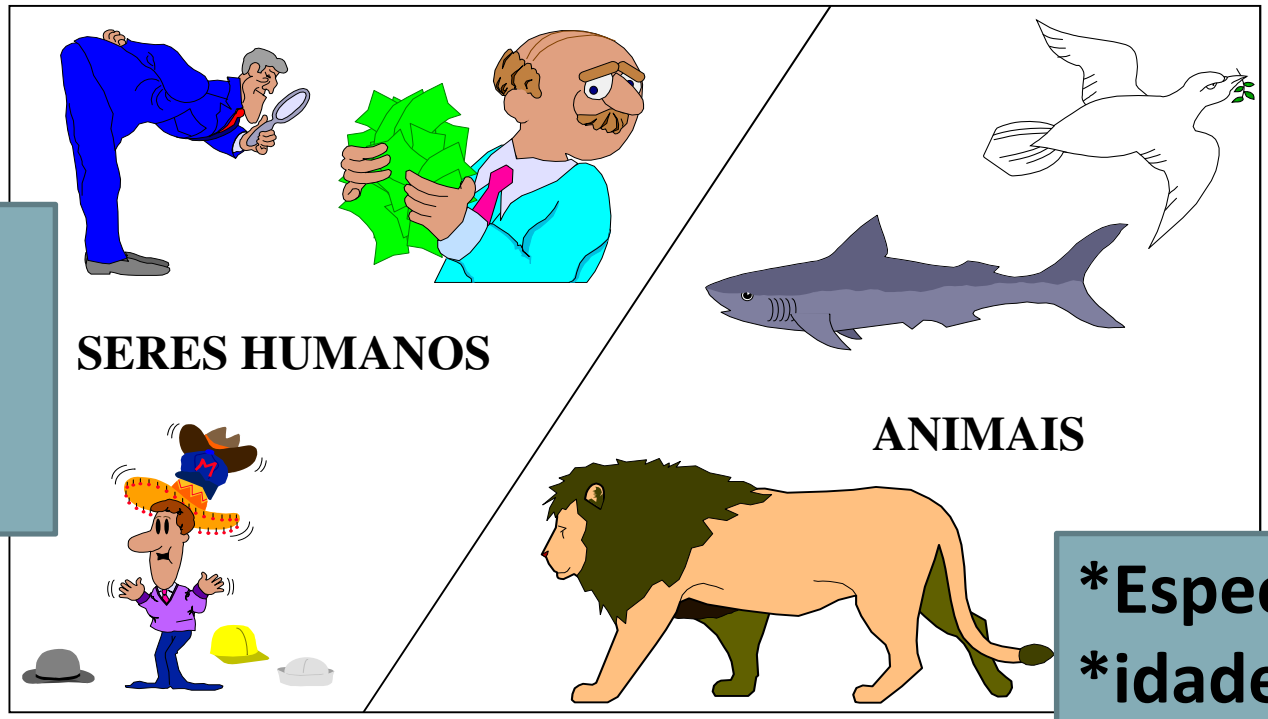
Objetos



Classificando os Objetos



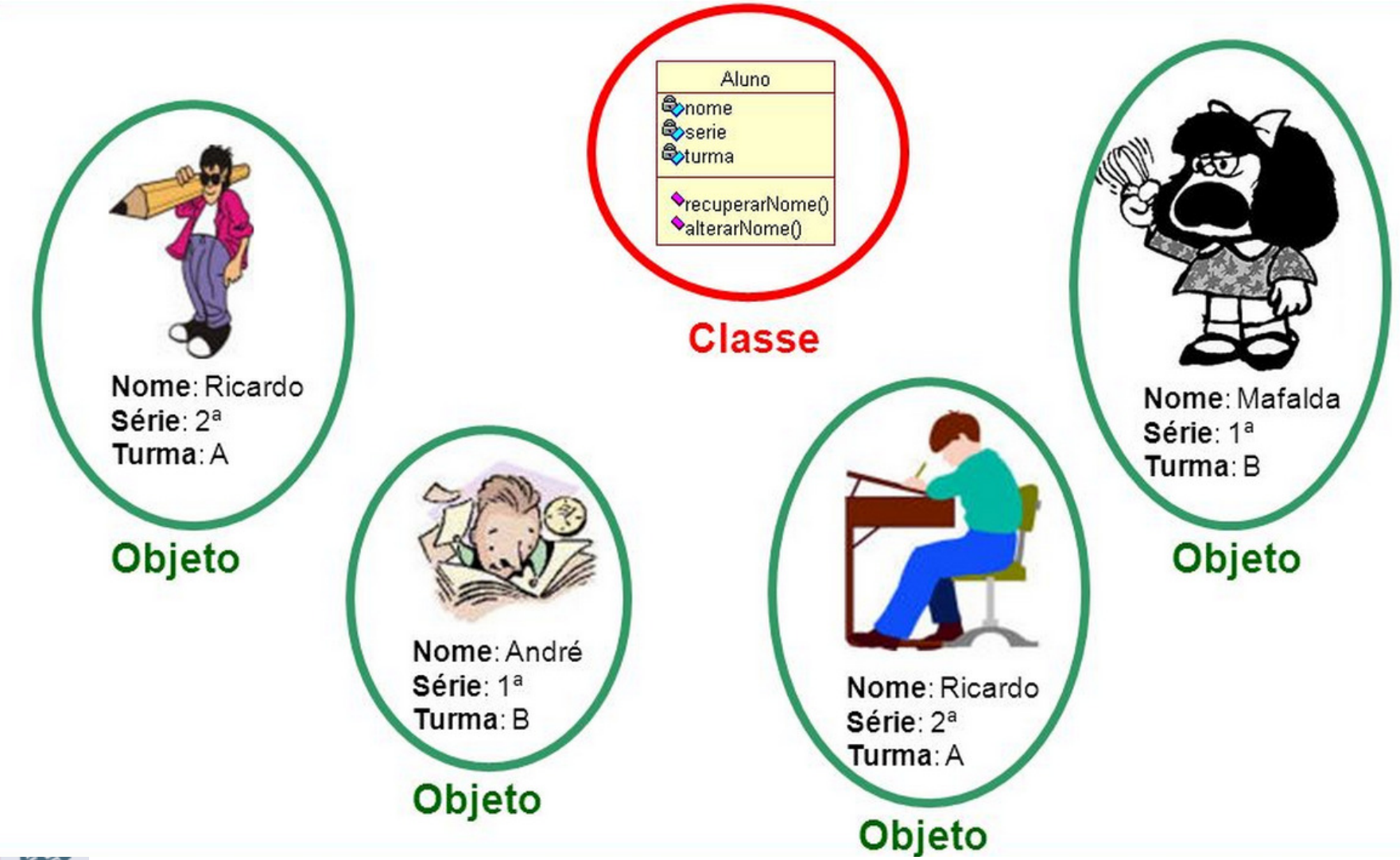
- O que são atributos?



***Nome;**
***idade;**
***CPF;**

***Especie;**
***idade;**
***localizacao;**

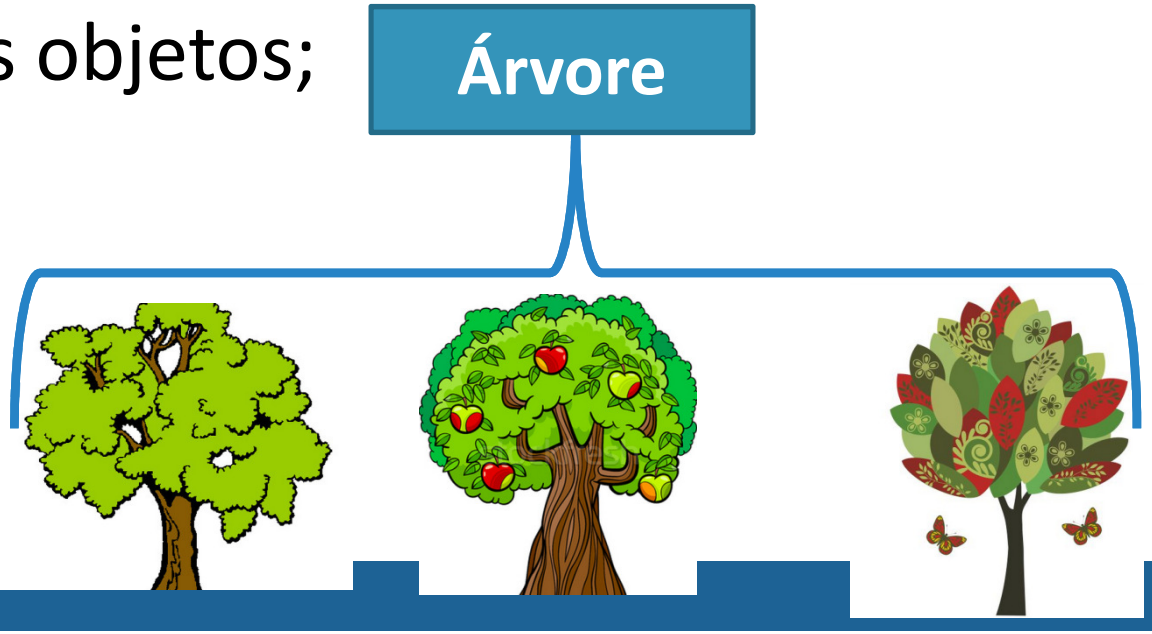
Outro exemplo



Classes

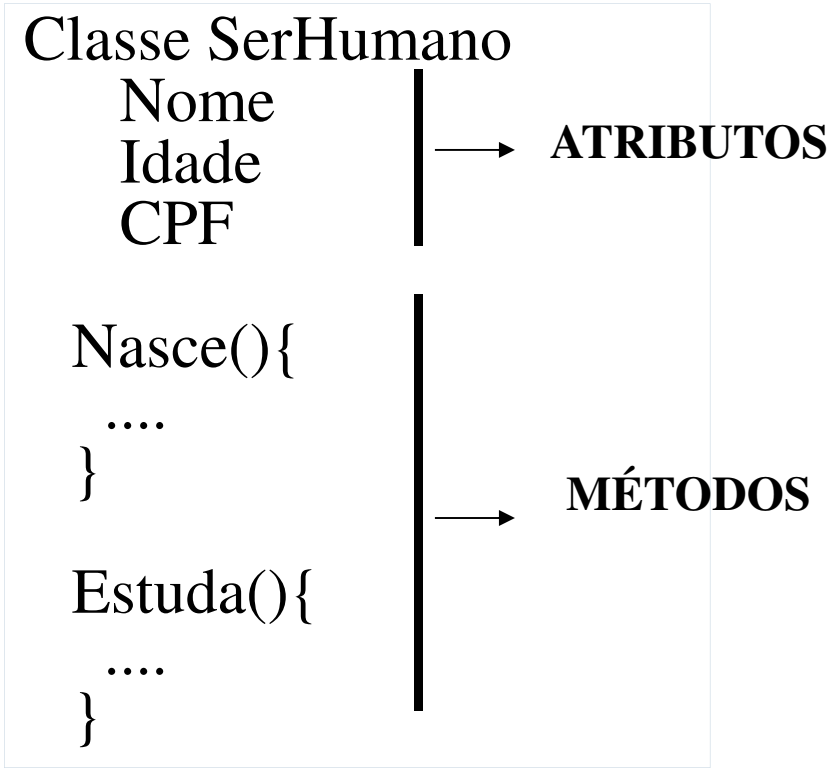
Dica: Escreva sempre o nome da classe em Maiúsculo!

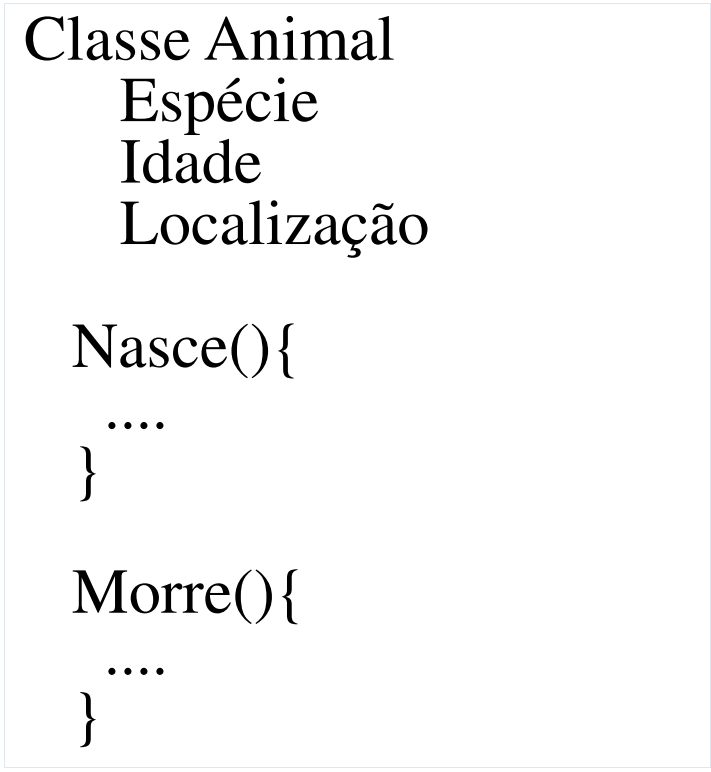
- Classes definem elementos da mesma natureza;
- Classes são abstrações de objetos;
  - "Modelos";
- Classes modelos características comuns a esses objetos;



Classes

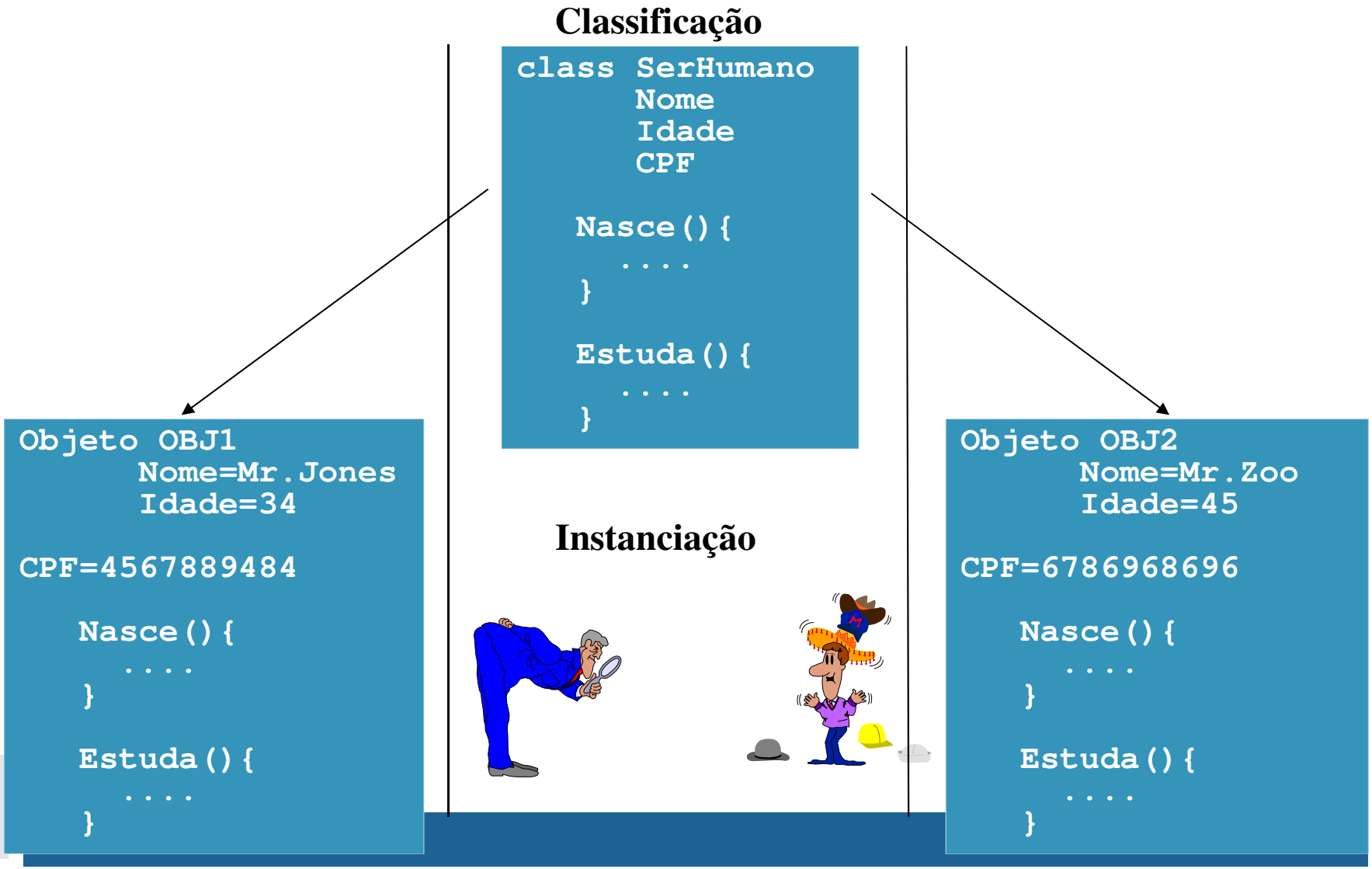
- São abstrações de conjuntos de objetos:





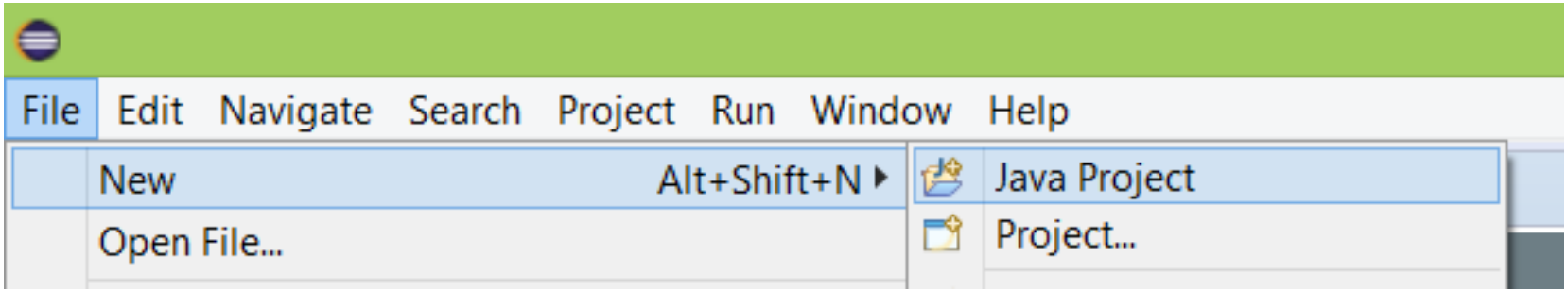
Instanciação

- Instância = objeto da classe

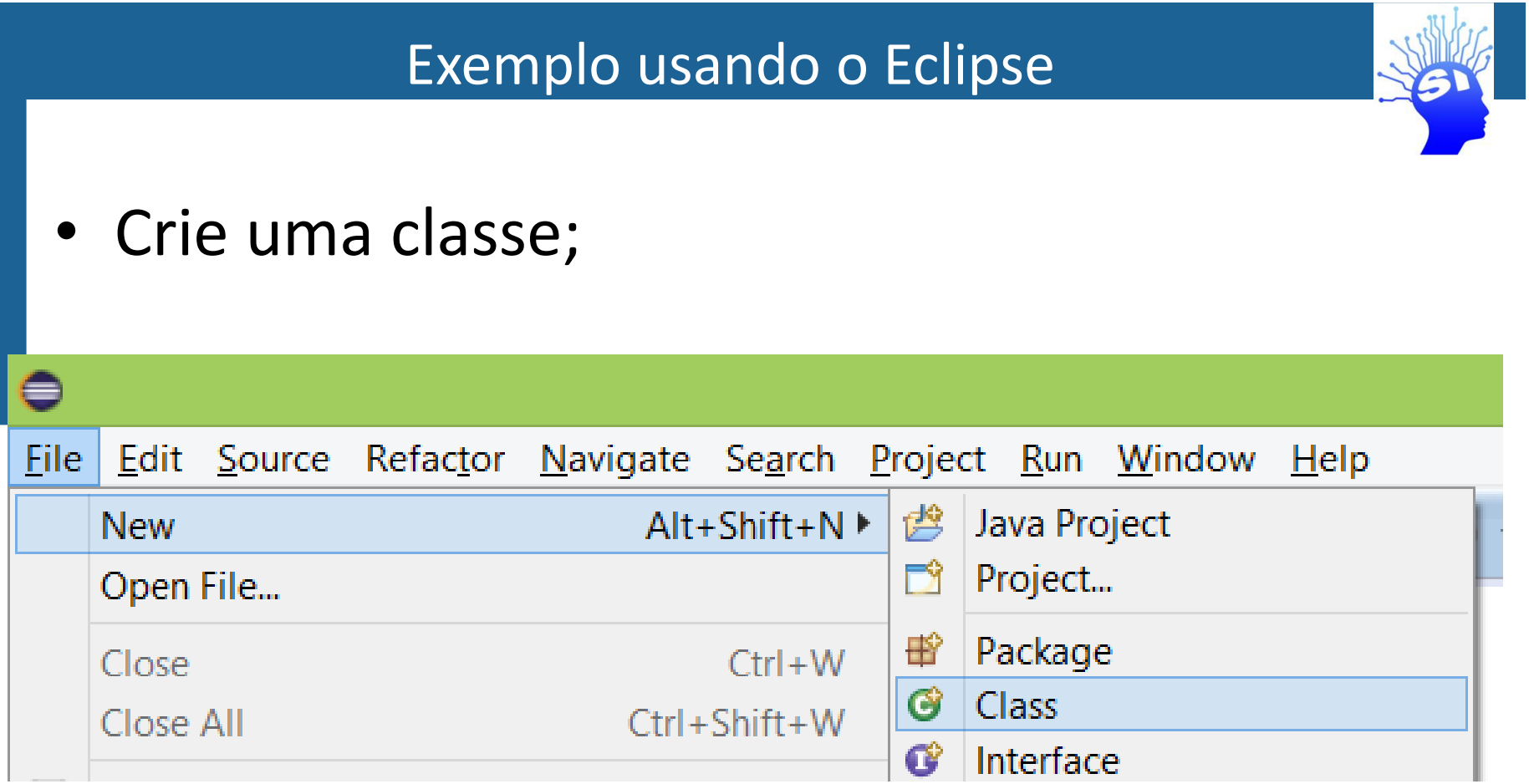


Exemplo usando o Eclipse

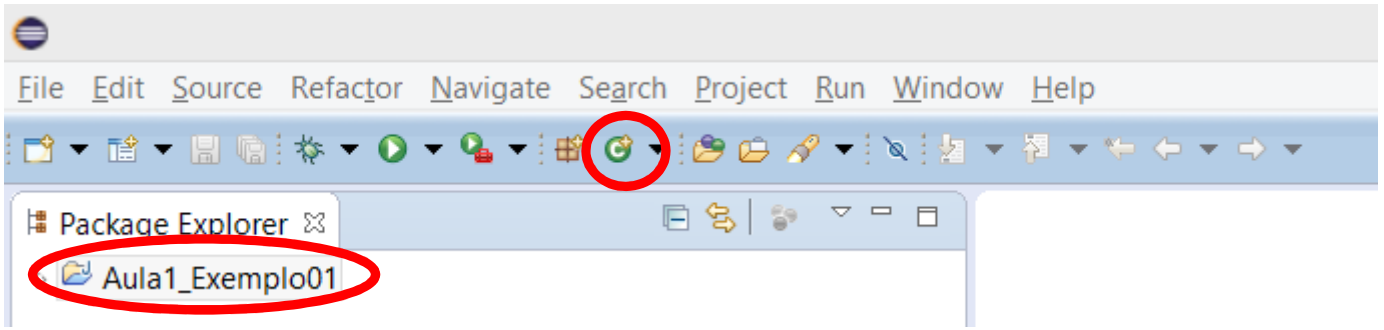
- Abra o Eclipse e clique em File\New\Java Project
- Escolha um nome para seu primeiro projeto;
- Finish;



Exemplo usando o Eclipse



- Crie uma classe;



Exemplo usando o Eclipse

```
 2 public class Aula1Exemplo01Animal {
 3
 4     String especie;
 5     String localizacao;
 6     int idade;
 7
 8     public void nasce() {
 9         System.out.println("Nasce");
10     };
11
12     public void morre() {
13         System.out.println("Morre");
14     };
15 }
```

Definição da Classe

Variáveis / Atributos

Métodos

UEMG FRUTAL

Exemplo usando o Eclipse

```
public class Aula1Exemplo01Humano {

    String nome;
    int idade;
    String cpf;

    public void nasce(){
        System.out.println("Nasce");
    }

    public void estuda(){
        System.out.println("Estuda");
    }
}
```

UEMG FRUTAL

Declaração de Classes

[<**modificadores**> **class** < nome da classe >

[extends <nome da superclasse>**]**

[implements <nome da interface1>**,**

<nome da interface2>**, ...]**

{

// **declarações dos atributos e métodos**

}

Modificadores de Classes

- O modificador **abstract** indica que uma classe possui um ou mais métodos abstratos, isto é, que são declarados mas não são implementados por ela (apenas pelas subclasses)

- O modificador **final** indica que a classe não admite subclasses; (Não admite sobrescrito);

Modificadores de Classes

- O modificador **public** indica que a classe pode ser utilizada (i.e., instanciada ou estendida) por qualquer objeto/classe que esteja no mesmo pacote ou que a importe:

  - classes **public** só podem ser declaradas em um arquivo com o mesmo nome da classe (i.e., <nome_da_classe>.java)

Atributos

- Os atributos são também chamados de **variáveis de instância** por armazenarem dados particulares sobre uma instância
- Atributos podem ter os seguintes modificadores:
  - **public**: livre acesso
  - **protected**: somente métodos de subclasses ou classes do mesmo pacote podem acessar
  - **private**: somente métodos da própria classe podem acessar. Nem mesmo as subclasses têm acesso;
  - **default**: o compilador resolve;

Atributos

- Além dos modificadores que operam sobre a visibilidade dos atributos, existem também:
  - **static**: indica que esse é um atributo da classe, e não da instância (i.e., o valor desse atributo é o mesmo para todas as instâncias da classe).
    - pode ser acessada em todas as instâncias de objetos; Em outra palavras a variável criada será a mesma em todas as instâncias; Quando seu conteúdo é alterado, isso ocorre para todas as demais;
- **final**: determina que o atributo deve receber um valor inicial, que não pode ser modificado no futuro.

Tabela de modificadores de acesso – Mais usados

	private	default	protected	public
Própria classe	sim	sim	sim	sim
Mesmo pacote	não	sim	sim	sim
Pacotes diferentes (subclasses)	não	não	sim	sim
Pacotes diferentes (s/subclasses)	não	não	não	sim

UEMG FRUTAL

Exercício

- Defina as seguintes classes (public) com os respectivos atributos (default); Para isso defina o pacote “aula1”;
  - Cliente;
    - Nome; *String*;
    - Idade; *int*;
    - CPF; *String*;
    - Endereço; *String*;
    - Profissão; *String*;
    - Plano Contratado; *String*;
    - Mensalidade; *double*;
  - ClienteVIP;
    - Bonus; String;
    - Cesta; String;

UEMG FRUTAL

Métodos

- Método é um comportamento da classe;
  - Ex: Nascer; Crescer; Reproduzir; Morrer;
- Como criamos um método?

<modificador de acesso> <tipo de retorno> <nome do método [Parâmetros]> {//comportamento}

```
public int getIdade(){
    return Idade;
}

public void setIdade(int x){
    Idade = x;
}
```

Exercício

- Continuando o exercício anterior, crie métodos get e set para todos os atributos;
- Acesso public;

Método Construtor

- Método construtor é aquele que inicializa os atributos quando a classe é instanciada;

```
<Tipo de Acesso> <CLASSE>
([Atributos])
{ //atribuições }
```

Método Construtor

```
public Cliente(String nome, int idade, String cpf,
        String endereço, String profissão,
        String planoContratado,double mensalidade) {
    Nome = nome;
    Idade = idade;
    CPF = cpf;
    Endereço = endereço;
    Profissão = profissão;
    PlanoContratado = planoContratado;
    Mensalidade = mensalidade;
}
```

UEMG FRUTAL

Método Construtor - Vazio

```
public Cliente() {

}
```

Exercício

- Continuando o exercício anterior, crie 2 construtores com todos os atributos, um para cliente e outro para clienteVIP; e 2 construtores sem atributos;

Método Main

Usa-se um array de Strings para quando é necessário passar parâmetros para a execução do Main

O método **main** deve ser público para que a máquina virtual Java o execute.

Static porque não precisa criar um objeto para iniciar a execução

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

    //Código a ser executado

}
```

Tudo que estiver dentro das chaves**{}** do método será executado pela JVM.

UEMG FRUTAL

Criar instância de Classe – Criação de objeto

<Nome da Classe> <Nome do objeto> =
new <Nome da Classe[Parâmetros]>;

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

    Cliente cli = new Cliente();

    }
```

Manipular objeto

```
1  package Empresa;
2
3  import java.util.Scanner;
4
5  public class Principal {
6
7      public static void main(String[] args) {
8
9          Scanner ler = new Scanner(System.in);
10
11         Cliente cli = new Cliente();
12
13         System.out.println("Digite o nome do cliente:" );
14         cli.Nome = ler.nextLine();
15
16         System.out.println(cli.Nome);
17
18     }
19 }
```

Import do Sanner

Instância de Scanner

Instância da Classe Cliente

Atribuição de conteúdo

Consulta de conteúdo

Saída no Console

Informações Adicionais – Conversão de double e int para String

```
System.out.println("Digite a Mensalidade do cliente:" );
cli.Mensalidade =  Double.valueOf(ler.nextLine());

System.out.println("Digite a idade do cliente:" );
cli.Idade =  Integer.parseInt(ler.nextLine());
```

Exercício

- Continuando o exercício anterior, faça um programa que solicite os dados do cliente e os atribua ao objeto cliente;
- Em seguida mostre os dados do cliente (a partir do objeto cliente);

UEMG FRUTAL

Vinicius Fernandes Caridá

vfcarida@gmail.com