Programação Orientada a Objetos

Instituto Politécnico do Cávado e do Ave

Escola Técnica Superior Profissional

CTeSP – Tecnologia e Inovação Informática

Ano Letivo: 2021/2022

DOCENTE: EDUARDO PEIXOTO

AGENDA

ORGANIZAÇÃO DA APRESENTAÇÃO

1. Objetivos
2. Programação estruturada
3. Programação orientada a objetos
4. Paradigmas da orientação a objetos
5. Revisão C#
6. Exercícios Práticos

OBJETIVOS

No final da aula, os alunos identificar as principais diferenças entre a:

- **Programação estruturada**
- **Programação orientada a objetos**

Programação estruturada

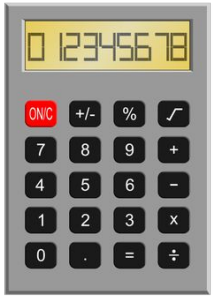
Forma de programação onde todos os programas são desenvolvidos sobre três estruturas:

- Sequências;
- Decisão;
- Iteração (Repetição).

A programação orientada a objetos a criação de estruturas simples nos programas, utilizando sub-rotinas e as funções. Foi a forma dominante no desenvolvimento de software anterior à programação orientada a objetos.

Programação estruturada vs POO

O que é um objeto



Programação orientada a objetos

- A primeira coisa que devemos ter em conta é que, tudo o nosso cotidiano é um objeto!!!!
- Cada objeto está inserido num contexto, que se chama de domínio.

Objeto

Domínio

Programação orientada a objetos

Considerando a ferramenta abaixo como um objeto, podemos identificar algumas caraterísticas (atributos):

- Cor;
- Tamanho;
- Material;
- Fabricante:
- Tipo
- Etc.

Objeto chave (ferramenta)

Objeto Pessoa (Outro exemplo)

Podemos dizer que uma pessoa é um objeto porque toda a pessoa possui atributos.

- Cor de cabelo;
- Tipo de cabelo;
- Cor de olhos;
- Formatos dos olhos;
- Idade;
- Etc.

Classe(1/2)

- São as especificações para os objetos;
- Representam um conjunto de objetos que partilham as características e comportamentos comuns.
  - A características de um objeto são os **atributos** ou **propriedades**.
  - O comportamento de um objeto é definido por **métodos**.

Todo o carro tem em comum:

Características:

- cor;
- pneus;
- Direção.

Comportamento:

- Conduzir;
- Travar.

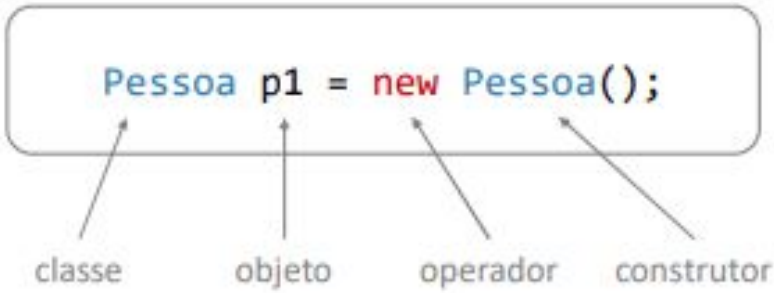
Classe(2/2)

- Definição:

```
class NomeClasse
{
        < atributos… >
        < métodos… >
}
```

Método

- Definição:



- Utilização: p1.**nome** = “Luís Santos”;
  console.Readline(p1.**idade**());

Método construtor (1/2)

- Método especial apenas utilizado na criação de objetos de uma dada classe.
- Para cada classe é automaticamente associado um construtor com o mesmo nome da classe.
- É possível criar tantos construtores quantos os que se achar necessários.
  - regra 1: todos os construtores possuem o mesmo nome, ou seja, o da classe.
  - regra 2: podem diferir quanto ao número e tipo de parâmetros

Método construtor (2/2)

- Sobrecarga do método construtor.

```
Pessoa() {
        nome = "";
        idade = 0;
}
```

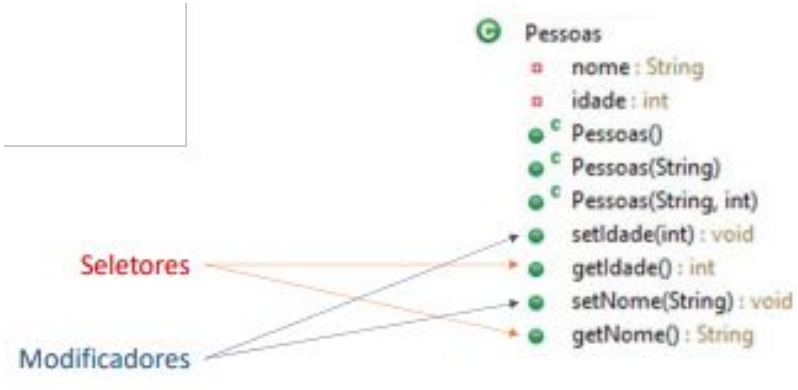
```
Pessoa(String n) {
        nome = n;
        idade = 0;
}

Pessoa(String n, int i) {
        nome = n;
        idade = i;
}
```

Visibilidade / acesso

- Conjunto de palavras chave que controlam a visibilidade e o tipo de acesso a cada um dos membros da classe.
  - **public:** visível em qualquer classe.
  - **private:** o membro só é visível na classe em que foi declarado.
  - **protected:** visível na classe onde foi declarado, qualquer classe do mesmo package em classes derivadas ou herdadas desta.
  - **Internal:** o acesso é limitado apenas ao assembly atual.
  - **Protected Internal:** o acesso é limitado ao assembly atual e aos tipos derivados da classe que contém o modificador

Encapsulamento

- Técnica que permite esconder detalhes desnecessários de um objeto.
- Uma propriedade parece um atributo mas comporta-se como um método.



Seletores e Modificadores

- Seletores / getters
  - método que devolve o valor atual de um atributo

```
public String getNome() {
        return nome;
}
public int getIdade() {
        return idade;
}
```

- Modificadores / setters
  - método que altera o valor atual de um atributo

```
public void setNome(String n) {
        nome = n;
}
public void setIdade(int i) {
        idade = i;
}
```

Diagrama de classes

Nome da classe	Medicamento
Atributos	+ codigo : int + nome : String + descrição: String + preco : float
Métodos	+ inserir() : void + apagar() : void + alterar() : void + verInfo() : void

Notação	Visibilidade
+	public
-	private
#	protected
~	package visibility

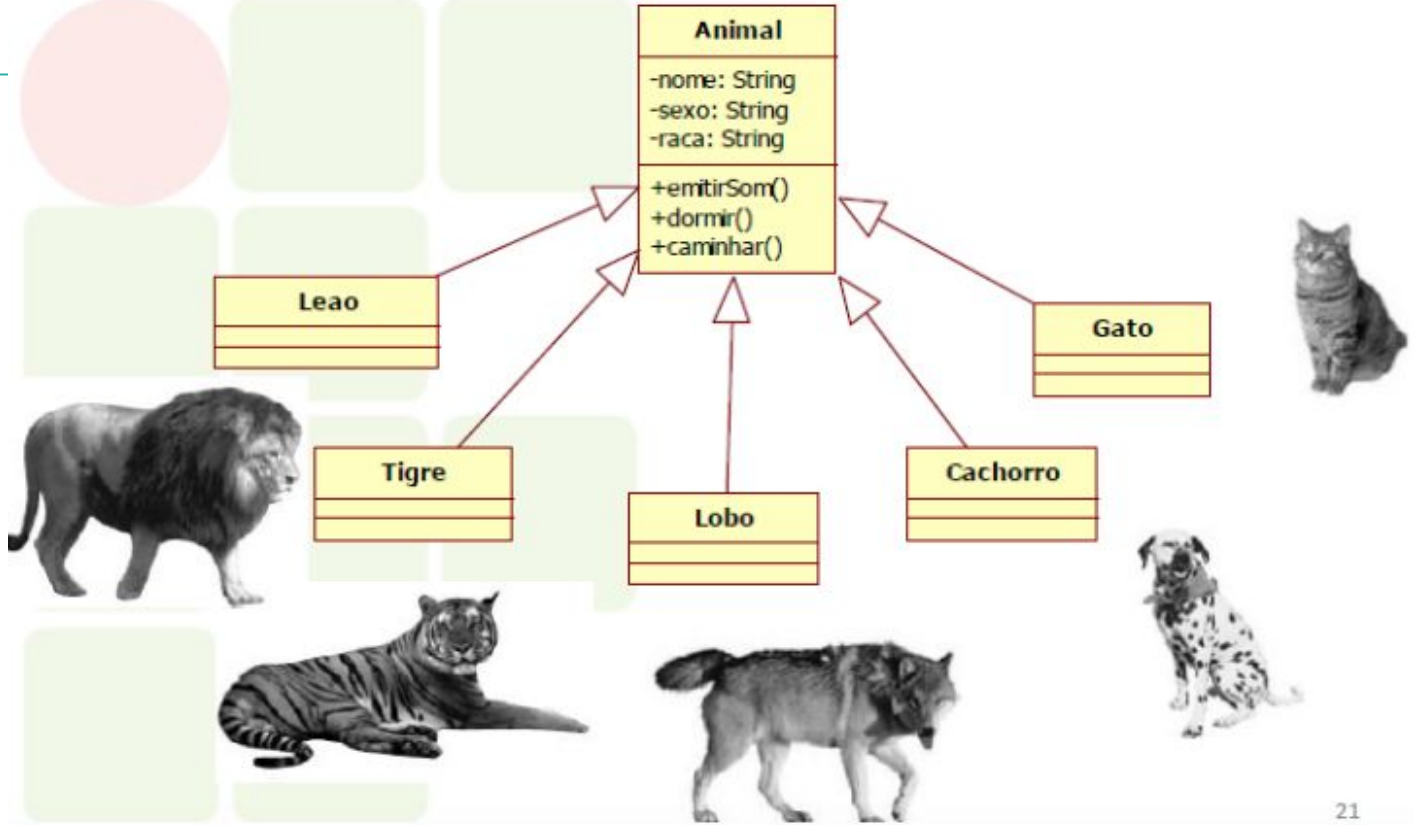
Sistema modular

- Sistema modular é o processo de dividir um todo em partes bem definidas, que podem ser construídas e examinadas separadamente.
- Essas partes interagem entre si, fazendo com que o sistema funcione de forma adequada
- Particionar um programa em componentes individuais, pode reduzir a complexidade.

Herança (1/2)

- Herança é o mecanismo para expressar a similaridade entre Classes, simplificando a definição de classes iguais que já foram definidas.
- O que um leão, um tigre, um gato, um lobo e um cão têm em comum?
- Como é que estão relacionados?

Herança (2/2)

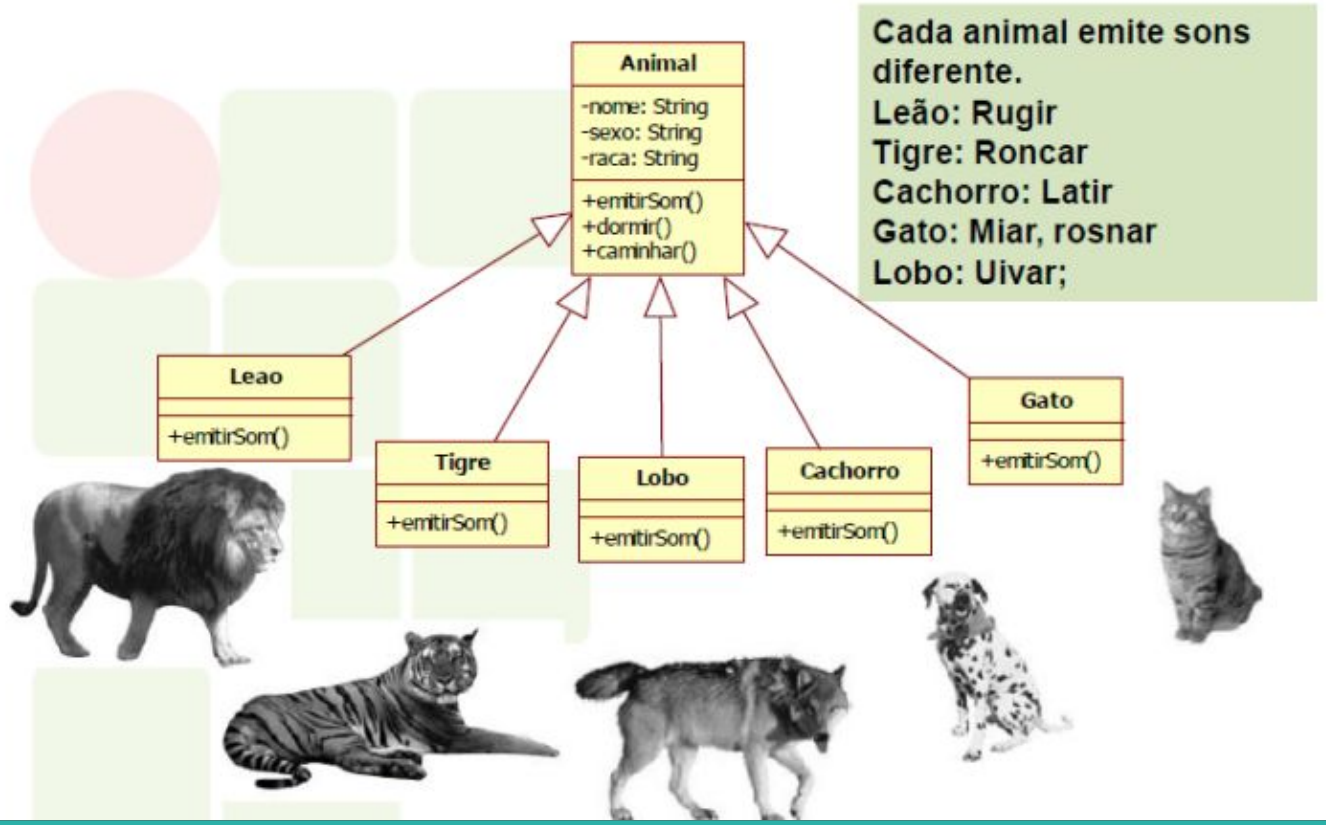


21

Polimorfismo (1/2)

- Polimorfismo
  - Poli: varias;
  - Morfos: formas;
- Significa que um objeto pode assumir diferentes formas;
- O conceito de polimorfismo está associado a Herança;
- É caracterizado como o fato de uma operação poder ser implementada de diferentes maneiras pelas classes na hierarquia.

Polimorfismo (2/2)



Revisão de C# - Sintaxe (1/3)

```
//Bibliotecas utilizadas
(...)

//Namespace do projeto
namespace InserirPessoas {

    //Classe
    public class Pessoa{
        (...)
    }
```

Revisão de C# - Classe (2/3)

```
//Classe
public class Pessoa{
        //variáveis
        private string _nome;
        (...)

        //Construtor sem parâmetros
        public Pessoa(){
                (...)
        }

        //Construtor com parâmetros
        public Pessoa(string nome, ...){
                this.nome = nome;
        }
}
```

Revisão de C# - Encapsulamento (3/3)

```
//Modificadores
public string Nome{
        get { return _nome; }
        set { _nome = value; }
}
```

Resumo

- Objeto
  - Qualquer entidade possui características e comportamentos
- Classe
  - Descreve um tipo de objeto
  - Define atributos e métodos
- Atributo
  - Define características do objeto
- Método
  - Operações que o objeto pode realizar

Exercícios (1/2)

- Implemente a classe:

Exercícios (2/2)

- Desenvolva duas calculadora no windows forms / console no C# com as seguintes características:
  - Programação estruturada;
  - Programação orientada a objetos.

Questões