

Aula 1 - Introdução à Programação Orientada a Objetos

Pedro Lamkowski

LTIA

Dead Battery Studio

Motivação

- Insatisfação com o ensino do assunto na universidade.
- Pouca exploração das oportunidades de aprendizado em desenvolvimento de jogos.
- Incentivar a proatividade no aprendizado.

Background

- Programação procedural -
Conjunto de funções
- Programas complexos = Muitas
funções.
- Repetição de código e
dependência entre funções →



Código spaghetti

Porque orientada a objeto?

Conceitos de POO

Abstração

- Tentativa de representar objetos do mundo real como parte do programa.
- A funcionalidade interna do objeto não é relevante, somente a sua utilização.
- Sua identidade deve ser significativa.

Encapsulamento

- Funções e variáveis que são relacionados a uma funcionalidade são agrupados.
- Independência = Reusabilidade.
- Níveis de acesso.

Herança

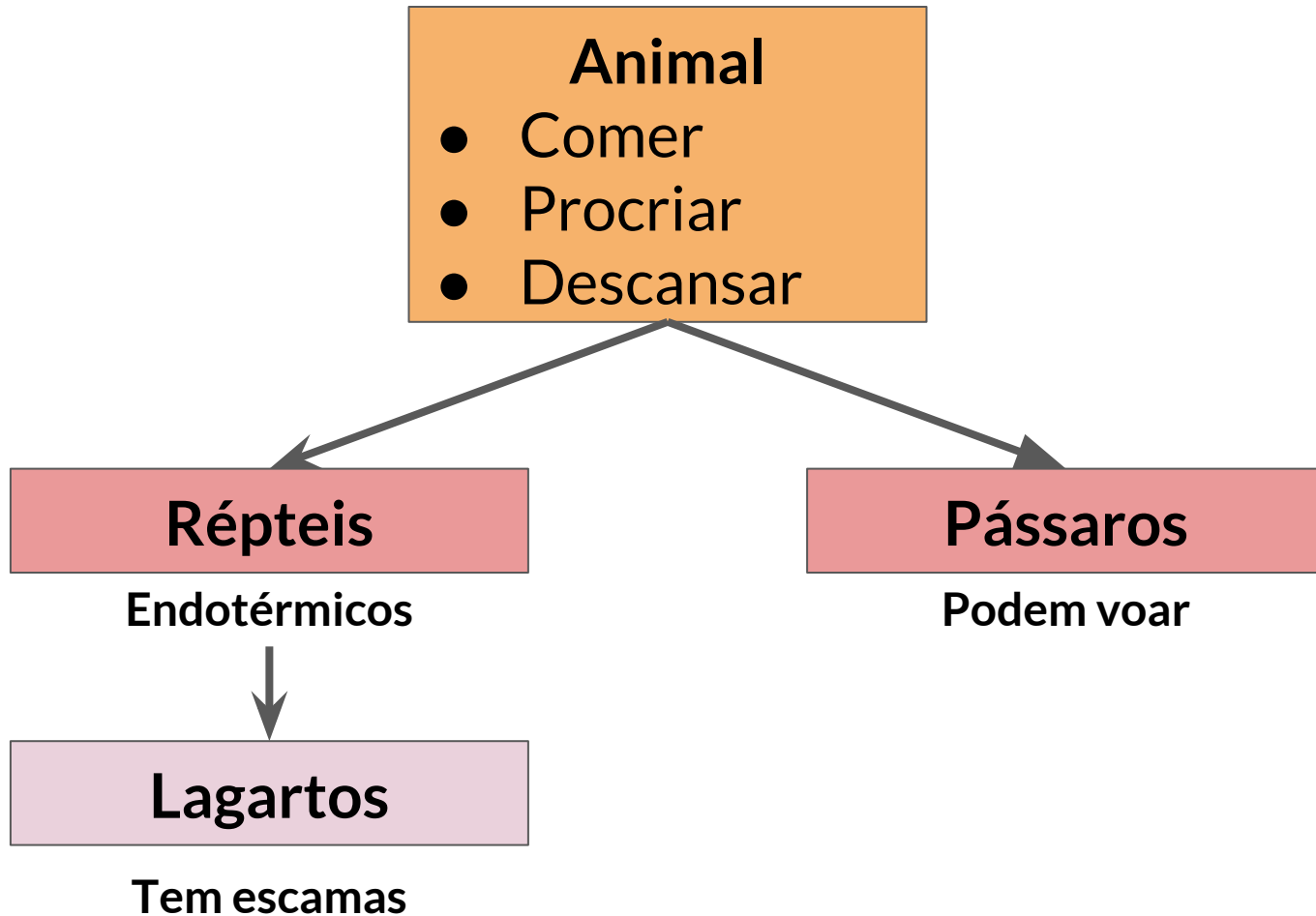
- Objetos com características comuns, entretanto com detalhes diferentes.
- Filhos herdam características de um Pai.
- Conceito de superclasses e subclasses.

Répteis

- Comer
- Procriar
- Descansar

Pássaros

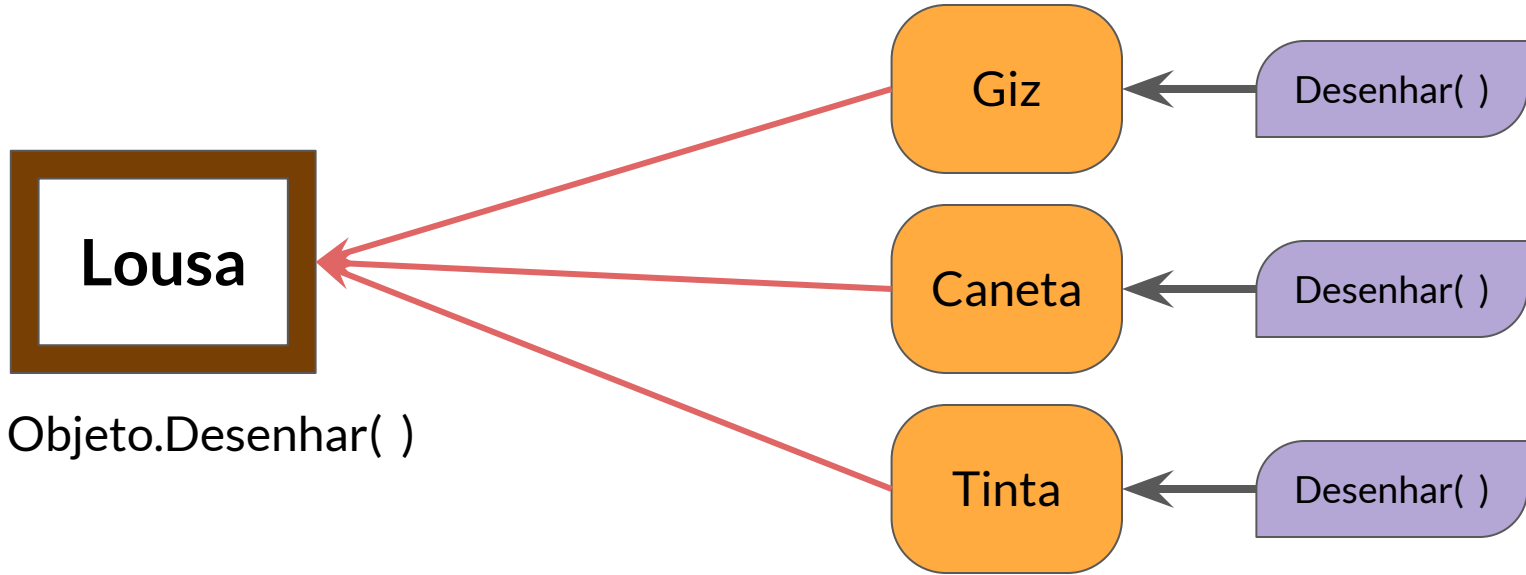
- Comer
- Procriar
- Descansar



Polimorfismo

Diferentes formas (diferentes objetos), que possuem algo que permite que sua funcionalidade não seja afetada por essa diferença.

Exemplo: Lousa.



- Objeto.Desenhar()

A lousa não precisa saber qual objeto ela está usando para desenhar.

Classes e Objetos

Classes

- Descrição do conceito.
- Formada por dados (**propriedades, ou variáveis**) e comportamentos (**métodos, ou funções**).
- Encapsulamento, Abstração, Herança e Polimorfismo.

```
class Animal{
```

Propriedades,
ou variáveis

```
    string nome;
```

```
    int peso;
```

```
void Comer(){
```

```
    // ... Comer ...
```

```
}
```

```
void Dormir(){
```

```
    // ... Dormir ...
```

```
}
```

```
}
```

Métodos, ou
funções

Paralelo com linguagem estruturada

Em C:

Tipo → `int` x; ← Variável



Em Java:

Classe → `Integer` x; ← Objeto



Objetos

- Quando o conceito toma forma.
- **Uma instanciação de uma classe é um objeto.**
- Uma classe pode ser usada para instanciar múltiplos objetos.

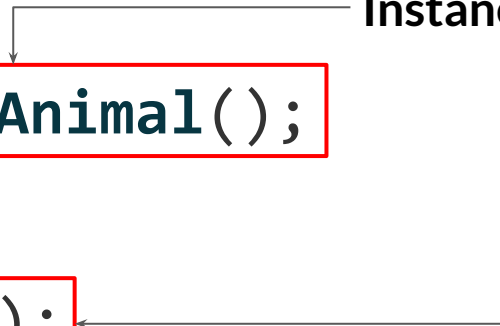
```
class Animal : Ser_Vivo {  
    // ...  
}  
// ...
```

Animal `a = new Animal();`

Animal b;

`b = new Animal();`

Instanciação



```
Animal a;
```

```
Animal b = new Animal();
```

Não funciona.
Objeto “a” não foi
instanciado.



```
a.Comer(); Exibir(a.peso);
```

```
b.Comer(); Exibir(b.peso);
```

Por onde um programa começa?

Em .NET (C#), Java , ... , o programa começa em um método Main:

```
using System;
namespace HelloWorld
{
    class Hello
    {
        static void Main()
        {
            Console.WriteLine("Hello World!");
            Console.WriteLine("Press any key to exit.");
            Console.ReadKey();
        }
    }
}
```

Não se preocupe com todas essas palavras, elas serão explicadas na aula 2.

Provavelmente.

Para qualquer linguagem de programação:

- Aconteceu um erro de compilação/interpretação/runtime?
Leia o erro e procure entender o que ele diz.
- Depuração com pato de borracha (Rubber duck debugging).
- Stack Overflow.