Introdução à programação de jogos em C#

CSJ ACADEMY

Aula 02 – Classes

Classes: Uma forma de agrupar métodos e variáveis juntas

Classes: Uma forma de agrupar métodos e variáveis juntas

```
class Player {

}
```

Classes: Uma forma de agrupar métodos e variáveis juntas

```
class Player {
        //métodos (ações) e variáveis (atributos) pertencentes à classe
        Player vão aqui
}
```

Mamíferos

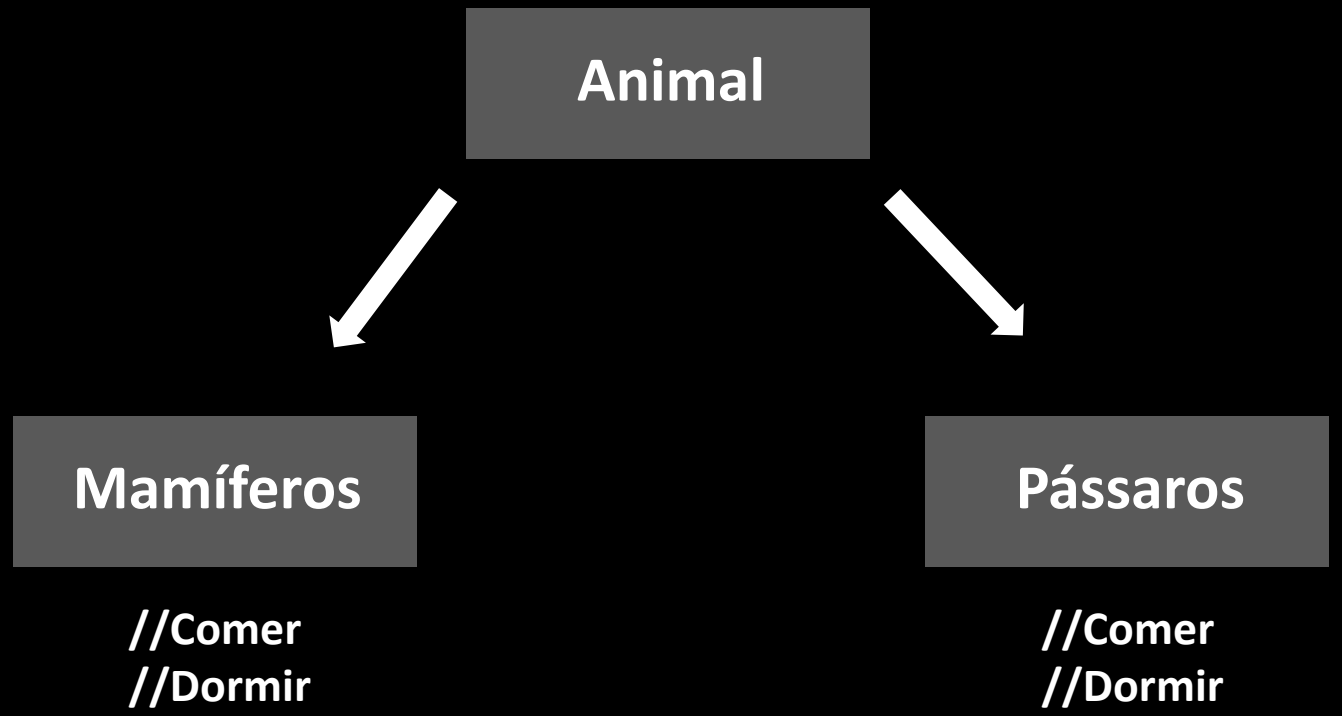
Pássaros

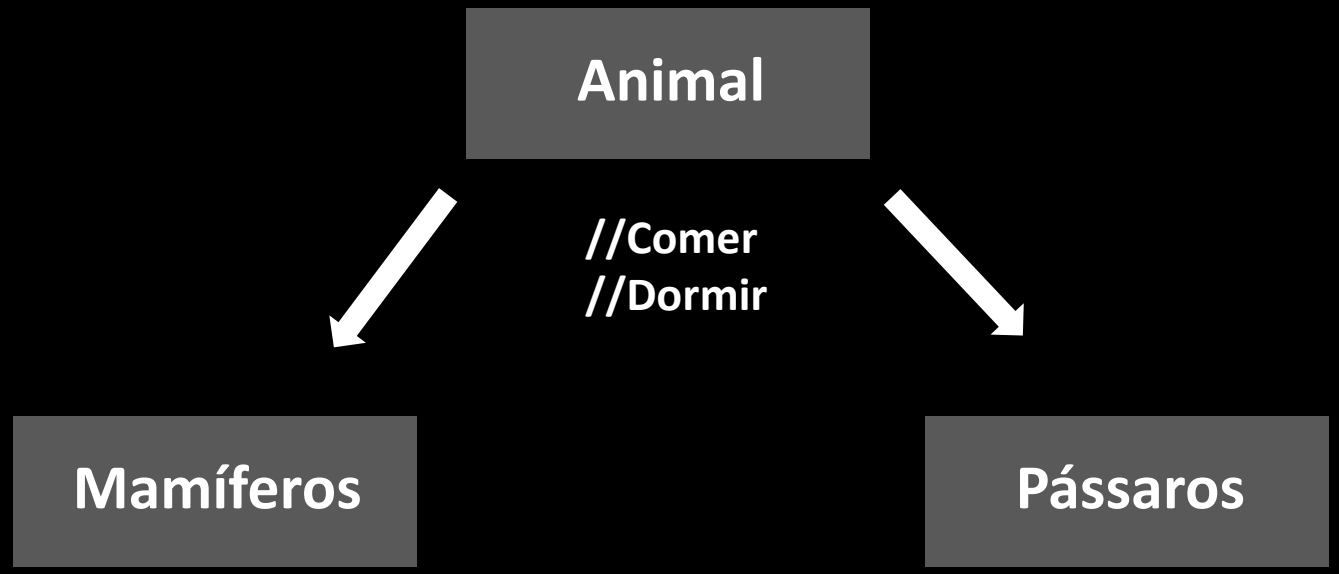
Mamíferos

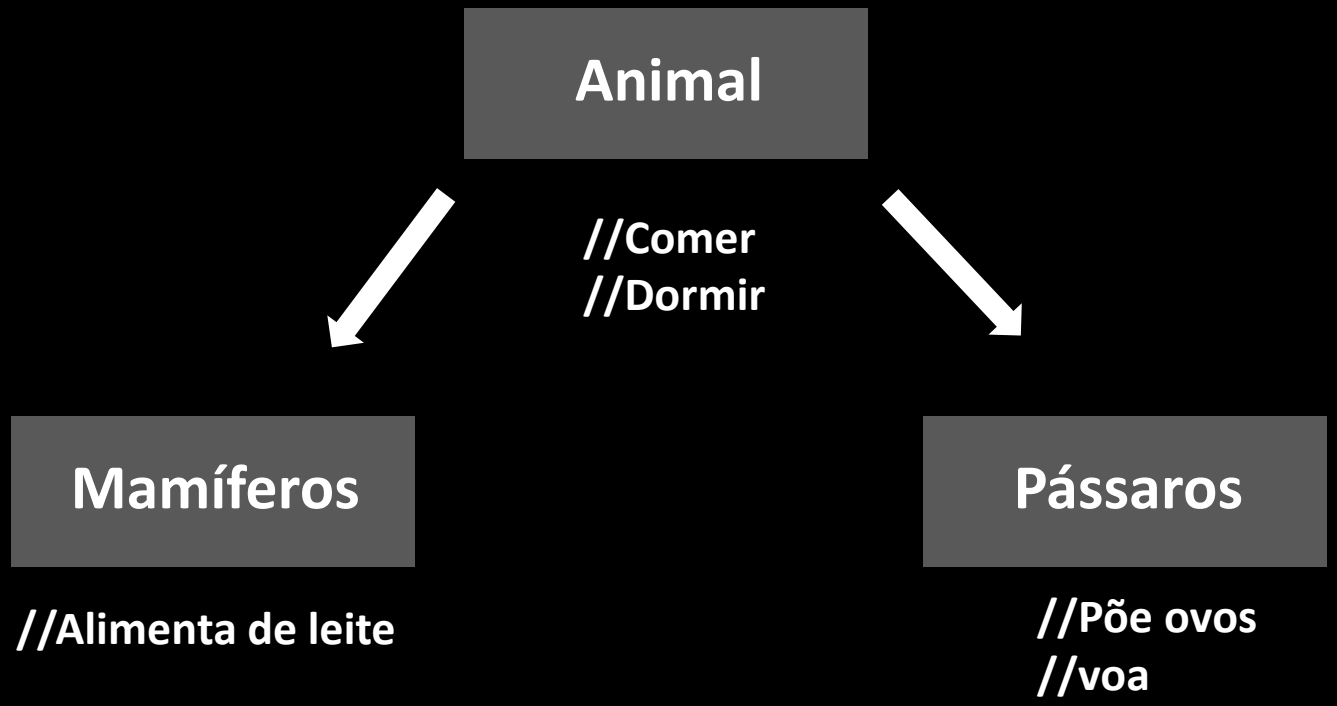
//Comer
//Dormir

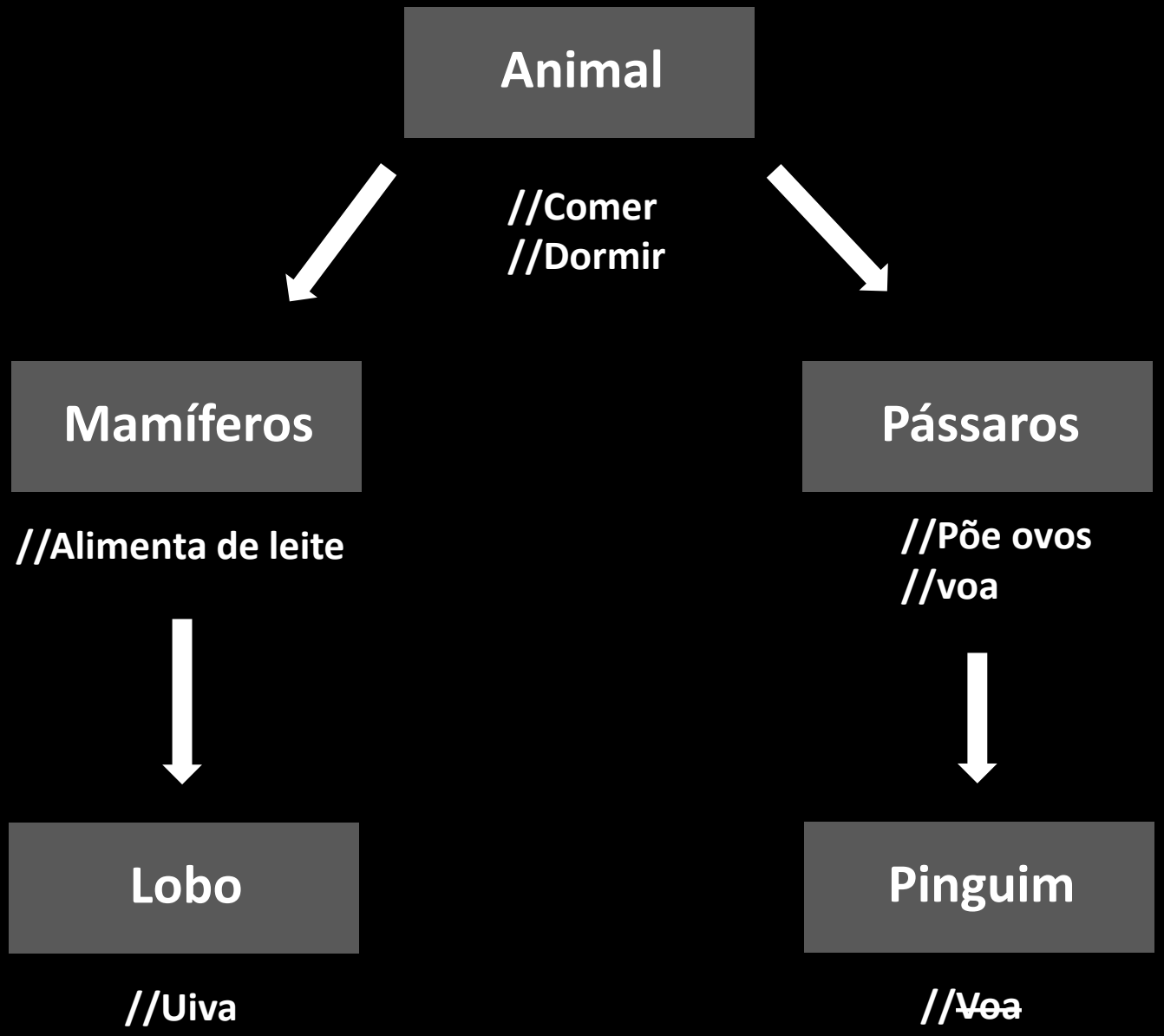
Pássaros

//Comer
//Dormir


Animal
Mamíferos
//Comer
//Dormir
Pássaros
//Comer
//Dormir


Animal
//Comer
//Dormir
Mamíferos
Pássaros


Animal
//Comer
//Dormir
Mamíferos
//Alimenta de leite
Pássaros
//Põe ovos
//voa


Animal
//Comer
//Dormir
Mamíferos
//Alimenta de leite
Pássaros
//Põe ovos
//voa
Lobo
//Uiva
Pinguim
//~~Voa~~

Andar

Nadar

Andar

Nadar

Voar

Pôr ovos

Andar

Nadar

Voar

Pôr ovos

Comer

Dormir

Andar

Nadar

Voar

Pôr ovos

Comer

Dormir

Andar

Nadar

Voar

Pôr ovos

Comer

Dormir

Elefante

Andar

Nadar

Comer

Dormir

Pelicano

Andar

Nadar

Comer

Dormir

Voar

Pôr ovos

Classes: Uma forma de agrupar métodos e variáveis juntas

- Manter o código flexível e minimizar repetições;
- Uso de herança e composição são as metodologias que nos permitem fazer isso.

Classes do MonoBehaviour

Esse tipo de classe herda da classe MonoBehaviour da Unity.

```
//Exemplo
void Start() {

}
```

Classes do MonoBehaviour

Esse tipo de classe herda da classe MonoBehaviour da Unity.

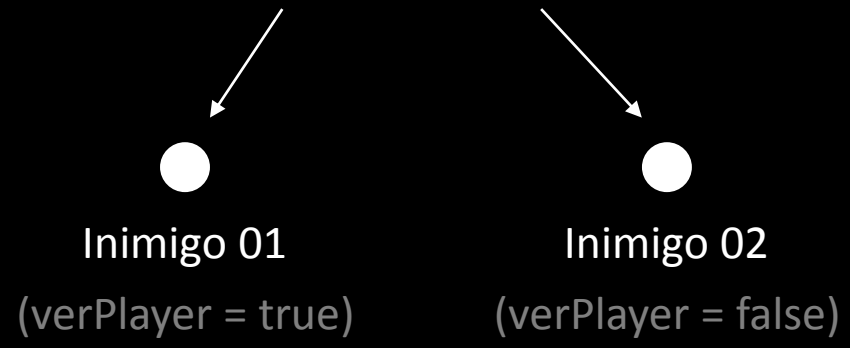
```
//Chamado uma vez ao início do jogo
void Start() {

}

//Chamado a cada frame
void Update() {

}
```

```
class Inimigo : MonoBehaviour {

    void Update() {

    }

}
```

```
class Inimigo : MonoBehaviour {

    void Update() {

        bool verPlayer = false;
        //Mudar o valor de “verPlayer” para true quando o Player estiver a uma certa distância

        if(verPlayer) {

            //ataque o player!

        }

    }

}
```

```
class Inimigo : MonoBehaviour {

    void Update() {

        bool verPlayer = false;
        //Mudar o valor de “verPlayer” para true quando o Player estiver a uma certa distância

        if(verPlayer) {

            //ataque o player!

        }

    }

}
```

Ambos possuem uma cópia própria – **instância** – da classe “inimigo”



Classes: Uma forma de agrupar métodos e variáveis juntas

- Manter o código flexível e minimizar repetições;
- Uso de herança e composição são as metodologias que nos permitem fazer isso.

Andar

Nadar

Voar

Pôr ovos

Comer

Dormir

Elefante

Andar

Nadar

Comer

Dormir

Pelicano

Andar

Nadar

Comer

Dormir

Voar

Pôr ovos

CSJ ACADEMY

FIM