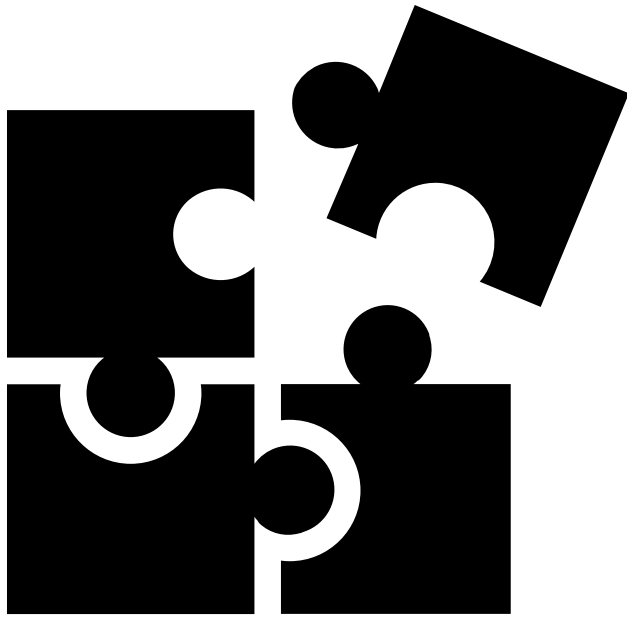


Orientação a Objetos

PROFESSOR: EDUARDO HABIB BECHELANE MAIA
HABIB@CEFETMG.BR



Conceitos

Definição: o termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam ***estrutura de dados*** e ***um conjunto de operações*** que manipulam estes dados.

Conceitos



Projeto OO é o método que produz arquiteturas de *software* baseadas nos **objetos** que o sistema manipula;



A propriedade básica do método:

Evite perguntar: *O que o sistema faz?*

Melhor perguntar: *Sobre o que o sistema faz o que?*



Olhar primeiro para o **dado** é a regra que favorece a **reusabilidade**;



Descrição de objetos deve ser: completa, precisa, não ambígua e independente de representação física.

Conceitos



O comportamento dos objetos é definido pelo seu Tipo Abstrato de Dados (TAD);



A descrição de um **TAD** deve compreender:

Interface do TAD;
Comportamento do TAD;



Um tipo abstrato de dados é uma **classe** de estrutura de dados descrita por uma interface externa:

Lista de serviços disponíveis;
Propriedades destes serviços.

Conceitos



A representação das estruturas de dados do TAD fica completamente **encapsulada**;



A representação das estruturas de dados do TAD **não** faz parte de sua **definição**;

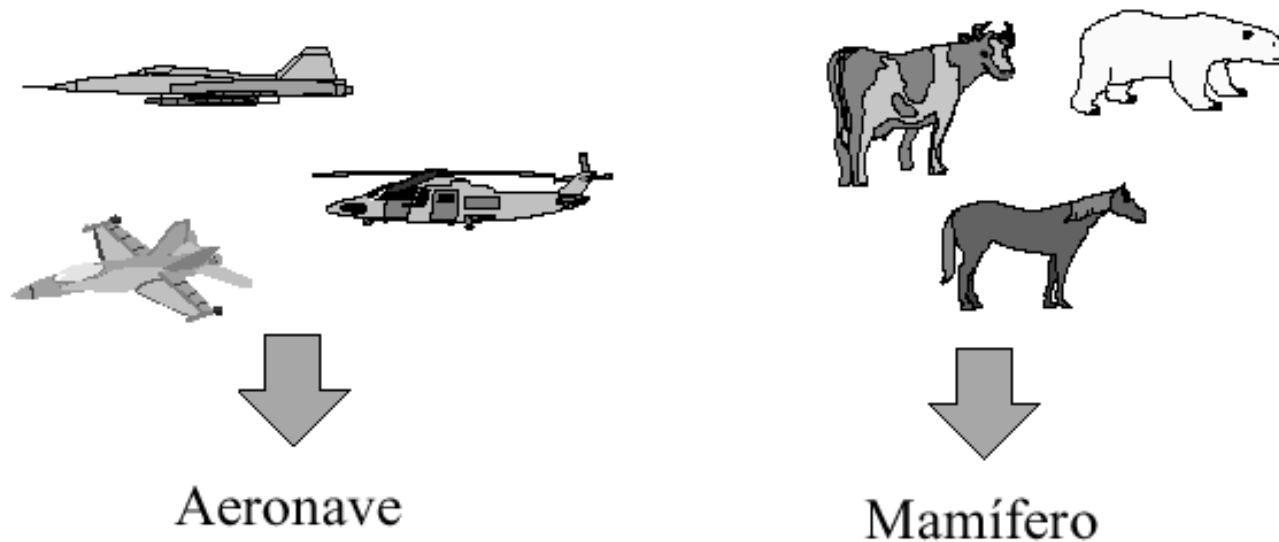


O adjetivo **abstrato** de um tipo abstrato de dados enfatiza o fato de as estruturas de dados que representam o tipo **não** fazerem parte da definição, isto é, da **interface** do tipo.

Abstração

Focalizar o essencial, ignorar propriedades acidentais

A abstração deve ser sempre feita com algum objetivo, porque este determina o que é e o que não é importante.



Abstração



Técnica para lidar com a complexidade de um problema.



Destaca os aspectos importantes do objeto real abstraído, segundo perspectiva do observador.



Ignora os detalhes não relevantes para o observador.



Exemplos de uso da abstração no dia a dia:



Dirigir um carro.



Usar um tocador de DVD ou outro dispositivo eletrônico.



Tomar um voo num aeroporto.



Retirar dinheiro num caixa eletrônico.

Abstração

-

Exemplos

Abstração (Exemplo)

Quem usa as funções `sqrt`, `pow`, etc, da classe `Math`, interessa-se por:

Saber quantos
são os
parâmetros para
as funções
chamadas e em
que ordem são
passados

Saber quais os
tipos dos
parâmetros

Saber os tipos
dos dados que as
funções
retornam

Saber o
significado desse
dato retornado

Classes



Classe é um conceito de *software* que descreve a implementação de um TAD;



Uma classe define:

A estrutura de dados que representa o TAD;
A implementação das operações, métodos, sobre esta estrutura;
Uma interface explícita;



Classe é apenas um molde para criação de TAD;



Uma classe é a representação de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

Classes

Encapsulamento;

Herança;

Abstração;

Proteção de dados;

Polimorfismo;

Hierarquia.

Conceitos OO - Classes



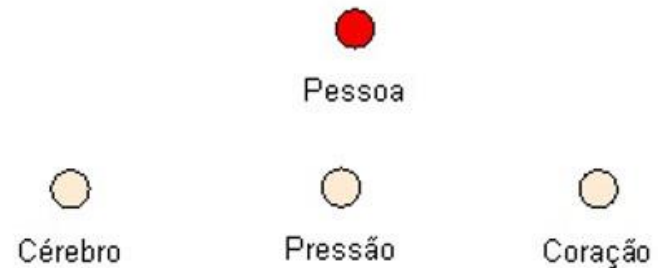
Abstração é correlacionada aos objetivos de quem se abstrai;



A abstração do TAD Pessoa depende de seu uso no sistema;

Conceitos OO - Classes

Pessoa sob a perspectiva de um médico.



Pessoa sob a perspectiva de cadastro de dados:



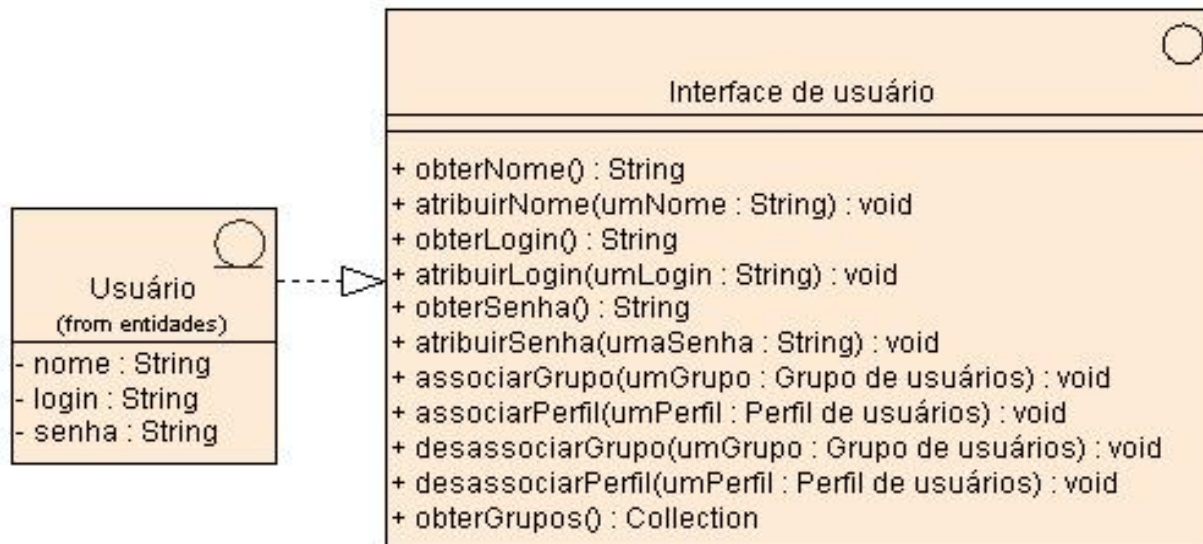
Conceitos OO - Classes



Encapsular consiste em incluir, proteger em uma cápsula, classe;



Encapsular é ocultar do usuário o funcionamento interno de uma classe.



Objetos

Objetos são conceitos de *software* que modelam entidades da aplicação;

Objetos são abstrações de dados;

Objetos tem estado (estrutura interna);

Objetos são manipulados só pelas operações;

Objetos são instâncias de **classes**;

Uma classe é a representação de um conjunto de **objetos** que compartilham os mesmos atributos, operações, relacionamentos e semântica.

Classes Exemplo

Arquivo Pessoa.hpp

```
#ifndef __PESSOA_HPP
#define __PESSOA_HPP
#include <string>
using namespace std;
class Pessoa {
private:
    float peso, altura; //variável
    char sexo;
    string nome;

public:
    void setNome(string nome);
    string getNome();
    void setPeso(float p);
    float getPeso();
    void setSexo(char sexo);
    char getSexo();
    void setAltura(float altura);
    float getAltura();
    void imprimir();
};
#endif
```


Classes Exemplo

Arquivo Pessoa.cpp

```
#include<iostream> //biblioteca
#include "Pessoa.hpp"
using namespace std;
void Pessoa::setNome(string
nome){
    this->nome=nome;
}
string Pessoa::getNome(){
    return nome;
}
void Pessoa::setPeso(float p){
    this->peso=p;
}
float Pessoa::getPeso(){
    return peso;
}
void Pessoa::setSexo(char sexo){
    this->sexo=sexo;
}
char Pessoa::getSexo(){
    return sexo;
}
}

void Pessoa::setAltura(float
altura){
    this->altura=altura;
}
float Pessoa::getAltura(){
    return altura;
}
void Pessoa::imprimir(){
    //Imprime os dados da Pessoa
    cout<< "-----\n";
    cout<< "Nome: " << nome
<< endl;
    cout<< "sexo: " << sexo <<
endl;
    cout<< "Altura: " << altura
<< endl;
    cout<< "Peso: " << peso <<
endl;
}
```

Classes – Exemplo

Arquivo
main.cpp

```
#include <iostream>
#include "Pessoa.hpp"
int main(){
    Pessoa p,z;
    float peso,altura;
    char s;
    string nome;
    cout<< "Digite o nome: ";
    getline(cin,nome);
    p.setNome(nome);
    cout<< "Digite o sexo: ";
    cin>> s;
    p.setSexo(s);
    cout<< "Digite a altura: ";
    cin>> altura;
    p.setAltura(altura);
    cout<< "Digite o peso: ";
    cin>> peso;
    p.setPeso(peso);
    p.imprimir();
} //fim
```

Compilar

```
g++ Pessoa.hpp Pessoa.cpp main.cpp -o ex
```