Paradigma Orientado a Objetos

Profª Maria Adelina Raupp Sganzerla

Paradigmas de Linguagens de Programação

Ulbra Gravataí – 2016/2

Paradigma Orientado a Objetos

- Foco no problema
  - Identificação dos objetos do mundo real do problema
  - Criação de soluções, onde os objetos mandam mensagens entre si de modo a solucionar o problema
- Primeira representante: Smalltalk, 1972 (puramente orientada a objetos)
- Representantes atuais: C++, Java, C#, ...
- Características principais: abstração, herança e polimorfismo

Abstração

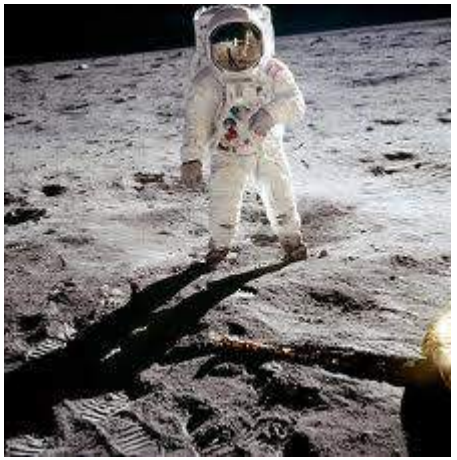
- Omissão de detalhes de representação de dados
  - **Encapsulamento**
- **Classes** são dados abstratos
  - Variáveis + métodos (funções e procedimentos)
- **Objetos** são instâncias de classes
- **Entidades** são variáveis e métodos herdados

Abstração

- Mecanismo utilizado na análise de um domínio;
- Através dela, o individuo observa a realidade e dela abstrai entidades, ações, …, consideradas essenciais para uma aplicação, excluindo todos os aspectos julgados irrelevantes;
- Exemplo:
  - Fotografia por satélite (imagem da realidade), despida de alguns aspectos (por exemplo, cor, movimento)
  - Da foto, pode-se abstrair um **mapa**, que elimina diversas propriedades da foto (detalhes particulares de um edifício ou praça)
  - O mapa pode ser a base para abstrair um **grafo**

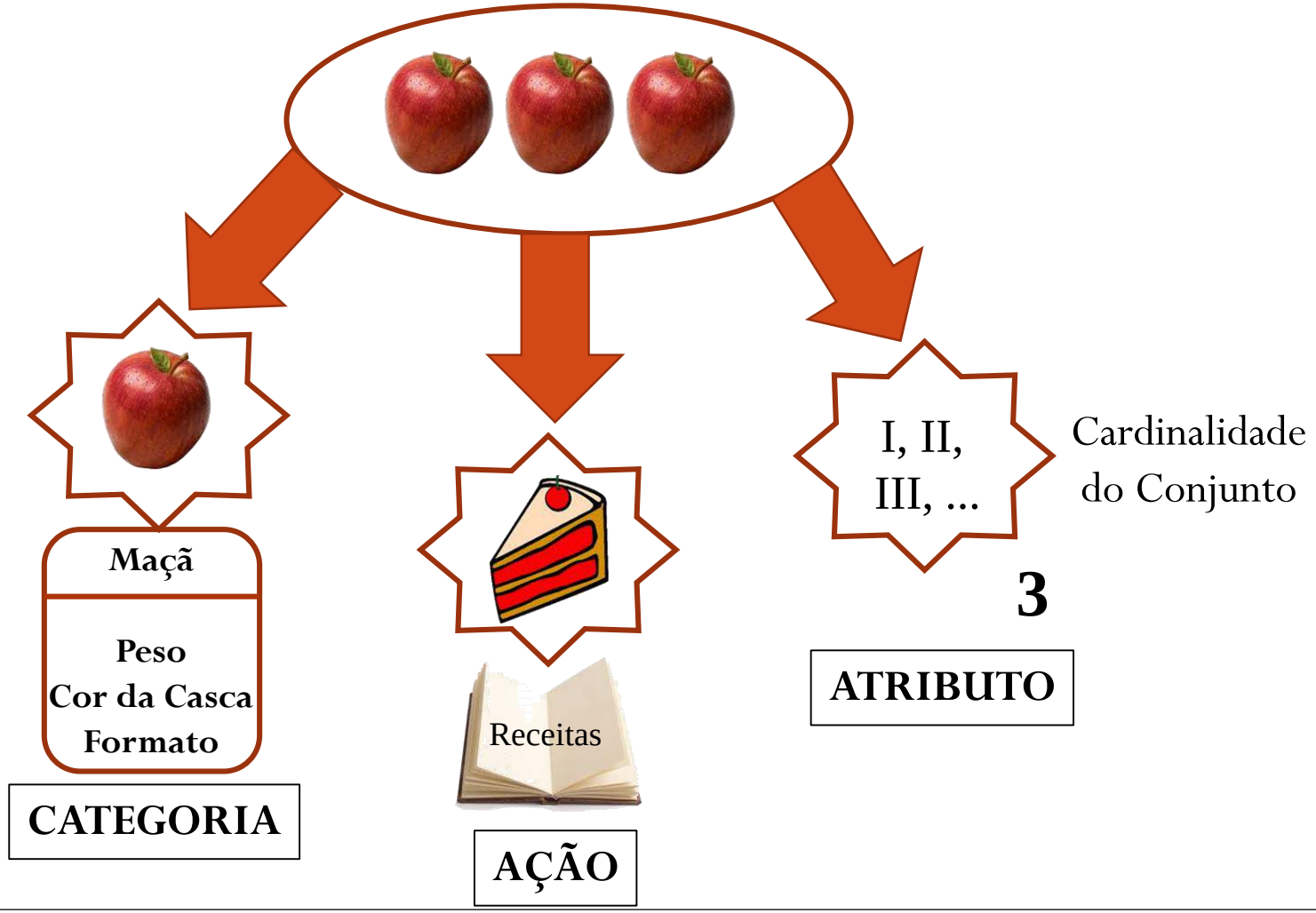
Abstração

- O resultado de uma operação mental de abstração depende não tanto do fenômeno observado, mas do interesse do observador.



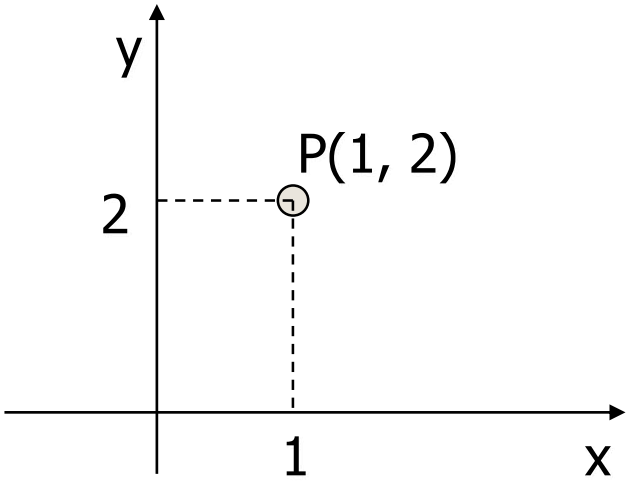
Abstração

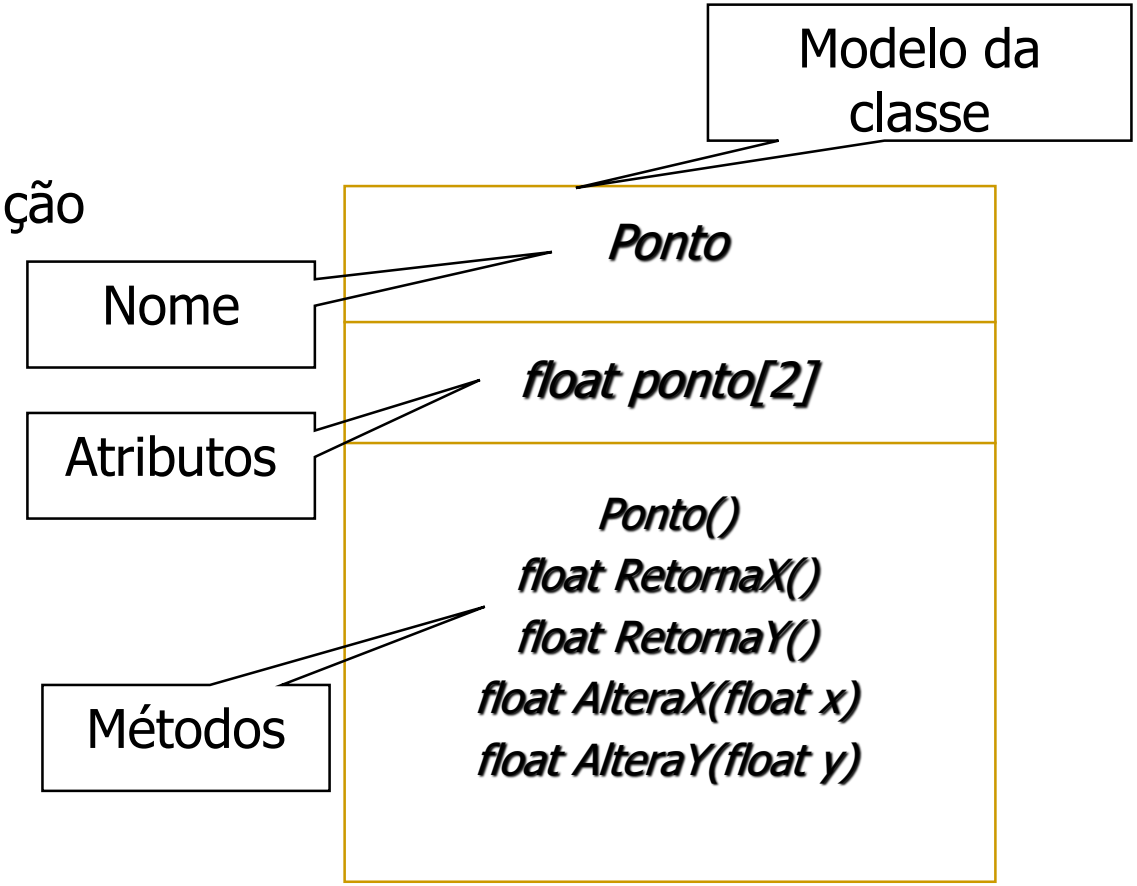
- Diferentes abstrações a partir de um mesmo objeto do mundo real:



Abstração em C++ - Classe: Exemplo

- Exemplo de declaração:
  Criação de uma classe para a definição de ponto com coordenadas x e y





Abstração em C++ Classe

- Forma geral de definição de classe:

```
class <nome_classe> {
        private:
        <declaração variáveis privadas>
        protected:
                <declaração variáveis protegidas>
        public:
        <declaração variáveis públicas>
        <método(s) construtor(es)>
        <demais métodos>
};
```

Constrói o objeto

Mesmo nome da classe

Abstração em C++ - Objeto

- Classes possuem extensão .h
- Os objetos devem ser instanciados em variáveis fora da classe construtora
- Programa que usará a classe tem que fazer seu "include"

Polimorfismo

- “Várias formas”

- Entidades do programa podem pertencer a mais de um tipo
  - Exemplo: <float>:= <int> + <float>
  - Todas as linguagens de programação possuem alguma forma de polimorfismo!

- Propriedade que permite que uma mesma mensagem seja enviada a diferentes objetos
  - Cada objeto executa a operação apropriada à sua classe
  - O objeto que envia a mensagem não precisa conhecer a classe do objeto receptor
    - O objeto receptor irá responder com o método que for apropriado à sua classe

Conceitos – Resumo (1/3)

- **Classe**: Agrupamento de objetos simples que apresentam os mesmos atributos e operações.
  - Exemplo: Indivíduo, caracterizando as pessoas do mundo.
- **Atributo**: Característica particular de uma ocorrência da classe.
  - Exemplo: Indivíduo possui nome, sexo, data de nascimento.
- **Operações**: Lógica contida em uma classe para designar-lhe um comportamento.
  - Exemplo: Cálculo da idade de uma pessoa em uma classe (Indivíduo).
- **Encapsulamento**: Combinação de atributos e operações de uma classe.
  - Exemplo: Atributo: Data de nascimento – Operação: Cálculo da idade.

Conceitos – Resumo (2/3)

- **Herança**: Compartilhamento de atributos e operações de uma classe.
  - Exemplo: Subclasse (Eucalipto) compartilha atributos e operações da Classe (Árvore).
- **Subclasse**: Característica particular de uma classe.
  - Exemplo: Classe (Árvore) -> Subclasses (Ipê, Eucalipto, Jacarandá, etc.).
- **Instância de Classe**: Uma ocorrência específica de uma classe. É o mesmo que objeto.
  - Exemplo: Uma pessoa, uma organização ou um equipamento.
- **Objeto**: Elemento do mundo real (natureza). Sinônimo de instância de classe.
  - Exemplo: Pessoa "Fulano de Tal", Organização "ACM", Equipamento "Extintor".

Conceitos – Resumo (3/3)

- **Mensagem**: Uma solicitação entre objetos para invocar certa operação.
  - Exemplo: Informar a idade da pessoa "Fulano de Tal".
- **Polimorfismo**: Habilidade para usar a mesma mensagem para invocar comportamentos diferentes do objeto.
  - Exemplo: Chamada da operação "Calcular Saldo" de correntista. Invoca as derivações correspondentes para cálculo de saldo de poupança, renda fixa, entre outras.

Migrando do C para o C++ (1/8)

- Considere o seguinte programa em C, que lê o nome e as notas de G1 e G2 de um aluno, imprimindo tais informações e a sua respectiva média:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define TAM 60

float CalculaMedia(float NG1, float NG2){
 float M;
 M = (NG1+(NG2*2))/(float)3;
 return M;
 }

int main(){
  char nome[TAM];
  float G1, G2, Media;
  printf("\nNome:  ");  gets(nome);
  printf("\nG1:  ");  scanf("%f",&G1);
  printf("\nG2:  ");  scanf("%f",&G2);
  Media = CalculaMedia(G1,G2);
  printf("Nome do(a) aluno(a): %s\n", nome);
  printf("Nota G1 do(a) aluno(a): %.2f\n", G1);
  printf("Nota G1 do(a) aluno(a): %.2f\n", G2);
  printf("Media do(a) aluno(a): %.2f\n", Media);
  getch();
}
```

Migrando do C para o C++ (2/8)

- A partir do programa anterior, criaremos uma classe chamada Aluno - que permite o instanciamento de objetos do tipo Aluno – com métodos (funções) para:
  - Construir o objeto
  - Definir valores para o nome e para as notas G1 e G2
  - Calcular e imprimir a média da G1 e da G2

- Neste exemplo, os objetos serão instanciados (criados) e usados no programa (arquivo) main.cpp

Migrando do C para o C++ (3/8)

- A classe possui extensão .h (Aluno.h) e o método construtor possui o mesmo nome da classe

Quando o objeto for instanciado (criado) no programa main.cpp, suas variáveis serão inicializadas vazias.

```
#include <string.h>
#define TAM 60

class Aluno {
      private:
         char Nome[TAM];
         float G1,G2,Media;

      public:
      Aluno(){
        strcpy(Nome," ");
        G1=0.0;
        G2=0.0;
        }
};
```

Migrando do C para o C++ (4/8)

- A classe “Aluno.h” é chamada pelo programa main.cpp
  - Deve-se indicar para o compilador onde buscar a classe utilizada
  - Menu “Ferramentas>>Opções de Compilador” do Devcpp
    - Na guia “Diretório>>C++ Includes” procure o diretório onde se encontra a classe “aluno.h” e adicione a mesma

Migrando do C para o C++ (5/8)

- No programa main.cpp, declara-se a classe “aluno.h” e instancia-se o objeto na variável X

```
#include <conio.h>
#include <stdio.h>
#include “Aluno.h”

using namespace std; //versões de compiladores

int main() {
  Aluno X;
  getch();
 }
```

Migrando do C para o C++ (6/8)

- Voltando para a classe Aluno.h, além do método construtor, iremos definir métodos para a atribuição de valores às variáveis “Nome”, “G1” e “G2”

Migrando do C para o C++ (7/8)

- Altere o programa main.cpp de forma que sejam lidos valores para as variáveis Nome, G1 e G2 e passadas para o objeto através dos respectivos métodos.

Migrando do C para o C++ (8/8)

- Voltando para a classe aluno.h, defina, por fim, o método que calcula a média do aluno.
- Altere o programa main.cpp para calcular a média do aluno.

Atividade – Implementação:

- Verificação de “Aprovado” ou em “G3”
- Verificação se G3, quanto precisa na Prova para ser aprovado
- Leitura das presenças do aluno durante o semestre
- Verificação de Aprovação, contanto com as presenças (75% no mínimo), sendo que o semestre possui 68 horas/aula