Programação Orientada a Objetos

Aula III – Encapsulamento e diagrama de classes

Prof.: Bruno E. G. Gomes
IFRN

Introdução

- Na aula de hoje:
  - Conceito “encapsulamento” e implementação
  - Diagrama de classes
  - Relacionamento entre objetos

Encapsulamento

- O “Encapsulamento” é um dos pontos fundamentais da programação OO.
- Consiste em uma unidade formada por um pacote de operações (métodos) e atributos do objeto
  - Atributos representam o “estado” do objeto;
  - O estado deve ser acessível ou modificado somente através de métodos (interface provida pelo encapsulamento)
- O bom encapsulamento se traduz em exibir apenas o que o usuário do objeto deve conhecer
  - Deve-se ocultar *informações* e *implementação*

Encapsulamento

- O que é normalmente escondido ?
  - Acesso a atributos
  - Acesso a métodos
  - Implementação dos métodos (por padrão)

- Acesso pode ser (por enquanto):
  - ***public*** – todo objeto da classe pode acessar diretamente o atributo ou método
  - ***private*** – atributo ou método pode ser acessado diretamente apenas dentro da classe

Encapsulamento

- Uso de encapsulamento diminui a chance de introdução de erros em seu programa.

- Normalmente, mas nem sempre:
  - *Atributos são privados*
    - Atributos constantes podem ser públicos, uma vez que o seu valor não pode ser modificado
  - *Métodos são públicos*
    - Métodos que serão usados apenas dentro da própria classe podem (e devem) ser privados

Diagrama de classes (UML)

- Descreve as classes componentes de um sistema (ou parte dele)
- É uma linguagem visual de modelagem, independente de linguagem de programação
  - Mais fácil de visualizar e entender o sistema
- Fornece uma documentação aos programadores
  - Implementam as classes do sistema a partir do diagrama
  - É possível também gerar código a partir do diagrama
- É um dos diagramas da linguagem de modelagem unificada (UML)

Diagrama de classes

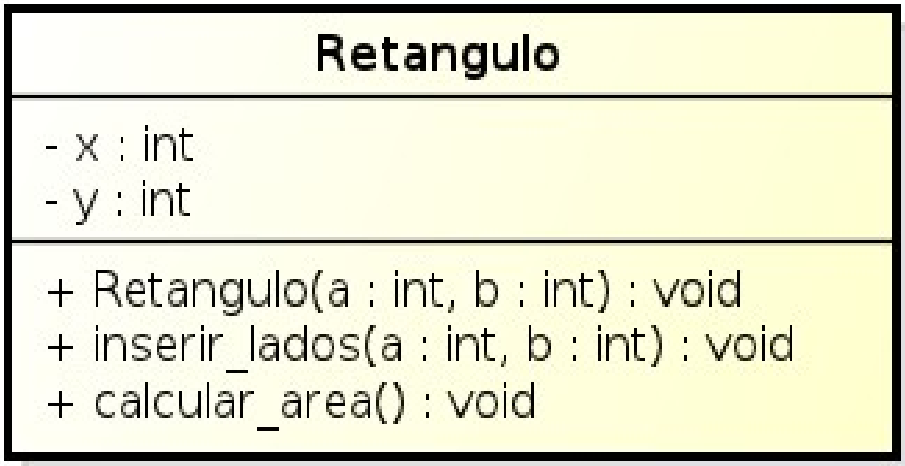
- O Diagrama é dividido em três partes:
  1) Nome da classe
  2) Atributos
  3) Métodos

- Atributos
  - <acesso> <nome> : <tipo>

- Métodos
  - <acesso> <nome> (<parâmetros>) : <tipo>

    - *acesso público: sinal de “+”*
    - *acesso privado: sina de “-”*

Exemplo: classe Retângulo




```
class Retangulo {
    private:
        int x, y;
    public:
        Retangulo(int a, int b);
        void inserir_lados (int a, int b);
        int calcular_area ();
};
```

Exercício

- Faça o diagrama das classes *Cliente* e *Funcionário (exemplo de sistema de banco visto em sala)*