PROGRAMAÇÃO ORIENTADA A OBJETOS

Professora Lyrane Bezerra

PARADIGMAS DE PROGRAMAÇÃO

Conceito que pressupõe a forma como o programador trata a abstração e organização do código para a programação e execução de um programa

- Programação estruturada (procedural)
  - Trabalhada a partir de uma rigorosa sequencia lógica de passos, determinada para solução um problema específico
  - Prima pela divisão do algoritmo em partes menores, a medida que o sistema vai
  - tomando proporções maiores.
    - Essas partes são chamadas de: subrotinas, funções ou procedimentos.
  - Permite uma “tradução” direta da lógica sequencial para uma linguagem de programação
  - É de fácil compreensão para iniciantes, contudo tende a deixar o código confuso
  - em sistemas de grande porte

PARADIGMAS DE PROGRAMAÇÃO

Programação Orientada a Objetos

Ao contrário da PE (que prima pela manipulação de várias tarefas isoladas para um objetivo comum), a POO busca meios de criar operações diretas com as entidades que compõem o problema

- Não basta identificar os passos, é necessário determinar quem (ou o que) será responsável por realizar cada um deles e/ou quem (ou que) será "estimulado" por eles
- quem (ou o que) == objetos do programa

Apesar de não possuírem o mesmo desempenho de códigos estruturados similares, oferece uma modularidade muito maior, com as seguintes vantagens:

- Melhor organização e mais semântica
- Maior reaproveitamento (bibliotecas multifuncionais)
- Códigos mais flexíveis e adaptáveis (ideal para sistemas com evolução em versões)

PROGRAMAÇÃO ORIENTADA A OBJETOS

- Origem nos anos 60 na Noruega, com Kristen Nygaard e Ole-Johan Dahl
  - (Centro Norueguês de Computação)
    - Primeira linguagem OO: Simula 67, extensão da ALGOL 60
    - Foram introduzidos os conceitos de classes de objetos e herança
- Apesar do novo paradigma trazer grande euforia à Computação, seu impacto só foi significante nos anos 80, com a linguagem Smalltalk
    - Projeto liderado por Alan Curtis Kay, nos laboratórios da Xerox
    - A partir dos seus conhecimentos em Biologia e Matemática, Alan Kay formulou sua analogia "algébrico-biológica"
    - Postulado de Alan Kay: o computador ideal deve funcionar como um organismo vivo, isto é, cada "célula" comportar-se-á relacionando-se com outras células a fim de alcançar um objetivo, entretanto, funcionando de forma autônoma.
- Desta forma, pode-se dizer que a POO está baseada na composição e interação entre diversas unidades de software chamadas de objetos

OBJETOS DE PROGRAMAÇÃO

Em um dado problema qualquer coisa é um objeto

- Objeto: seres vivos ou inanimados identificados no problema como agentes ativos ou passivos importantes para a solução do problema
- Objetos possuem características (atributos) e comportamentos (métodos)

Objetos realizam tarefas através da requisição de serviços

- Requisição de serviços é o estímulo dado a um objeto para que uma ação desejada seja executada por ele

Mensagens: recurso utilizado para prover a comunicação entre objetos

- Constituidas por: Nome do serviço requisitado, informação necessária para a execução do serviço e nome do requisitante
- Na pratica, mensagens são implementadas como ativações de um método definido no objeto chamado, no qual:
  - Nome do serviço: identifica um método a ser realizado
  - Informação necessária: lista de parâmetros para o método
  - Requisitante: objeto que realizou a chamada

CLASSES DE OBJETOS

- Definição formal para um conjunto de objetos com caracateristicas e ações semelhantes
  - Cada objeto pertence a uma determinada classe
  - Uma classe agrupa objetos similares
  - Define quais métodos e atributos um determinado objeto deverá possuir
  - Normalmente classes substantivos importantes para a solução do problema, enquanto verbos são operações candidatas
- Classes são organizadas em hierarquias
  - Um nível descendente de uma classe pode herdar características da camada superior da hierarquia

ANALOGIA: POO X PE

POO	PE
Objetos	Varáveis
Classes	Tipos
Métodos	Funções/procedimentos
Atributos	Varáveis (elementos de uma struct)

OUTROS CONCEITOS IMPORTANTES PARA A POO

Abstração

- Consiste em uma forma de pensar na solução de um problema de modo a gerenciar a sua complexidade
- A partir da descrição de um problema, deve-se focar e “modelar” apenas os aspectos essenciais

Encapsulamento

- Um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos
- Técnica que faz com que detalhes internos do funcionamento dos métodos de uma classe permaneçam **ocultos** para os objetos

PROGRAMAR OO É MODELAR!

Identificar classes == identificar entidades (como em BD)

- Ex.: um elevador pode ser uma classe, pois possui atributos (características) e métodos (comprtamento)
- Dependendo do problema, a classe elevador pode ser importante ou não (abstração)

- Classe elevador - atributos:
  - Peso máximo
  - Peso atual
  - Estado da porta
  - Estado do motor
  - Número do andar
- Classe elevador -métodos:
  - Abrir porta
  - Fechar porta
  - Ligar motor/subir
  - Ligar motor/descer
  - Para motor

Dois objetos da classe elevador!

MODELOS PARA POO

- Linguagem de Modelagem Unificada (do inglês, **UML - Unified Modeling Language**)
  - Diagrama de Classes
    - Define a estrutura das classes do sistema
      - Apresenta uma visão estática de como as classes estão organizadas
    - Estabelece como as classes se relacionam

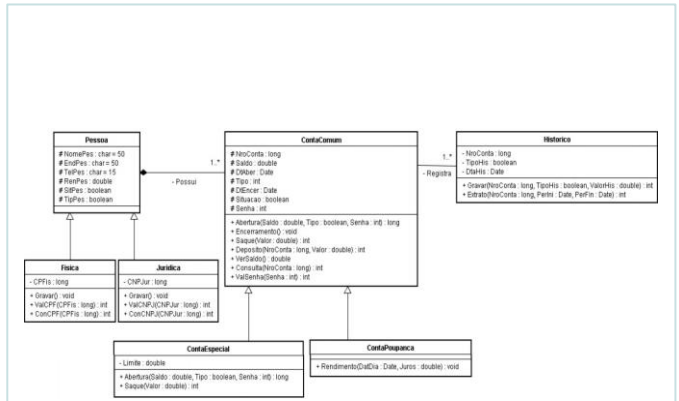

DIAGRAMA DE CLASSES E OO

- Representa:
  - Atributos e métodos de uma classe
  - Os relacionamentos entre classes
    - Associação
      - Agregação
      - Composição
    - Generalização (herança)
    - Dependência

DIAGRAMA DE CLASSES E OO

- Representação de uma Classe
  - Uma classe é representada por um retângulo com três divisões
    - Nome da Classe
    - Atributos da Classe
    - Métodos da Classe

PILARES DA POO

EXERCÍCIO

- Escolha um objeto/ser a sua volta e cite alguns de seus atributos e comportamentos
  - Modele uma classe que se sirva de definição para esse objeto em duas perspectivas diferentes de abstração