**RESUMO DA AULA 04 - OBJETOS, CLASSES E MÉTODOS**

ALUNO: Denilson José do Bom Jesus Silva de Lima

Os componentes principais de um Software são os Dados (Data) e as Funções. Em LPI, normalmente para construir um projeto, aplica-se um conceito de Decomposição Funcional, onde existe o projeto como um todo e vai-se dividindo-o em diversas funções para construí-lo.

Com a Orientação a Objetos primeiro se define as Estruturas dos Dados e depois os métodos; primeiro modula-se dos dados e depois se define as operações onde cada objeto (a óptica por onde se analisa tudo) é responsável pela integridade de seus próprios dados (encapsulamento = a mudança na estrutura de um objeto diz respeito somente a sua própria lógica interna).

Objetos é uma estrutura complexa que possui dados e operações (encapsulamento); dados podem ser Objetos e os métodos manifestam o comportamento de um objeto.

Normalmente um atributo ao ser implementado recebe chamada "Private". Os atributos são manipulados pelos métodos da classe. Uma classe é uma abstração que define características comuns entre objetos - agrupa categorias semelhantes; define um tipo.

O Objeto existe em tempo de execução - enquanto o seu uso, já a Classe na ideia própria do desenvolvimento. Objetos são partes de uma classe e possuem Identidade, Estado e Comportamento.

A Identidade no Objeto se dá quando dois objetos são idênticos, o mesmo. A Instanciação reserva um espaço na memória para armazenar os atributos de um novo objeto e associa comportamentos ao mesmo.

Tipos de relacionamento entre classes: Usa, Inclusão (contém), É Um (herança). O relacionamento de Inclusão diz respeito a uma classe possuir como um de seus atributos um objeto de outra classe.

Para definir classes em Java, o nome da classe deve começar com letra maiúscula, não usa-se separador "\_" e deve-se evitar abreviações; o modelo a seguir é:

*modificador class NomedaClasse {Corpo (atributos, construtores e métodos)}*

A declaração de atributos se dá por modificadores de acesso - como private, public e protected -, de escopo - static -, de definição de constante - ex: final -, um tipo (ex:. String, Int) e um nome, com o modelo:

*modificadores tipo nome;*

Os nomes de atributo seguem um padrão de começar com letra minúscula, não possuir separação por "\_" e evitar abreviações.

Uso do this - modificador que pode ser usado para acessar atributos de um objeto - opcional (quando não necessário).

A declaração de métodos, que segue o mesmo padrão de nome para atributos, dá-se seguindo o modelo:

*modificadores tipo de retorno nome (parâmetros);* ex: public int media (String num1, int parametro);

Os tipos de retorno de um método podem ser do Tipo Primitivo ou Tipo Referenciado; se o método retorna algum valor, usa-se o comando return se não, possui o tipo void.

O corpo de um método contém os comandos que determinam suas ações no próprio método, como: retornar valores, atualizar os atributos de um objeto, chamar métodos de outros objetos, etc. Um boa prática é declarar atributos sempre como private.

Para permitir acesso de leitura ou de escrita num dado atributo Y, deve-se criar um método getY (devolve o valor do atributo) ou setY (modifica o valor do atributo), respectivamente.

A chamada de métodos é feita seguindo o modelo:

(Ex:.) *conta.creditar(45.30)* - onde "conta" é a variável contendo, o ponto "." é o operador de chamada de método e "creditar" é o nome do método a ser executado.

Os objetos se comunicam para realizar tarefas e essa comunicação é feita através da chamada de métodos.

Objetos são criados através do operador "new" onde aloca um objeto na memória do tipo de uma classe e devolve o endereço para armazenar na variável - ex:

*Conta c = new Conta()* - onde "Conta()" é o construtor.

Todo objeto, por padrão, ao ser criado possui o "construtor" vazio. O construtor não tem nenhum tipo de retorno e tem o mesmo nome da classe; possui modificador.

É possível ter mais de um construtor, ou seja, instanciar o objeto na classe de uma maneira diferente com o comando "new".

Caso não seja definido um construtor, um construtor default é fornecido implicitamente e inicializa os atributos com seus valores default. O construtor

default não tem parâmetros. Uma classe pode ter mais de um construtor que se diferenciam pela lista de parâmetros. Quando um construtor com parâmetros

é definido, o default não é mais gerado.

Valores default para os atributos byte, short, int, long, float e double é *0*; para char é *\uoooo*; para boolean é *false* e para os tipos referenciais é *null*.

Os Objetos são manipulados através de referências e mais de uma variável pode armazenar a mesma referência para um dado objeto (aliasing).

(Ex:)

*Conta a;*

*Conta b;*

*b = a;* - onde b e a passam a referenciar a mesma coisa.

Métodos deprecated apresentam algum problema, bug ou efeito colateral e são corrigidos em atualizações do Java.

Garbage Collection é automação no gerenciamento da memória do Java - limpa a memória após o uso.

Resumo de Princípios de OO:

- Dividir-e-conquistar, Encapsulamento, Interface, Esconder informação, Generalização, Extensibilidade e Abstração.