Elementos de programação Orientada a Objeto (conceitos gerais)

A programação como qualquer outra atividade do dia a dia, possui diferentes métodos para ser realizada. Os métodos são chamado de **paradigmas de programação**, envolvendo a programação estruturada e a **programação orientada a objetos (POO),** em que nos aprofundaremos nesse tópico.

Paradigmas de programação pode ser explicado como um tipo de estruturação pela qual a linguagem deverá seguir. Depende-se de qual objetivo deverá ser alcançado, a solução que será oferecida obedecerá a um tipo de paradigma.

Em paradigma de POO o foco da programação é em cima dos dados e sua estruturação e em seguida examina os algoritmos que manipularão esses dados.

Embora a POO exista desde a década de 60, os últimos 10 anos têm visto um crescimento exponencial no uso e aceitação de tecnologias de objeto, sendo o paradigma mais utilizado no momento para o desenvolvimento de projetos de software. Dentre as linguagens que aceitam OO, podemos destacar SMALTALK, SIMULA-67, EIFFEL, Modula-3, **JAVA**, C++, Object Pascal, C#, etc.

As características marcantes do paradigma da orientação a objetos são:

* A alta reutilização de software: O software produzido é encapsulado em unidades básicas denominadas **classes** que possuem semânticas que podem ser reaproveitadas em novos projetos;
* Modelos: Uma representação adequadamente simplificada do mundo real, facilitando o entendimento e a manutenção.
* Desenvolvimento sempre evolutivo;
* Construções complexas a partir de construções simples;
* Manutenção: impactos sobre o software tendem a serem minimizados pelo uso da tecnologia de **objetos**.

A metodologia OO é baseada em “objetos do mundo real”, e por este motivo, é mais intuitiva ao ser humano, tais como: objetos e atributos, classes e membros, estruturas e componentes, ação e reação. Os métodos de desenvolvimento de software anteriores ao surgimento desse paradigma organizam a especificação de um sistema de acordo com suas funções ou com os dados manipulados.

Em um sistema orientado a objetos, os dados e todas as operações, que manipulam esses dados, são agrupados em uma única estrutura: “as classes”. Desde o início do desenvolvimento desses sistemas e, em todas as suas fases, o analista trabalha com o mesmo elemento de abstração, os objetos.

Como a área tecnológica está em constante mudança, tecnologias novas, objetivos novos e pessoas mudando seu jeito de pensar ao longo do tempo, os softwares precisam se manter atualizados e as pessoas não podem esperar muito tempo.

Portanto, precisamos de um paradigma de programação capaz de diminuir o tempo e o custo de mudanças e aumentar a capacidade de adaptação e, por isso, os objetos são ótimos recursos para melhorar isso.

* Classes

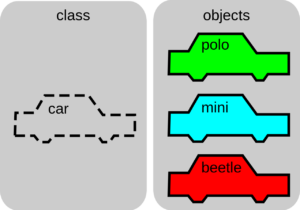
As classes, na Programação Orientada a Objeto, são projetos (**modelos**) de um elemento “da vida real” (objeto). As classes são as partes mais importantes de qualquer sistema orientado a objetos, pois elas representam a modelagem do objeto do mundo real no sistema computacional. Usamos as classes para capturar o vocabulário do sistema que está em desenvolvimento.

Figura 1- Classes

Uma classe é um conjunto de características e comportamentos que definem o conjunto de objetos pertencentes à essa classe. Repare que a classe em si é um conceito abstrato, como um molde, que se torna concreto e palpável através da criação de um objeto. Chamamos essa criação de *instanciação da classe*, como se estivéssemos usando esse molde (classe) para criar um objeto.

* **Objeto**

Objetos são quaisquer coisas na natureza que possuam propriedades (características) e comportamentos (operações). Orientação a objetos significa organizar o mundo real como uma coleção de objetosque incorporam estrutura de dados e um conjunto de operações que manipulam estes dados.

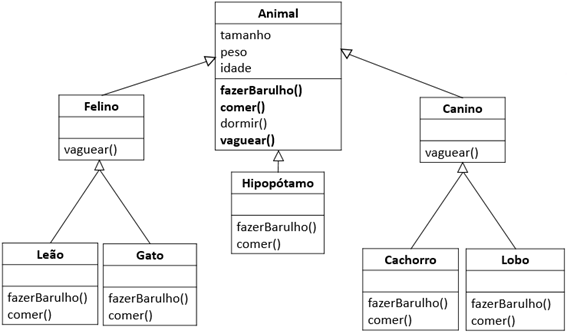


Figura 2 – Objetos

A Classe é como um "molde" de objetos. A partir de uma Classe, podemos gerar em memória vários objetos que possuirão a mesma estrutura, que poderão realizar as mesmas ações, mas que serão independentes entre si e que, no geral, possuirão informações diferentes.

* **Atributos**

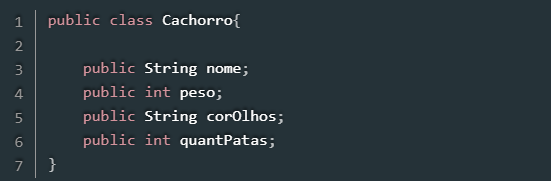
Os atributos são as propriedades de um objeto, também são conhecidos como variáveis ou campos. Essas propriedades definem o estado de um objeto, fazendo com que esses valores possam sofrer alterações. A Figura 3 mostra as características de um cachorro.

Figura 3 – Atributos

Logo em seguida na figura 4, tem-se a palavra “Cachorro” instanciada três vezes, diferenciando as características de cada um.Figura 4 - Atributos

* Métodos

Métodos são comportamentos, ações e reações dos objetos, uma função fornecido pelo objeto para outros objetos. Podendo manipular os atributos do objeto. A execução dessas ações, assim como a resposta dessas ações, se dá através de "mensagens", tendo como objetivo o envio de uma solicitação ao objeto para que seja efetuada a rotina (ação) desejada, e para que seja devolvida uma resposta, quando necessário.

Exemplos: Um automóvel tem o comportamento de se locomover.

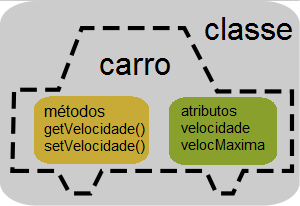


Figura 5 – Métodos

Como boas práticas, é indicado sempre usar o nome dos métodos declarados como verbos, para que quando for efetuada alguma manutenção seja de fácil entendimento. Veja algumas nomenclaturas de nomes de métodos:

* acaoVoltar
* voltar
* avançar
* correr
* resgatarValor
* pesquisarNomes
* **Método “Main”**

Usa-se o Método main para inicializar um programa Java para se criar outros métodos primeiro deve-se adicionar o método “main” ao código. Para que seja possível adicionar o método, precisará ter ao menos uma classe existente.

Sendo a seguinte sintaxe para adicionar:

Public class ClasseQualquer {

public static void main (String[ ] args) {

}

}

* Anatomia do Método “Main”

Public: qualquer classe poderá acessar esse método.

Static: os instanciamentos da classe vao compartilhar a cópia do método main.

Void: O método não retornará valor.

(String[ ] args): Argumentos obrigátorios para o método.

* **Variáveis**

A linguagem Java é conhecida como fortemente tipada, sendo obrigatória a declaração do tipo da variável.

A variável guarda as informações e pode reservar o tamanho de espaço especifico que pode ser guardado nela.

Ex:

* short: números de -32.768 a 32.767
* Int: Números de -2.147.483.648 a 2.147.483.647 (pode ser reservado um espaço maior a esta variável)
* long: números de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807 (pode ser reservado um espaço maior que int à esta variável)
* float: ótima precisão para números reais com até 7 casas decimais.
* double: ótima precisão para números reais com até 15 casas decimais.