Java Básico 07/10/22

Orientação a objeto

Ao compilarmos o programa java ele se torna um arquivo .class que é o bytecode. E independentemente da plataforma ele funciona.

* A orientação a objetos, pode ser considerada uma outra forma de implementação de sistemas computacionais.
* A programação orientada a objeto, é um método relativamente novo na concepção e implementação de software.

Seus maiores objetivos são:

* Aumentar a produtividade do programador através do reuso de software.
* Controlar a complexidade e o custo de manutenção do software.
* A programação orientada a objetos possuí vários conceitos, destacando-se:
* Tipo abstrato de dados
* Herança
* Encapsulamento
* Polimorfismo

Com o objetivo principal permitir que os programas sejam organizados de forma a espelhar o modo como os objetos são organizados no mundo real.

Tipo abstrato de dados

* Um tipo abstrato de dado é um modelo formado por propriedades (atributos) e operações (métodos)
* Classe: representa a implementação de um tipo de dado abstrato
* Objeto: é a instância de uma classe possuindo valores próprios para os atributos definidos na classe.

Exemplos

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

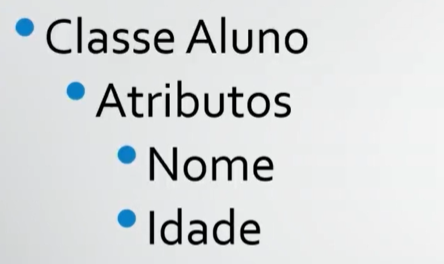
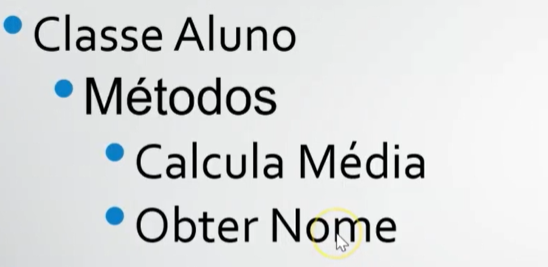
Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Objeto é o dado que recebe os atributos, como exemplo da classe pessoa. Cada pessoa é um objeto diferente.

Outros exemplos para entender métodos.

Métodos são as ações que podem ser feitas com os atributos que temos

Exemplo em código

Linha do tempo

Descrição gerada automaticamente

Método construtor

* Os construtores constituem um tipo especial de método, chamado quando um objeto é inicializado
* O objetivo de um construtor é permitir a inicialização dos atributos
* O construtor tem o mesmo nome da classe em questão
* O construtor é sempre invocado pela palavra new

Encapsulamento

Usando os getters and setters

Quando o método da classe é public, mas os atributos não a gente pode visualizar

Para acessar os elementos precisamos do get, o encapsulamento basicamente são atributos privados que são acessados por métodos públicos

Usado para esconder os detalhes da implementação das classes, permitindo um maior domínio do projeto. Mostrando somente o que a classe pode fazer, mas não necessariamente o que ela faz

Uma classe deve impedir o acesso direto aos seus atributos e métodos internos e disponibilizar métodos públicos.

Controle de acesso

Os membros de uma classe podem ser protegidos através das formas de acesso

As principais são public e private.

* Public: membros (métodos ou atributos) podem ser acessados livremente
* Private: membros (métodos ou atributos) só podem ser acessados pela própria classe

Mensagem

* A forma como o objeto interage com o outro objeto
* Corresponde a chamada (execução) de um método do objeto, o que pode exigir a passagem de parâmetro(s);

Encapsulamento e Herança

Métodos acessadores e modificadores

* Método acessador: permite acessar os atributos de um objeto.
* Utiliza-se como assinatura o nome get+atributo. Ex: getNome
* Método modificador: permite modificar um atributo de um objeto
* Utiliza-se como assinatura o nome set+atributo. Ex: setNome();

Herança

* Redefinição de métodos
* Uso do identificador extends
* Uma subclasse pode possuir apenas uma superclasse
* Herança múltipla não é permitida em Java

Ex:

Diagrama

Descrição gerada automaticamente

Assim podemos receber os atributos de cliente nas outras classes

Como exemplo: nome,endereço.

E as diferenças de pessoa física e jurídica é o documento de identificação, no caso cpf e cnpj.