Java SE Platform at a Glance (gráfico de estrutura de java)

Java Se (Standart Edition)

Java EE (Enterprise Edition)

Java ME (Micro Edition)

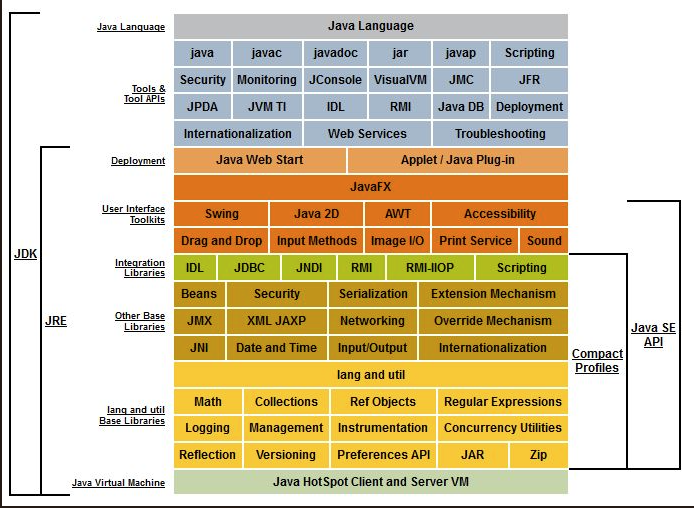
Java Card (Parte de cartão)

JSR – java especification request

JCP - Java Community process

Variável primitiva - faz atribuição por copia

x

variável derivada/classe – atribui por ponteiro ou atribuição por referência

SOLID:

S = SRP – cada classe tem que tomar conta daquilo que ela nasceu pra fazer

Objetos de negócio do banco: Funcionários, Conta, Cliente

As principais classes dos sistemas são aquelas que emulam as entidades do mundo real, nome mais técnico/especial de JAVA BEANS, do ponto de vista estrutural contenta os atributos do objeto de negócio, devem ser obrigatoriamente privados, para os métodos públicos serem utilizados.

Encapsulamento em java, esconde parte do código através do private, utilizando os métodos públicocomo porta de entrada.

Modificador de acesso / diretiva de visibilidade, no java existem 4 modificadores de acesso.

Protected: fica no meio do caminho, usado quando quer montar uma estrutura de heranças, usado para quando uma classe filha quer usar

Public: o mais vísivel

Private: mais restritivo, só o dono pode acessar

Default – tem haver com pacotes

Conceituamente falando quando você cria um método pra acessar atributos privado, está criando uma interface, conecta elas de uma forma menos acoplada, em orientação objeto, menos manutenção e menos re-uso, é **muito mais importante você aprender a mexer com interface do que implementação**.

É quando você quer usar polimorfismo porém elas não estão na mesma linha de herança

A interface por definição só vai conter a regra de negócio, aa aplicação será na classe

Por default o construtor é vazio implícito, quando eu crio um construtor explicito e crio outro

**Static -** todos os ponteiros apontando pro mesmo endereço de memória, Atributo de classe

**Como faz pra criar um método na classe**

**Herança:**

**Construção top to bottom –** Criando as classes mais gerais e depois as mais especializadas, através do extends: **public** **class** Gerente **extends** Funcionario {

**Polimorfismo – existem vários tipos:**

**Sobreescrita –**

Classes abstratas, o que é abstrato do ponto de vista de completo:]

É aqui que não é concreto

Tipo a classe Pessoa que tem como filho PF e PJ, Pessoa sozinho não existe.

Design Pattern + SOLID – próxima etapas da programação POO

O java funciona com o empilhamento dos métodos no stackframe

**Exceções** (TesteExcepetion1 e 2)

Para regras de negócio é o try, e o catch é a conseguência que pode acontecer em função dessa regra de negócio, isso evita da JVM destruir o processo que está executando