**Teoria sobre JAVA**

* Na programação existe o código fonte, objeto e executável O código fonte é o feito em uma IDE usada para criar algoritmos e após o código fonte ser feito, se usa um “compilador” para transformá-lo em código objeto e ai sim através do “Linker” se consegue usar este algoritmo como executável.
* Existe também uma maneira de mover o código fonte direto como um executável que é usando um “interpretador”, mas é claro que de vista o interpretador parece ser bem melhor de se usar, porém existem desvantagens.
* Um fluxograma é uma representação gráfica de algoritmos.

**Teoria - Como funciona a linguagem JAVA**

* O compilador transforma o algoritmo do código fonte em linguagem de máquina.
* O Java é um pouco diferente, o compilador do JAVA é “JavaC” ou também conhecido como “JavaCompiler”, após o código fonte passar pelo compilador é gerado um “Bytecode” ao invés de ser como os outros compiladores que transformam em executáveis (Não é código executável).
* Para o Bytecode ser transformado em executável ele vai precisar passar pelo “Java Virtual Machine” ou “JVM” (Máquina virtual Java), apenas assim o código se torna executável para os computadores.
* A vantagem do Bytecode é que ele transforma um mesmo “Código objeto” para qualquer plataforma (Windows, MacOS, Linux, dispositivos móveis e outros.), diferente das outras linguagens que tem compiladores diferentes pra cada uma delas.



* **WORA: (Write Once Run Anywhere)** “Escreva uma vez e execute em qualquer lugar”.
* Existem dois grupos em JAVA - **JRE**: Destinado a quem quer apenas usar as aplicações JAVA, já quem quer desenvolver vai usar o **JDK.**
* **JAVA RUNTIME ENVIROMENT:** “Ambiente de execução java”

O JRE é composto por duas partes (JVM E BIBLIOTECAS) bibliotecas são API’s que são utilizadas para o programa ficar mais atrativo e com mais funcionalidades.

A JVM possui algumas partes internas como o **“Loader”** que carrega o Bytecode na memória da máquina virtual e o **“Verificador”** que é usado para estar verificando o código se ele é executável ou não. Além disso tem também o **“interpretador”** onde vai pegar o código fonte e transformá-lo na linguagem nativa do hardware/Software em questão e o **“Gerenciador de memória”** que vai tratar como os códigos vão ser gerenciados na memória da JVM. E por fim também tem o “Compilador JIT” (Just in time - “Tradução em tempo real”), além destes, existem outros componentes, porém estes são os principais.

* **JAVA DEVELOPMENT KIT: “**Kit de desenvolvimento java”

Já vem com a JRE inclusa, Linguagem “JavaLang” e conjunto de ferramentas “JavaTools” algumas destas ferramentas como javaC (Compilador Java), Debugger (Permite verificar como o programa está sendo executado em tempo real), API’s (Bibliotecas com funcionalidades diversas para desenvolvimento de software).

Pode-se vincular também ao JDK uma IDE (ambiente de desenvolvimento facilitador).

