# Aula I de Desenvolvimento de Aplicativo I

**Objetivo da Disciplina**
**Conhecimentos e Objetivos Específicos**
**Lógica de Programação**
**Algoritmos e sua representação**
**Introdução a Linguagem de Programação Java**
**Características do Java**
**Principais características;**
**Variáveis e constantes;**
**Convenções de Nomes na declaração do Java**
**Bloco de programa do Java**

---

***Fontes:***
https://www.oracle.com/java/technologies/javase-documentation.html **----> DOCUMENTAÇÃO**
https://www.guj.com.br/t/documentacao-java-em-portugues/377706 **----> DOCUMENTAÇÃO**
https://docs.oracle.com/en/java/javase/15/ **----> DOCUMENTAÇÃO**
https://developer.ibm.com/br/tutorials/j-introtojava1/
https://www.devmedia.com.br/iniciando-na-linguagem-java/21136
http://ftp.unicamp.br/pub/apoio/treinamentos/linguagens/java_basico.pdf

---

## Objetivo da Disciplina

Desenvolver aplicativos para dispositivos móveis utilizando tecnologia nativa.

## Conhecimentos específicos da disciplina

- Conhecer a estrutura da plataforma Android ou similar
- Criar interfaces amigáveis utilizando os padrões de interface para dispositivos móveis.
- Aplicar os recursos da linguagem Java ou similar para o desenvolvimento de aplicativos para dispositivos móveis.
- Utilizar ferramenta de controle de versionamento (GitHub ou GitLab)
- Conceito de programação orientada a objeto (classes, atributos, métodos, argumento, objetos, encapsulamento, visibilidade)
- Variáveis, constantes e tipos de dados
- Estruturas condicionais e de repetição
- Estruturas de dados unidimensionais (vetores)
- Banco de dados para aplicativos nativos Layout para aplicativos nativos

## Algoritmos

Uma boa definição do que é um algoritmo é de que o mesmo trata-se de “uma sequência finita de passos para solucionar um problema”. Executamos algoritmos em nosso dia-a-dia a todo tempo, como por exemplo, sair de casa e ir até o trabalho, realizar compras em um mercado, os passos desde o momento em que acordamos até chegarmos à parada de ônibus, etc.
Mesmo para essas questões do cotidiano, podem-se ter diversos algoritmos para solucionar o mesmo problema, e que muitas vezes, estes só funcionam dependendo de fatores externos. Como fazer um bolo de chocolate à noite visto que faltou luz? A solução provavelmente será diferente da solução convencional.

## Método para a construção de algoritmos

Para a construção de qualquer tipo de algoritmo, é necessário seguir estes passos:

1. Entender completamente o problema a ser resolvido, destacando os pontos mais importantes.
2. Definir os dados de entrada
3. Criar um passo-a-passo para a solução do problema.

O computador/dispoistivo pode auxiliá-lo em qualquer tarefa. Ele tem um processamento muito rápido e não cansa, porém ele não tem iniciativa e de certa forma não é inteligente, precisa ser programado para executar suas tarefas e para isso utilizamos a lógica de programação.

## Lógica

Normalmente utiliza-se a palavra “lógica” no dia-a-dia para indicar algo coerente e racional. Quando se encadeia alguns pensamentos e chega-se a uma conclusão, pode-se ao final concluir.
Pode-se dizer que a lógica seria a “correção do pensamento”, pois uma de suas preocupações é determinar quais operações são válidas e quais são inválidas.

### Exemplos

a) Todo mamífero é um animal.
Todo cachorro é um mamífero.
Portanto, todo cachorro é um animal

b) José é mais alto que Maria.
João é mais alto que José.
Portanto, João é mais alto que Maria.

c) Todos os filhos de João são mais altos do que Maria
Antônio é filho de João

#### *O que pode-se concluir logicamente?*

Como Antônio é filho de João, e os filhos de João são mais altos que Maria, logo, Antônio é mais alto que Maria. Com a mesma sentença e usando lógica, verifica-se também que Maria é menor do que Antônio, além disso, nada pode-se concluir entre a altura de Maria e João.

Esses exemplos representam um argumento composto de duas premissas e uma conclusão e estabelece uma relação que pode ser válida ou não. Esse é um dos objetivos da lógica. Outro desses objetivos é criar uma representação mais formal de apresentar premissas e conclusões, de forma diferente da linguagem natural que muitas vezes depende de interpretação.
Essa mesma sequência lógica de pensamentos, descritos em uma linguagem formal, chegando a conclusões válidas e inválidas será utilizada na lógica de programação para se criar programas de computador.

## Representação de um algoritmo

Algoritmos na verdade são linhas de raciocínio e podem ser representados de diversas maneiras, tanto de forma textual como em uma lista de instruções/ações utilizando o português, quanto de forma gráfica em formato de diagramas.
Nos diagramas, as palavras são substituídas por formas geométricas representando instruções e opções dentro do algoritmo. Segue exemplo de um algoritmo que descreve a troca de uma lâmpada.

**Algoritmo - Troca de uma lâmpada**

- A lâmpada não ligou
- se a lâmpada não estiver plugada
  - plugar a lâmpada;
- senão
  - se o bulbo queimou
  - pegar uma escada
  - colocar a escada embaixo da lâmpada
    - trocar o bulbo
  - senão
    - comprar uma lâmpada nova

Exemplos:

### Troca de resistência com teste e repetições

- ligar o chuveiro
- **se** o chuveiro não esquentar, **então**
- pegar uma escada;
- posicionar a escada embaixo do chuveiro
- buscar uma resistência nova
- subir na escada
- retirar a resistência queimada
- colocar a resistência nova
- **se** o chuveiro não esquentar, **então**
  - retirar a resistência queimada
  - colocar a resistência nova
  - **se** o chuveiro não esquentar, **então**
    - retirar a resistência queimada

        - colocar a resistência nova
        - **se** o chuveiro não esquentar, **então**
            - retirar a resistência queimada
            - colocar a resistência nova
            - .
            - .
            - .
            - **Até quando?**

É importante notar que o algoritmo acima não está terminado, falta determinar até quando será feito o teste do chuveiro. A execução do algoritmo deve terminar quando se conseguir colocar uma resistência que faça o chuveiro esquentar, caso contrário, os testes serão executados indefinidamente. Essa solução está mais próxima do objetivo, pois garante que o chuveiro volte a esquentar, mesmo que sejam trocadas várias resistências. Porém não se sabe qual vai ser o número exato de trocas.

## Proposta de Algoritmo:

https://canaltech.com.br/infra/fim-do-flash-resulta-em-travamento-de-linha-de-trem-na-china-entenda-177971/

Criar um algoritmo para solucionar o problema acima de forma adequada.

# Introdução ao Java

## Aspectos gerais

A documentação oficial com releases, recomendações e comentários, encontra-se neste link abaixo!

Java SE - Documentation ou ainda, em:
JDK 15 Documentation - Home

As dúvidas mais comuns de quem pretende começar a desenvolver na plataforma Java são:

1. Por onde começar?
2. O que preciso instalar?
3. Preciso de uma IDE?
4. Java EE, Java ME, Java SE, JSF, JDBC, JSP?

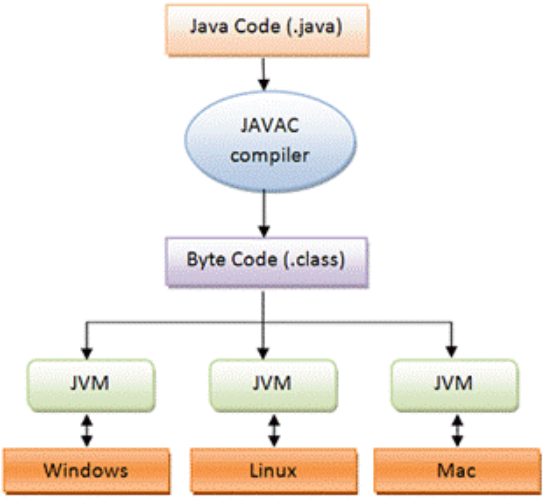
É importante entender que Java não é apenas uma linguagem de programação. Java é uma completa plataforma de desenvolvimento e execução, composta por três pilares:

1. **A máquina virtual java (JVM);**
2. **Um completo conjunto de APIs (bibliotecas);**
3. **A linguagem Java.**

Java é uma tecnologia independente de sistema operacional e hardware. Atualmente, ela está presente nos principais sistemas operacionais existentes, entre eles: Windows, Linux, Unix, Mac e Solaris. Além de ser utilizado para desenvolvimento de aplicações desktop, podemos adotar Java para implementar sistemas para web, dispositivos móveis, cartões de crédito, televisões digitais, geladeiras entre outros.

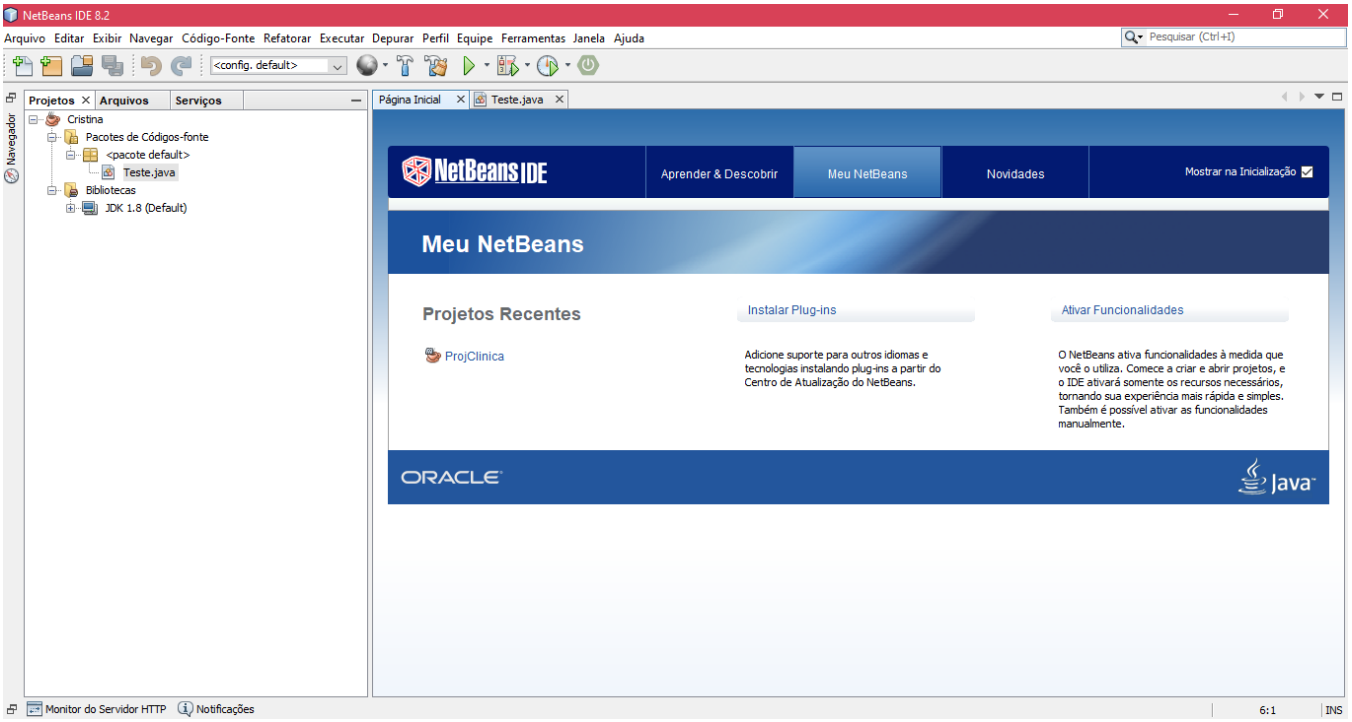
O funcionamento do Java. Quando compilamos o código, não é gerado um executável específico para um sistema operacional, como acontece em outras linguagens de programação. **O compilador, chamado javac, traduz o código para bytecodes, e estes bytecodes serão interpretados pela JVM.**

A JVM é uma máquina de computação abstrata (não física) que executa instruções, vindas de bytecodes, no sistema operacional e hardware específico sob o qual está rodando.




Fonte da Imagem: https://www.devmedia.com.br/iniciando-na-linguagem-java/21136

A IDE que utilizarei para estas primeiras aulas, será o NetBeans pela facilidade de configuração das bibliotecas do Java.




## Alguns conceitos direcionados para a programação

Computadores são máquinas que não tem iniciativa, elas executam as instruções que são passadas a elas. Informamos essas instruções por meio de algoritmos, porém para que os computadores entendam os algoritmos escritos é necessário transformar a sequência de passos que escreve-se em linguagem natural para uma linguagem que possa ser “entendida” pelo computador. Essas linguagens são chamadas de linguagens de programação.

## Memória

O primeiro passo para um programa ser executado em um computador é o carregamento do programa para a memória. Memórias são todos os componentes internos do computador que armazenam informações como a memória RAM e o disco rígido (HD). Esse armazenamento pode ser temporário, quando as informações são perdidas ao desligar o computador ou permanente, quando ao desligar o computador a informação permanece no dispositivo para os próximos usos.

Quando um programa é executado (tornando-se o que chamamos de processo), todo o seu conteúdo é colocado na memória RAM (*Random Access Memory – Memória de Acesso Randômico*) do computador. Sendo que é somente nela que o mesmo é executado. Essa memória é utilizada para armazenar as instruções dos programas, que são os comandos dados ao computador pelos algoritmos que serão construídos.

**Como um programa é tratado na memória?**
Todo o programa que é posto da memória RAM do computador é dividido em três áreas.


**Área de Dados:** É o local destinado para o armazenamento dos dados vindos do usuário, cálculos, saídas. Enfim, é o espaço que o algoritmo tem para trabalhar.
**Área de Código:** É o local onde está o algoritmo propriamente dito.
**Pilha:** É quem garante a continuidade do programa. Se o programa efetua "saltos" dentro do código, é a pilha que mantém a informação de onde o código estava antes do salto consequentemente garantindo um retorno correto ao final do salto. Pilhas utilizam uma estrutura de dados manipulada pelo que chamamos de **FILO (*First Input Last Output – Primeiro que entra é o último que sai*).**
Para guardar os dados na memória em sua área de dados, criam-se "rótulos" para cada informação a ser armazenada e posteriormente acessada pelos programas. Esses rótulos são chamados de **variáveis**.

## Variáveis

Basicamente, uma variável é um tipo de dado armazenado em um dispositivo. Chama-se variável, pois o valor contido nesse espaço de memória do computador é mutável ao longo da execução do programa. Esse valor pode ser lido do teclado, poderá ser obtido de um *hardware*, poderá ser o resultado de um cálculo dentro do programa, etc. Como bem se sabe, **o computador utiliza memória RAM** para armazenar dados temporários e essa memória é utilizada também pelos programas.

**Quando se declara uma variável em um programa, está na verdade se definindo e reservando um espaço na memória para armazenar o valor que aquela variável conterá em determinado tempo de execução do programa. Para que o computador saiba qual o tamanho da informação que deverá ser armazenada e possa reservar na memória a quantidade necessária de espaço, é necessário se definir o tipo dessa variável.**

Além do tipo, a variável precisará de um rótulo ou nome que a identifique dentro do programa e através desse nome a variável será referenciada em todo o programa.
A variável, antes de ser utilizada, precisa ser declarada. Na declaração, define-se o nome e o tipo da variável. Supondo que seja necessário fazer um programa que solicite ao usuário dois números, some esses dois números e apresente o resultado da soma para o usuário ao final, seriam necessárias as variáveis e seus respectivos tipos definidos.

## Entendendo o código fonte

Na **variável *número1*** armazena-se o primeiro número digitado, na variável ***número2*** o segundo número e na variável *soma* o resultado da soma desses dois números. Importante notar que foi utilizado o tipo inteiro nessa declaração de variáveis, dessa forma o computador sabe que precisa reservar um espaço na memória suficiente para armazenar um número inteiro negativo ou positivo sem as casas decimais.

## Tipos primitivos

O **tipo inteiro** utilizado no exemplo anterior **é um tipo primitivo no Java**, ou seja, ele é fornecido por esta linguagem de programação. Cada linguagem possui um conjunto de tipos primitivos "nativos" da linguagem. **A linguagem Java é fortemente tipada, ou seja, é necessário declarar o tipo de dado que irá passar pelo código!**

Tipos primitivos normalmente incluem a possibilidade de armazenar valores numéricos **inteiros, valores com vírgula (decimais ou também chamados de números de ponto flutuante) , texto, valores lógicos (verdadeiro ou falso) entre outros.**

### Alguns exemplos:

**Alguns tipos primitivos em Java:**

**int:** Toda e qualquer informação numérica que pertença ao conjunto dos números inteiros (negativo, zero ou positivo).
**Exemplos:**
**filhos = 3**
**casas = 2**
**degraus = 14**

**double:** Todo e qualquer valor numérico que pertença ao conjunto dos números reais (negativos, zero ou positivos).

**Exemplos:**
**altura = 1,71**
**peso = 75,5**
**saldo = -50,20**
**temperatura = -4,2**

**string:** Toda informação composta de um conjunto de caracteres alfanuméricos, numéricos (0..9), alfabéticos (A..Z, a..z) e ainda caracteres especiais (*, ?, !, @).

**Exemplos:**

**nome = “José da Silva”**
**cidade = “Porto Alegre”**
**bairro = “Vila Velha”**
**idade = “25”**

**Exemplos:**
**int n = 4;**
**long numero = 456L;**
**float pi = 3.14f;**
**double tamanho = 3.873457486513793;**
**string = nome;**

Resumindo: **Variáveis são utilizadas para representar valores desconhecidos, porém necessários para a resolução de um problema e que poderão ser alterados de acordo com a situação.**

## Constantes

Um determinado valor é constante quando não sofre alteração no decorrer do tempo, da mesma forma ocorre no desenvolvimento de software. O valor definido em uma constante será igual desde o início até o final do algoritmo.
Como as variáveis, as constantes possuem um nome e um tipo, porém se define um valor inicial para essa constante e ela não poderá mais ser alterada. Constantes são importantes para referenciar valores úteis ao longo do nosso programa, como o “Pi” para o cálculo da circunferência (3,1416).

## Convenção de Nomes no Java

O código fonte de programas Java é armazenado em arquivos com extensão **.java**. Um arquivo fonte poderá armazenar a definição de mais de uma classe, mas é uma boa prática armazenar apenas uma classe por arquivo fonte. Use o nome da classe principal do arquivo como sendo o nome do arquivo. Quando compilamos um arquivo fonte, será gerado um arquivo com extensão **.class** para cada classe definida e com o mesmo nome da classe. O arquivo .class contém o bytecode gerado com a compilação.

## Exemplos:

**Cliente.java , ItemEstoque.java**

## Nome de Classes:

Utilize substantivos ou frases substantivas descritivas para nome de classes. Deixe maiúscula a primeira letra de cada substantivo que compõe o nome, por exemplo: **MyFirstClassName.**

## Nome de Métodos:

Use verbos para nome de métodos. Faça a primeira letra do nome minúscula com cada letra inicial interna maiúscula. Por exemplo : **getUserName().**

## Nome de Variáveis:

Escolha nomes que indiquem o uso pretendido da variável. Utilize a primeira letra do nome minúscula e a inicial das outras palavras que formam o nome maiúscula. Exemplos : **customerName, customerCreditLimit.**

## Representação do código do programa do Java

### Estrutura do Programa Java

As primeiras coisas que devem ser abordadas para começar a desenvolver em qualquer linguagem são:

**- Bloco do programa;**
**- Declarar variáveis;**
**- Sintaxes dos comandos;**