Java
1
visão geral

Vitor Vaz da Silva

Introdução

1991 – Um grupo de projectistas da Sun "Green Team" tenta criar uma nova geração de computadores portáteis inteligentes e com grande capacidade de comunicação.

Esse processo criativo leva a criarem uma plataforma para o software ser portado para dispositivos diferentes.

Pensaram inicialmente em utilizar o C++ mas isso não facilitava a ideia que tinham de portabilidade

Introdução

Decidem criar uma nova linguagem OAK, nome em memória do carvalho que estava à frente do gabinete de James Gosling.

Desenvolveram também um sistema operativo Green OS para dar suporte a essa linguagem e uma interface gráfica padrão.

Introdução

Dificuldades de copyright levam à criação de um novo nome, inspirado nas bicas que a equipa tomava num café onde se reuniam

Java

Em 1994, a Java é utilizada quando foi desenvolvido a aplicação WebRunner capaz de realizar o download e executar o código Java pela Internet.

Introdução

A Sun decide disponibilizar gratuitamente a Java embora continue a deter os direitos relativos à linguagem e ferramentas.

Surge assim o JDK 1.0 – Java Developer's Kit

O JDK passa a ser chamado de J2SE
Java 2 Platform, Standard Edition

Há múltiplas plataformas e kits de desenvolvimento.

Características

- Orientado por objectos
  - Tudo são Classes

  - Excepto os tipos primitivos de dados

Características

- Independente das plataformas
  - Java é compilada para uma linguagem intermédia bytecode.
  - A bytecode é executada pela JVM
  - A JVM é a máquina virtual para java (Java Virtual Machine)

Características

- Ausência de ponteiros
  - Não há manipulação directa de endereços de memória
  - Não é necessário reciclar a memória

Características

- Alto desempenho
  - Linguagem Compilada - Java
  - Linguagem interpretada - Bytecode

Segurança

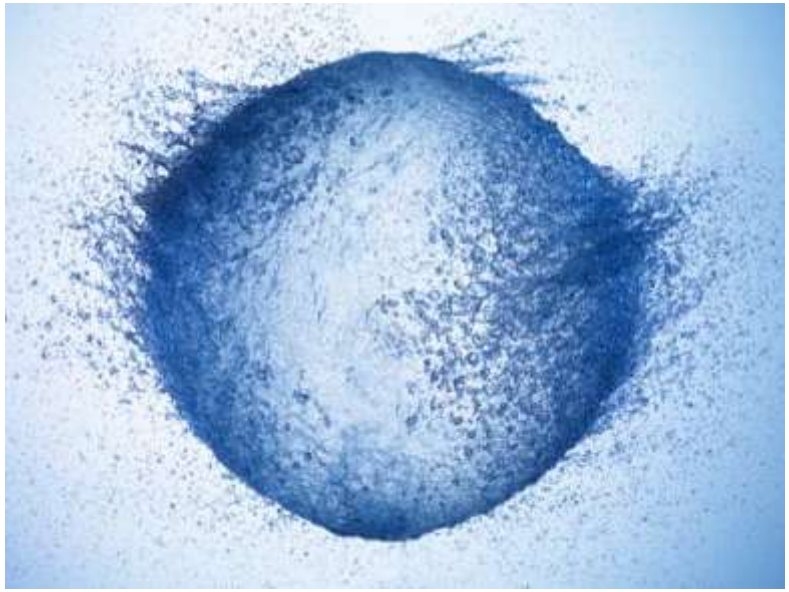
- Java possui mecanismos que impedem
  - Realizar qualquer operação no sistema de arquivos da máquina-alvo.
  - Caso o applet seja considerado seguro, são especificados diferentes níveis de acesso ao sistema alvo.

Concorrência

- Multithread – execução em simultâneo de rotinas.
- Thread é uma dessas rotinas.
- Contém mecanismos de sincronismo entre threads

E ainda….

- Tipo de inteiros e virgula flutuante compatíveis com o IEEE
- Suporta caracteres UNICODE
- Desenhada para desenvolvimento de aplicações distribuídas (e em rede)



?

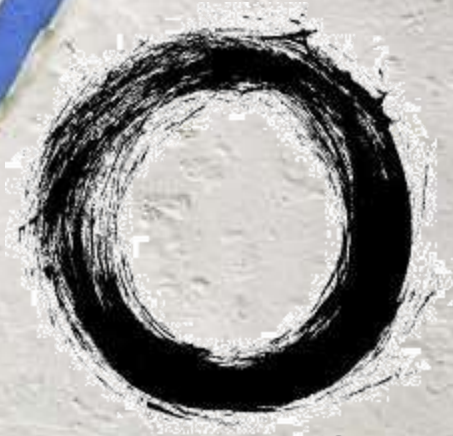
BOLA

?

CALÇADO

Aprender

Ensinar



Identidade

- Caracteriza um objecto
- Torna-o único
- Bilhete de Identidade

Classificação

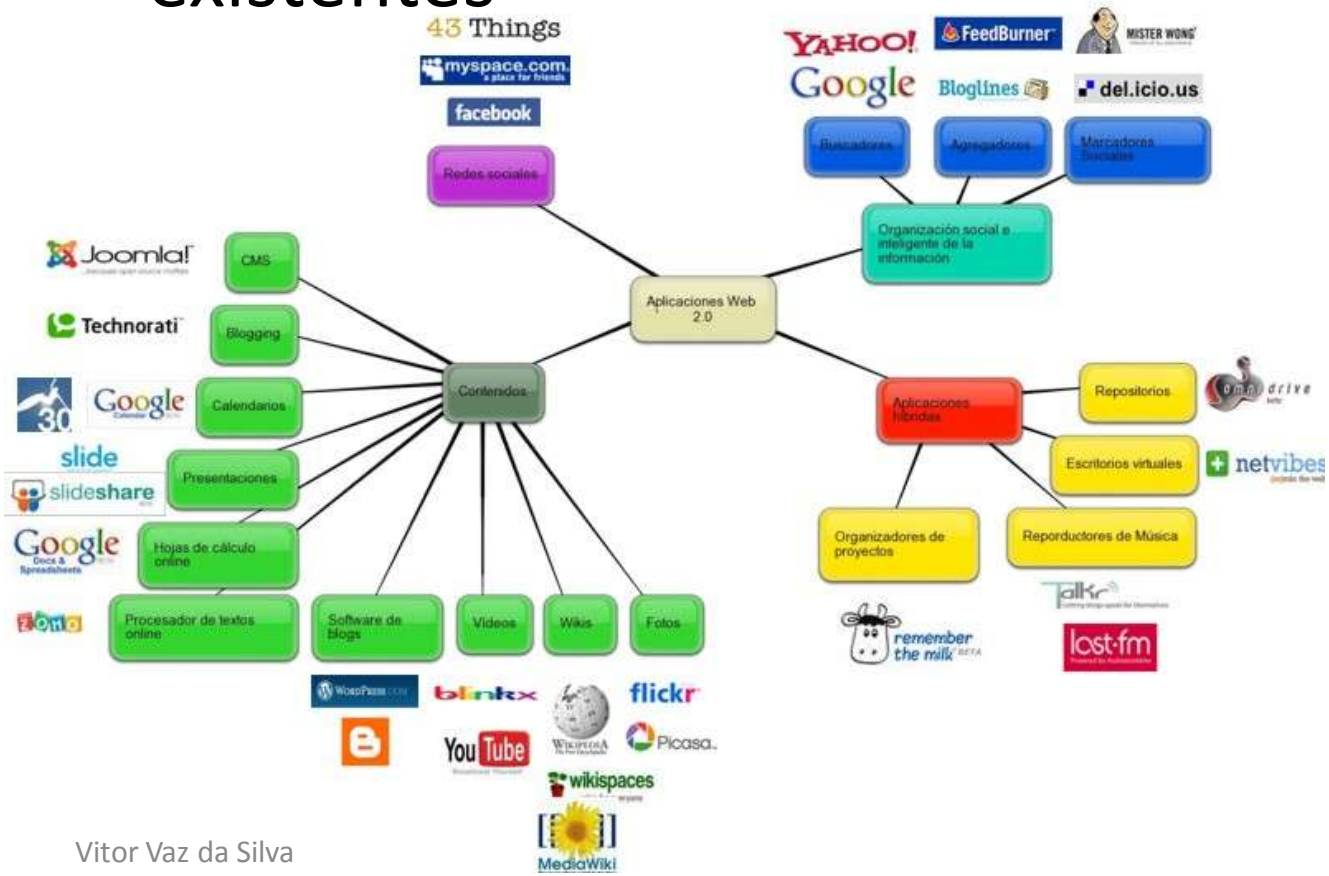
- Definir a que grupo (classe) pertence um objecto
- Sistemática – ordenação em grupos de acordo com critérios determinados

Polimorfismo

- Reconhecer num objecto outro mais geral

Hereditariedade

- Definir novas classes a partir de classes já existentes

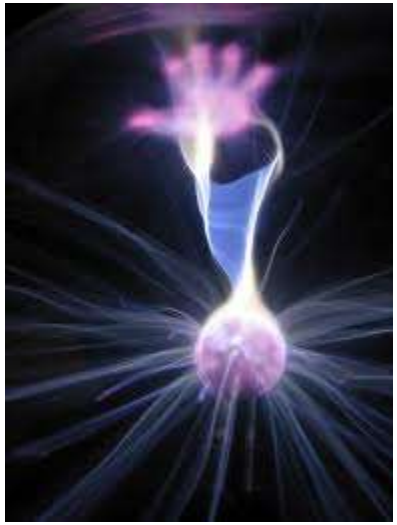


Encapsulamento

- Permite utilizar um objecto conhecendo apenas a sua interface.
- As propriedades internas ficam protegidas

Programar Orientado por Objectos

Pensar! Visualizar! Descrever!
O mundo em objectos
E os acontecimentos no
tecido do tempo



- Construir Classes
- Criar Objectos instanciando as classes

Classe

- Atributos
  - Características
  - O que é, como é
  - Dados
- Métodos
  - Funções
  - O que faz, como faz, quando faz
  - Manipula dados

Instanciação

- Instanciar – criar um novo objecto a partir de uma classe

- A classe e o objecto podem ter restrições, ou melhor qualificadores

Qualificador

Public – público

- O conteúdo da classe pode ser utilizado tanto pela classe como por outra classe que faça referência ao objecto

Qualificador

Protected – protegido

- A classe só pode ser referenciada por métodos que estão dentro do mesmo pacote ou em membros da própria classe

Qualificador

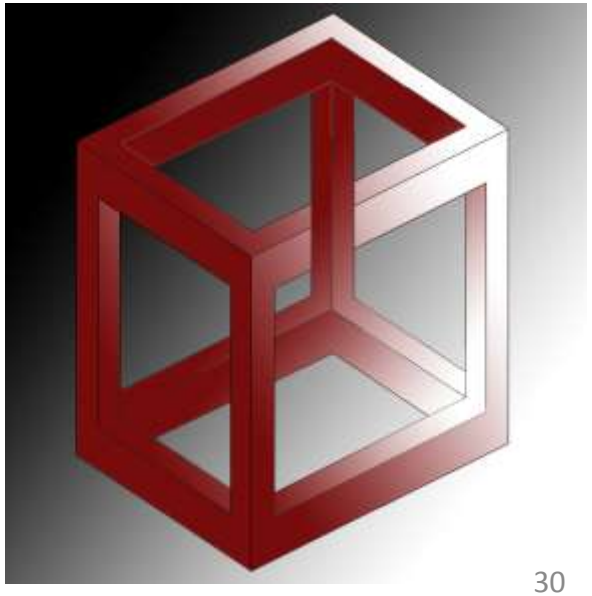
Private – privado

- Acesso restrito;
  membros da mesma classe.

Qualificador

Abstract – abstracto

- Possui métodos abstractos
- Não implementou todos os métodos de uma interface a que fez referência
- Especifica
  - o que fazer
- Não indica
  - como se faz



Qualificador

Final

- A classe não pode servir de base para herança de outra classe.

Qualificador

Strictfp

- Todos os valores utilizados são transformados em virgula flutuante

Qualificador

Static - estático

- Permite utilização de campos estáticos mesmo que a classe esteja dentro de outra classe

Identificadores

- Java distingue maiúsculas de minúsculas
- Classes – começam com Maiúscula

  public class EisUmaClasse{ … };

- Os objectos – começam com minúscula

  EisUmaClasse eisUmObjectoDaClasseEisUmaClasse;

- Constantes todos caracteres em Maiúculas

  Color.GREEN;

  Números e _ também podem ser utilizados excepto como primeiro caractere
  Acentos espaços e pontuação não podem ser utilizados nos identificdores

Pacotes

- Package
- Organização hierárquica semelhante às directorias
- Contém pacotes e classes
- Exemplo: java.awt.image

Import

- Indicar ao compilador onde encontrar as classes
- Pacote padrão já importado java.lang
- Usar directiva import
- Exemplo:

```
import java.awt.Graphics2D;
import java.io.*;
```

Package

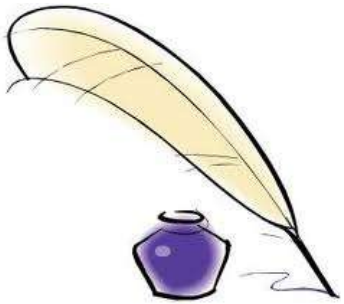
- Para incluir as classes em pacotes utiliza-se a directiva package.
- Exemplo

```
package minhas.classes;
public class Acetato{
….
}
```

- Algures …… import minhas.classes.Acetato;

Ficheiros e Classes

- Pormenores
- Um ficheiro só pode conter diversas classes mas só uma delas pode ser pública.
- O nome do ficheiro coincide com o nome da classe e com extensão java



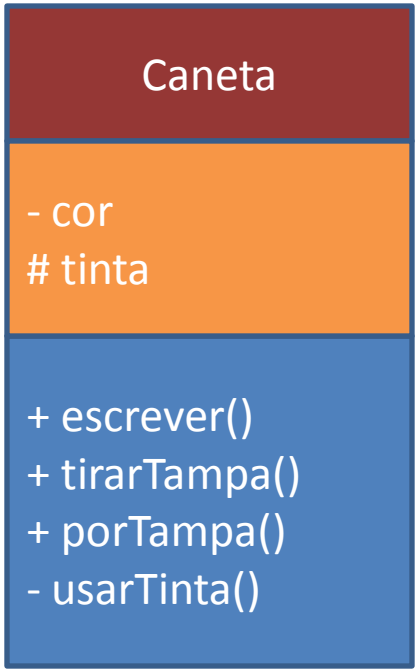
Class

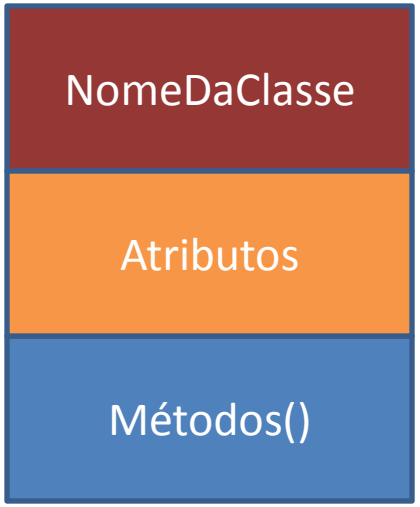
- Estrutura da classe
- Atributos - variáveis
- Métodos – funções

```
public class Caneta{
    int tinta;
    int cor;

    public void escreve(){ … }
}
```

UML

- UML – Unified Modeling Language
- OMT – Object Modeling Technique

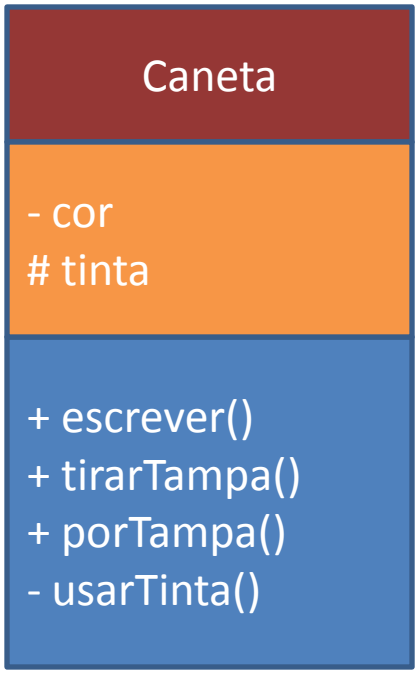




+ public
- private
protected

UML

```
Public class Caneta{
    private int cor;
    protected int tinta;

    public String escrever(){ … }
    public void tirarTampa(){ … }
    public void porTampa(){ … }
    protected void usarTinta(){ … }
}
```



SEM ACESSO

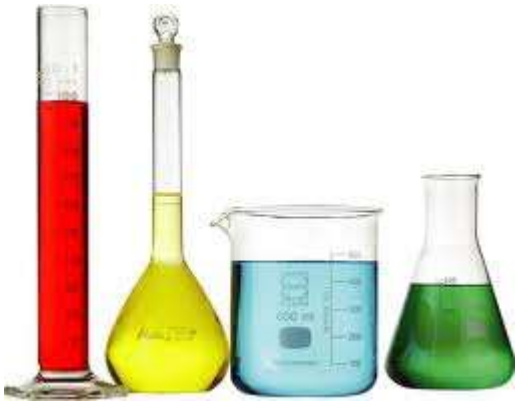
```
package editora;

import resma.Folha;

public class Impressao{

 public Impressao(){
 }

 public void inserir(){
   Folha pg = new Folha();

   pg.tamanhoA3();
   pg.tamanhoA4();
   pg.tamanhoA5();
}

 public static void main(String args[]){
   new Impressao(). inserir();
 }

}
```

```
package  resma;

public class Folha{

 public Folha(){
 }

 private void tamanhoA3(){
 }

 protected void tamanhoA4(){
 }

 public void tamanhoA5(){
 }

 public void tamanhoA2(){
   tamanhoA3();
   tamanhoA4();
   tamanhoA5();
 }
}
```

```
package  resma;

public class Cartolina extends Folha{

 public Cartolina(){
 }

 private void tamanhoA0(){
   tamanhoA3();
   tamanhoA4();
   tamanhoA5();
 }
}
```

```
// Uma linha de Comentário

/*
  Um Bloco Comentado
*/
```

Variáveis



- Podem começar com _ ou $
- Não podem ser palavras reservadas

abstract assert boolean break byte case catch char class const continue default do double else enum extends false final finally float for goto if implements import instanceof int interface long native new null package private protected public return short static strictfp super switch synchronized this throw throws transient true try void volatile while

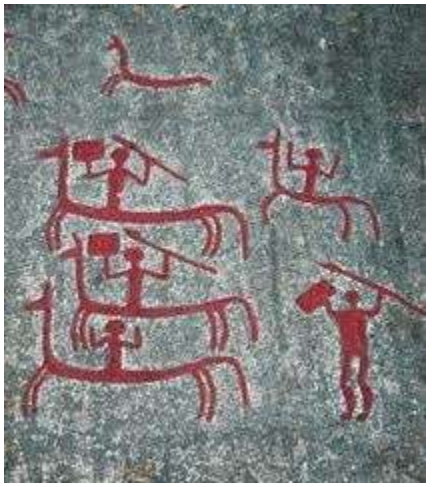
Atribuição

- Atribuir valores às variáveis

```
double precoBatatas=1.99;
boolean portaAberta=true;
int quantidade=4, idade=20;
int cadeiras=quantidade;
float valorReal=12.234F;
```

Tipos de Dados Primitivos

- Inteiro
  - byte
  - short
  - int
  - long
- Vírgula flutuante
  - float
  - double
- Caractere
  - char
- Lógico
  - boolean



Inteiros

Tipo	Bits	Mínimo	Máximo
byte	8	-128	+127
short	16	-32 768	+32 767
int	32	-2 147 483 648	+2 147 483 647
long	64	-9 223 372 036 854 775 808	+9 223 372 036 854 775 807

Vírgula Flutuante

Tipo	Bits	Mínimo	Máximo
float	32	1.4×10^{-45}	$3.4028235 \times 10^{+38}$
double	64	4.9×10^{-324}	$1.7976931348623157 \times 10^{+308}$

A mesma gama de valores para os números negativos

Caractere

Tipo	Bits	Mínimo	Máximo
char	16	0	65 535

‘A’
\x41
\0101
\u0041

65 em decimal

\n	Mudança de Linha
\r	Voltar ao início da linha
\b	Ir uma posição para a esquerda
\t	Um tab
\f	Mudança de página
\’	Um apóstrofo
\”	Uma aspa
\\	A própria barra invertida
\uhhhh	Caractere unicode (0-9, A-F)
\0ooo	Caractere octal (0-7)
\xhh	Caractere hexadecimal (0-9, A-F)

Lógico

Tipo	Bits	Mínimo	Máximo
boolean	8	false	true

Casting

- Permite converter um tipo noutro sem alterar o seu valor binário.
- Pode gerar erros
- Pode ser necessário:
  - Acrescentar bits
  - Remover bits

Operadores

- Matemáticos
- Incremento e Decremento
- Relacionais
- Lógicos

Operadores Matemáticos

\+ adição

\- subtracção

* multiplicação

/ divisão

% resto da divisão inteira

7 % 3 tem como resultado 1

Operadores Incremento e Decremento

++

--

+=

-=

*=

/=

%=

```
++a;          // a=a+1;
a++;          // a=a+1
```

Operação e Atribuição

```
b=++a;        //a=a+1; b=a;
b=a++;        //b=a; a=a+1;

a/=3;         // a=a/3;

a%=5;         //a=a%5;
```

Operadores Relacionais

==

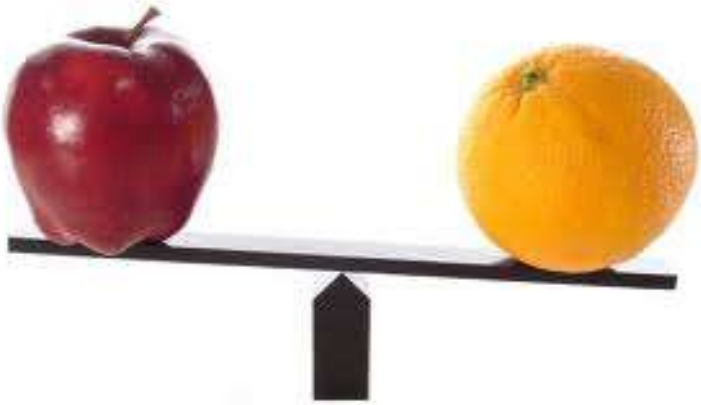
!=

>

<

>=

<=



Operadores Lógicos

true

false

&&

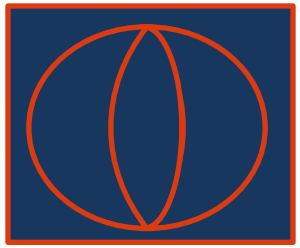
||

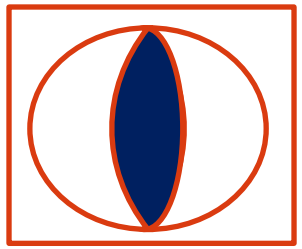
!

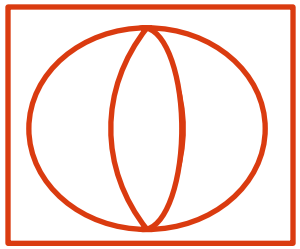
^

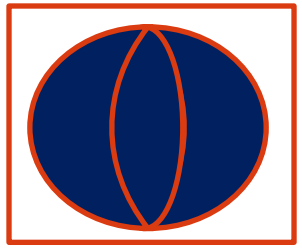
&

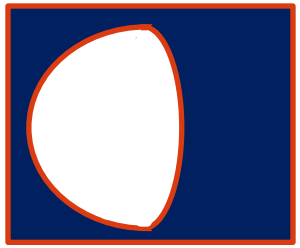
|

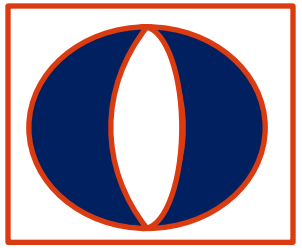






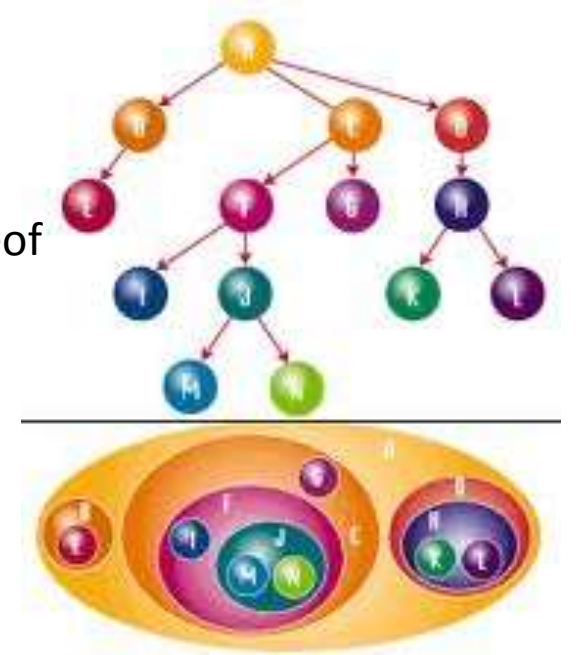




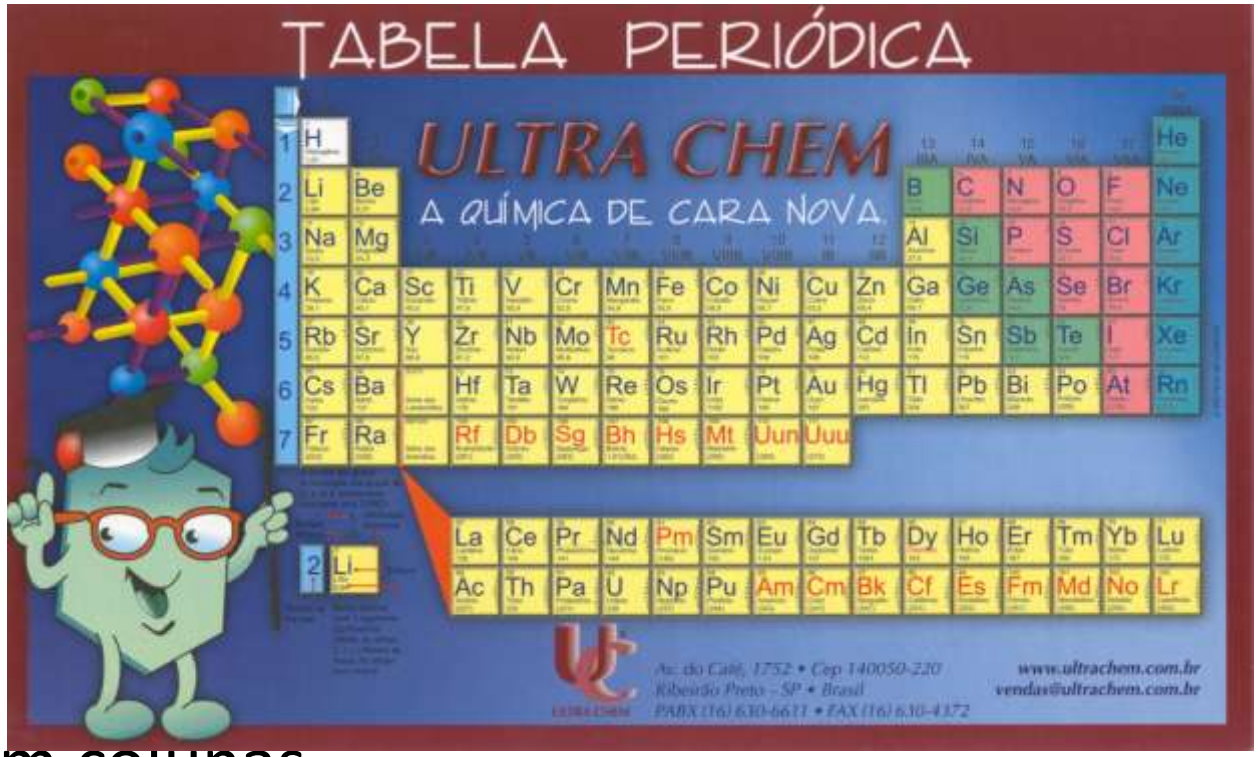


Operadores

Operadores	Precedência
array index, method call, member access	[] () .
postfix	*expr*++ *expr*--
unary	++*expr* --*expr* +*expr* -*expr* ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=



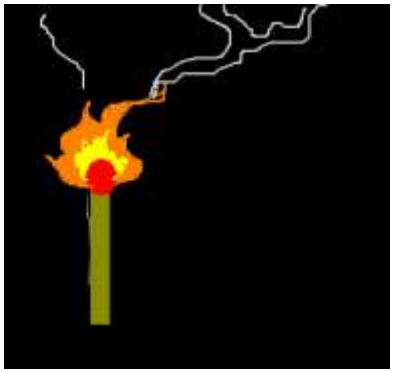
Matrizes



- Tabela
  - De linhas com colunas
  - De colunas com linhas

  int papeis[5][4];

  papeis[3] é o quê?



Ponto de entrada

- C
  int main(int argc, char *argv[])
- Java
  public static void main(String args[])

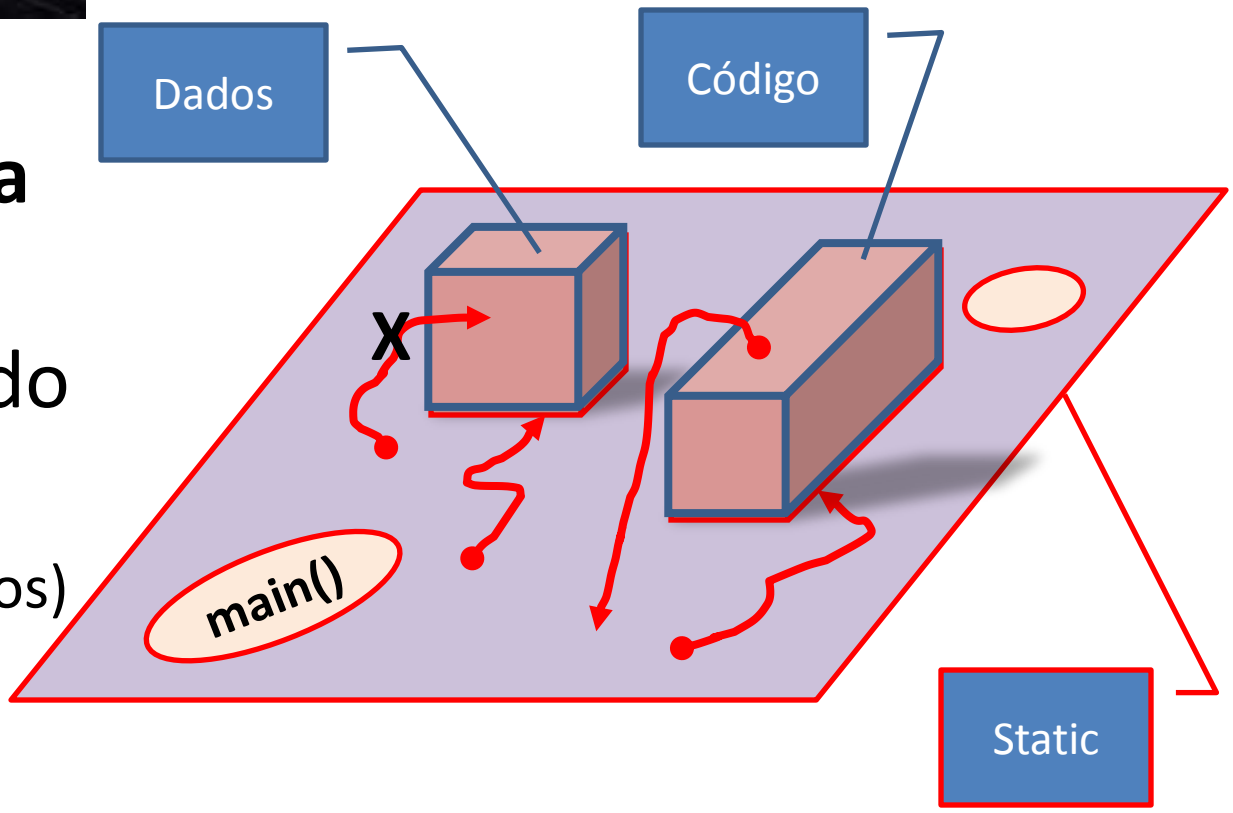
Início

- Executável - .exe
  - O sistema operativo executa uma função init() que abre os ficheiros de entrada e saída (teclado e placa de vídeo) e depois chama
    - C – main()
    - Java – main()
  - Para ser chamado da linha de comando
- Applet - primeira.class
  - O método init() da classe primeira

static main()

- Static -> O **método é da classe** caso contrário é do objecto

(Idem para Atributos)



Instanciar um Objecto

- Criar um objecto.

```
new Integer(3);
new String(“Penso”);
                                    Planeta terra = new Planeta();
Integer tres = 3;
String existo=“eu”;   //  String existo = new String(“eu”);
int bi= new Integer(9999);   // int bi= (int) new Integer(9999);
```

String

- Manipula caracteres que não se pretende serem alterados

String s = “Olá”;

// String s = new String(“Olá”);

s = s + “Bom “ + “Dia”;

// s = new String (s + new String(“Bom ”) + new String(“Dia”));

StringBuffer

- Manipula caracteres que serão alterados durante a execução do programa.

```
StringBuffer b = new StringBuffer();
b.append(“Olá”);
b.append(“Bom ”);
b.append(“Dia.”);
String s = b.toString();
```

.length .length()

- Uma medida do objecto
- Depende do contexto

```
int comp=new String(“ola”).length();
StringBuffer sb = new StringBuffer();
//sb.append(…);
comp = sb.length();
Livro liv = new Livro(“Contos de encantar!”);
comp = liv.length();  //tem de ser feito!
 String a[][] = new String[2][7];
        i=a[2].length;
        i=a[2][3].length();
```

Construtor

- Tem o mesmo nome da Classe
- Define as características iniciais
- Pode haver vários … com o mesmo nome e argumentos diferentes
  - tantos construtores quanto as variedades de objectos

Construtor

```
public class Livro{
        int NumPaginas;

        public Livro(){
           NumPaginas=0;
        }
        public Livro(int numPag){
          …
        }
        public Livro(String titulo){
          …
        }
        public Livro(int altura, int largura, int numPag, String isbn){
          …
        }

        public length(){
            return(NumPaginas);
        }
}
```

```
Livro lb = new Livro();
Livro liv = new Livro(123);
Livro lt = new Livro(23,17, 32, “5603452341”
Livro lj= new Livro(“Java é um objecto ?”);
```

Referências

- http://www.java2s.com/Tutorial/Java/CatalogJava.htm
- http://introcs.cs.princeton.edu/11style/
- http://tektonia.com