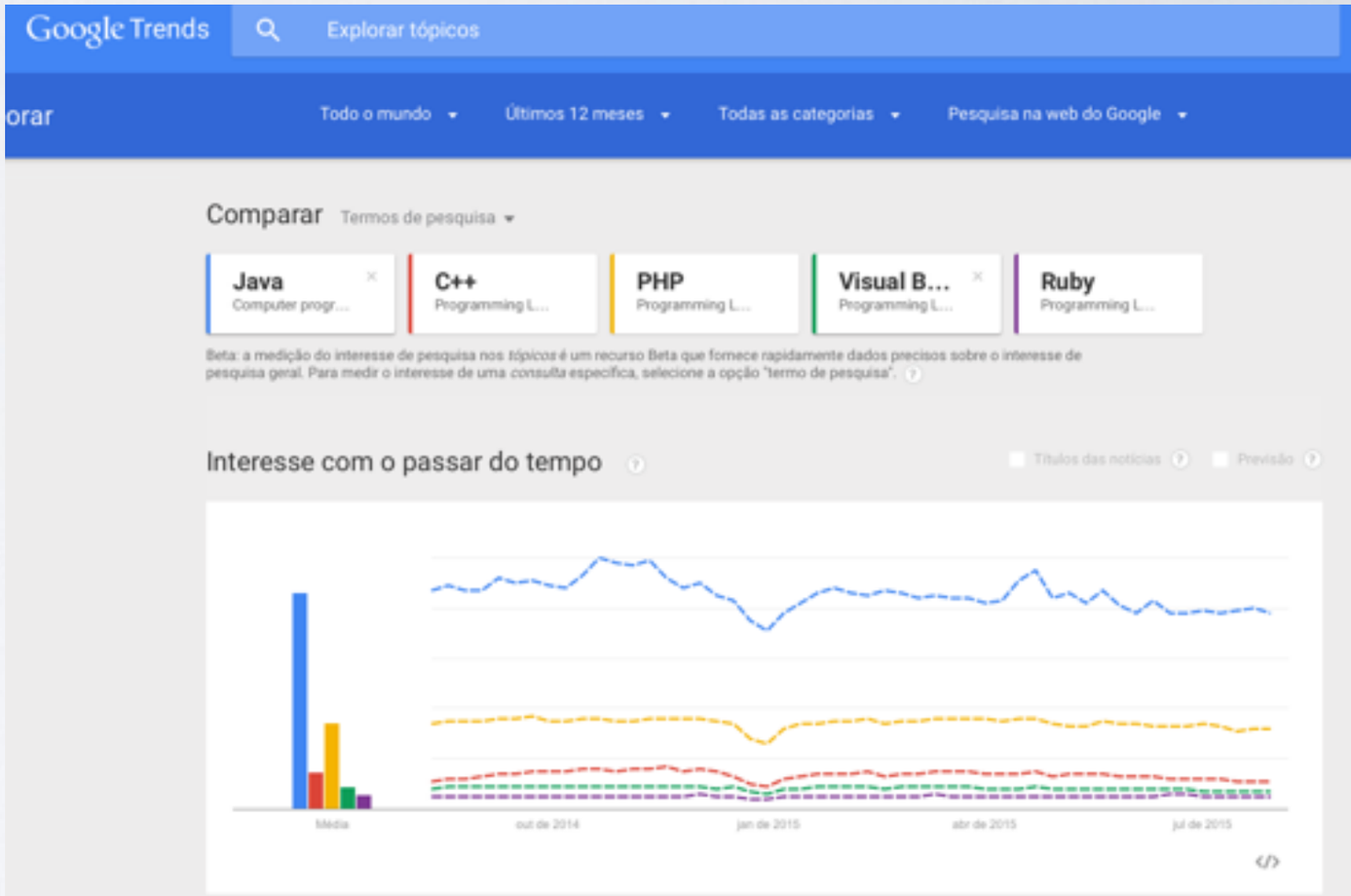
JAVA

Ramon Lummertz

PORQUÊ JAVA

Aug 2016	Aug 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.010%	-0.26%
2	2		C	11.303%	-3.43%
3	3		C++	5.800%	-1.94%
4	4		C#	4.907%	+0.07%
5	5		Python	4.404%	+0.34%
6	7	^	PHP	3.173%	+0.44%
7	9	^	JavaScript	2.705%	+0.54%
8	8		Visual Basic .NET	2.518%	-0.19%
9	10	^	Perl	2.511%	+0.39%
10	12	^	Assembly language	2.364%	+0.60%
11	14	^	Delphi/Object Pascal	2.278%	+0.87%
12	13	^	Ruby	2.278%	+0.86%
13	11	v	Visual Basic	2.046%	+0.26%
14	17	^	Swift	1.983%	+0.80%
15	6	vv	Objective-C	1.884%	-1.31%
16	37	^^	Groovy	1.637%	+1.27%
17	20	^	R	1.605%	+0.60%
18	15	v	MATLAB	1.538%	+0.31%
19	19		PL/SQL	1.349%	+0.21%
20	95	^^	Go	1.270%	+1.19%



PRINCIPIOS DO JAVA

Deve ser simples, orientada a objeto e com sintaxe familiar

Deve ser robusta e segura

Deve possuir uma arquitetura neutra e portátil

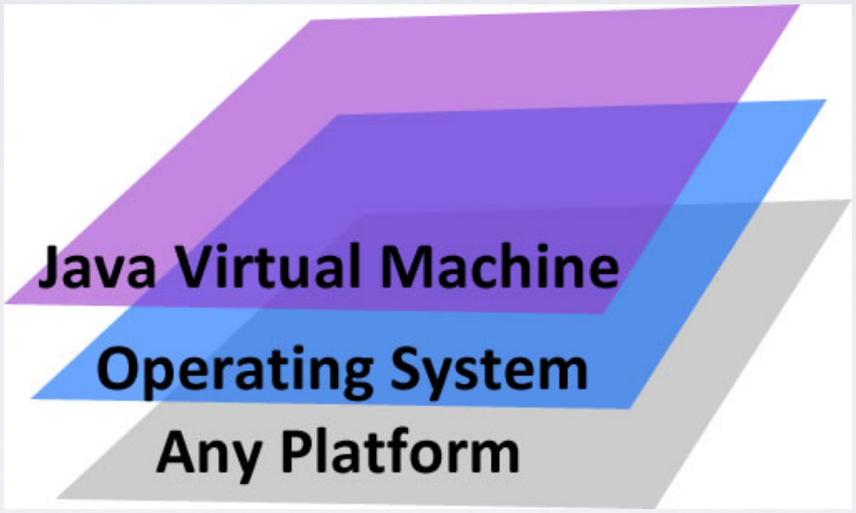
Deve ser executada em alta performance

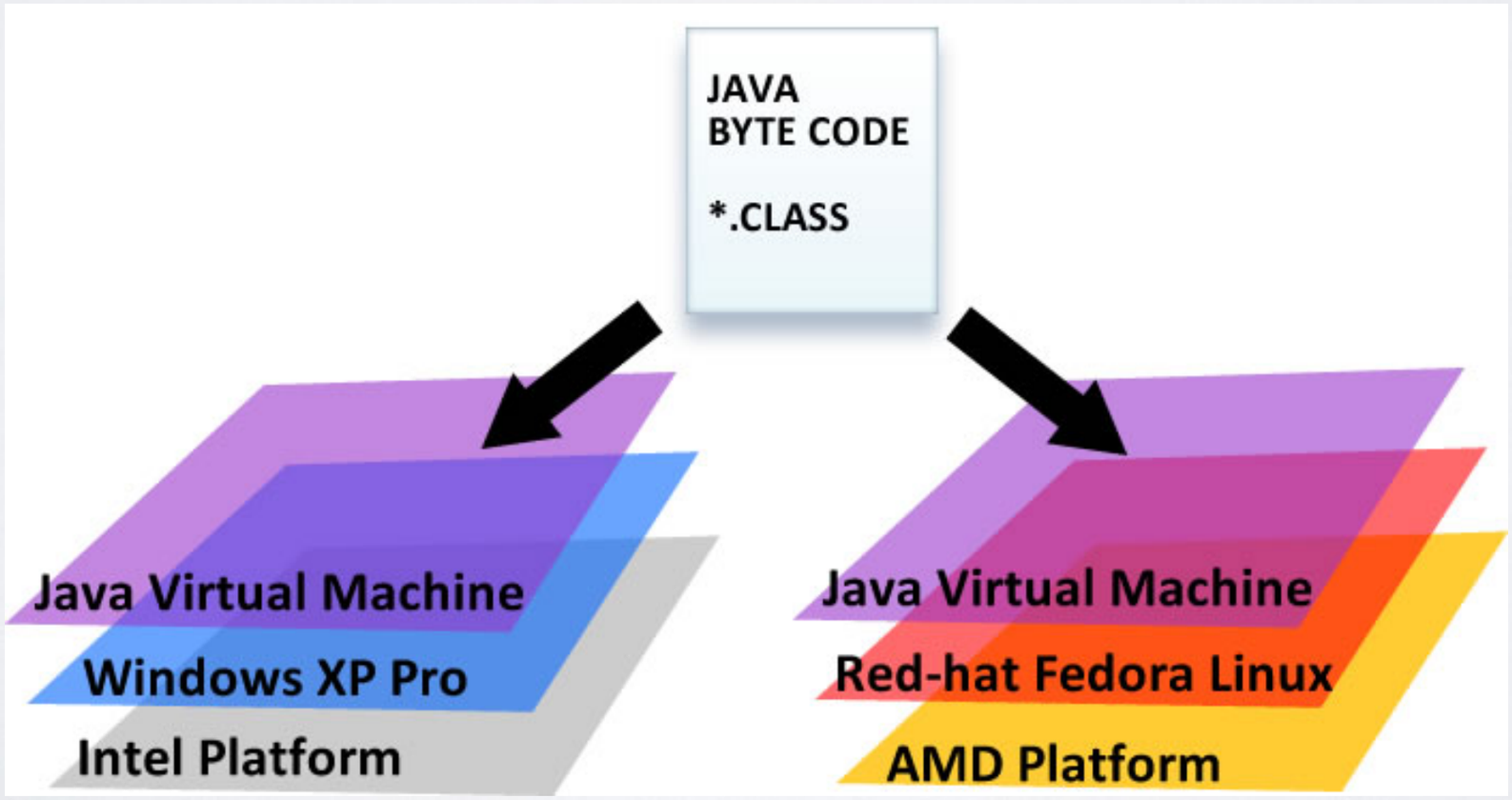
CARACTERISTICAS

JAVA Virtual Machine
Orientadas a Objetos
Garbage Collection
Segurança do código

JVM

JVM Bytecode





ORIENTAÇÃO A OBJETOS

Suporte Nativo pra OO

Tudo é objeto*
* - os tipos primitivos

Polimorfismo
Herança
Encapsulamento

GARBAGER COLLECTION

Gerenciador de limpeza de memória em java

IDENTIFICADORES DE VARIÁVEL

Inicio

a-z, A-Z, _, $

Identificadores permitidos
minhaVariavel
$valor
valorMaximo

Caracteres subsequentes

a-z, A-Z, _, $, 0-9

Identificadores ilegais

12x
3_$
1$dog

HELLOWORD.JAVA

```
 * To change this template use File | Settings | File Templates
 */
public class Hello {
    public static void main(String[] args){
        System.out.println("HELLO");

    }

}
```

OPERAÇÕES

```java
public class Hello {
    public static void main(String[] args){
        int a=21;
        int b=31;

         int c= a + b;
        System.out.print("Resposta é" + c);

    }

}
```

DATA TYPES AND OPERATORS

O QUE SÃO?

Data Types define:
Valores de variável
Operadores

Java = Fortemente tipada(Tipificada)
Possíveis erros : não copila!
Vantagens X Desvantagens

TIPOS DE DADOS

VALORES PRIMITIVOS

Inteiros	Ponto flutuante	Boolean	Caractere
Byte	Float	Boolean	Char
Short	Double		
Int			
long			

Orientado a Objetos

String
Array
Pessoa

TIPOS PRIMITIVOS

		Valores possíveis				
Tipos	***Primitivo***	***Menor***	***Maior***	***Valor Padrão***	***Tamanho***	***Exemplo***
Inteiro	byte	-128	127	0	8 bits	byte ex1 = (byte)1;
	short	-32768	32767	0	16 bits	short ex2 = (short)1;
	int	-2.147.483.648	2.147.483.647	0	32 bits	int ex3 = 1;
	long	-9.223.372.036.854.770.000	9.223.372.036.854.770.000	0	64 bits	long ex4 = 1l;
Ponto Flutuante	float	-1,4024E-37	3.40282347E + 38	0	32 bits	float ex5 = 5.50f;

	double	-4,94E-307	1.79769313486231570E + 308	0	64 bits	double ex6 = 10.20d;
						ou
						double ex6 = 10.20;

Caractere	char	0	65535	\0	16 bits	char ex7 = 194; ou char ex8 = 'a';
Booleano	boolean	false	true	false	1 bit	boolean ex9 = true;

VARIAVEIS E SEU ESCOPO

VARIÁVEIS E SEU ESCOPO

São acessadas no escopo definido por { }
Como:
Classes
Métodos

```java
public class Hello {

    public static void main(String[] args) {
        System.out.println("Hello");

        /// TipoVariavel nomeVariavel
        int var1;
        int var2;
        double var3;
        var1=0;
        var3=2.2;

        if (var1==1){ // escopo de variaveuis são limitadas pelas chaves
            int var4=0;
        } // var4 só é valida dentro das chaves
         System.out.println(var4);

    }
}
```

CONVERSÃO DE TIPOS

Conversão automática:
Tipos compatíveis
Destino ser maior que a origem

```
int a;
float b;
a=10;
b=a;
```

```
short a;
double b;
b=10.7;
a=b;
```

CAST

Casting
(Tipo destino) expressão ;

```
short a;
double b;
b=10.7;
a=(short)b;
```