Processamento da Informação

Ambientes de programação

Prof. Jesús P. Mena-Chalco
CMCC/UFABC

Q2/2018

Apresentação

Apresentação

- **Professor:**

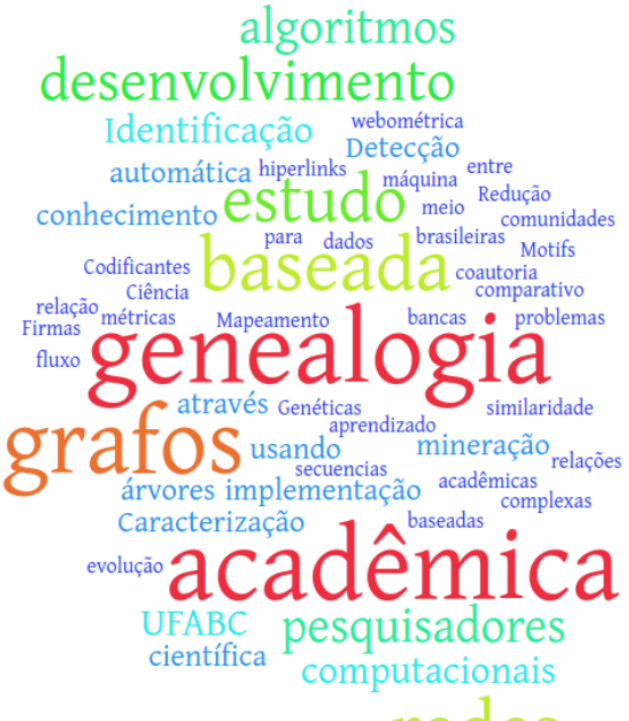
  Jesús P. Mena-Chalco (CMCC)

  jesus.mena@ufabc.edu.br

- **Formação:**
  - Engenheiro da Computação.
  - Mestre e Doutor em Ciência da Computação.
    Instituto de Matemática e Estatística da USP.

- Sala 517-A, torre 2, 5º Andar.

- **Áreas de pesquisa:**
  - Pattern recognition
  - Graph mining
  - Scientometrics/Bibliometrics



Pós-Doutorado			
Laura Cristina Simões Viana	Genealogia acadêmica e sua relação com a constituição e a evolução do conhecimento na Escola Nacional de Saúde Pública Sergio Arouca	2017/Q3	Área: Engenharia do conhecimento - UFABC - ENSP/FIOCRUZ
Doutorado			
Antônio de Abreu Batista Júnior	Identificação automática de pesquisadores especialistas: Uma abordagem baseada em mineração de dados	2017/Q1	Ciência da Computação - UFABC - Bolsa FAPEMA
Julia Ferreira Bernardo (Coorientador)	Mapeamento de pesquisadores difusores da área de prevenção de drogas em contextos educacionais	2017/Q1	Educação e Saúde na Infância e Adolescência - UNIFESP - Bolsa CAPES
Priscilla Labanca	Modelos computacionais preditivos em grafos de genealogia acadêmica	2018/Q1	Ciência da Computação - UFABC
Rafael Jeferson Pezzuto Damaceno	Criação de grafos de genealogia acadêmica: desenvolvimento de metodologias baseadas em mineração de dados	2016/Q1	Ciência da Computação - UFABC - Bolsa CAPES
Luciano Rossi	Os caminhos da ciência por meio do fluxo de conhecimento na perspectiva da genealogia acadêmica	2015/Q3	Ciência da Computação - UFABC - Bolsa CAPES
Mestrado			
Iara Miranda Prates	Algoritmos de simplificação em grafos de genealogia acadêmica	2017/Q1	Ciência da Computação - UFABC - Bolsa CAPES
Rozivaldo Zacarias de Jesus	Caracterização de pesquisadores usando medidas de multidisciplinaridade: Uma abordagem baseada em mineração de publicações acadêmicas	2016/Q3	Ciência da Computação - UFABC
Diogo Fornaziero Segura Ramos	Análise temporal das relações de co-participação em bancas de defesa: uma abordagem baseado em evolução de grafos	2016/Q1	Ciência da Computação - UFABC - Bolsa CAPES
Iniciação científica			
Ana Laura Belotto Claudio (Coorientador)	Identificação de similaridade em disciplinas do catálogo da UFABC	2017-18	Ciência da Computação - UFABC - Bolsa IC
Arthur Veloso Kamienski	Identificando pesquisadores relevantes em grafos de genealogia acadêmica: Uma abordagem baseada em PageRank	2017-18	Ciência da Computação - UFABC - Bolsa FAPESP
Brian Alves Andreossi	Heurísticas computacionais para identificação e redução de potenciais conflitos de interesse em bancas de concursos	2017-18	Ciência da Computação - UFABC
Bruna Pereira Santos	Caracterização de obras literárias usando redes de co-ocorrências	2017-18	Ciência da Computação - UFABC - Bolsa FAPESP
Erick Akio Oti Aoyagui (Coorientador)	Detecção de comunidades em grafos de genealogia acadêmica através de propagação de rótulos	2017-18	Ciência da Computação - UFABC - Bolsa IC
Isabela Maria Biagioni Pedro (Coorientador)	Análise de regiões urbanas brasileiras com base em características da configuração de logradouros	2017-18	Bacharelado em C&T - UFABC - Bolsa IC
Trabalho de conclusão de curso			
Leonardo Nascimento	O impacto da Ciência da Computação nas diferentes áreas do conhecimento a partir da análise de redes brasileiras de coautoria	2018/Q1	Ciência da Computação - UFABC
Lucas Hideki Kimura	Mineração de dados acadêmicos: uma abordagem baseada no catálogo de teses e dissertações da Capes	2018/Q1	Ciência da Computação - UFABC

http://professor.ufabc.edu.br/~jesus.mena/students.html

Aprender a programar: considerações

- Fall in love with mathematics (pratique matemática)
- Be self-motivated (trabalhe com pares)
- Never back down (seja persistente)
- Become a master (ensine aos colegas)
- Be a bookworm (seja leitor ávido)

Leia as seguintes sugestões:
http://www.wikihow.com/Learn-a-Programming-Language

URLs

- **URL:** http://professor.ufabc.edu.br/~jesus.mena/courses
- **Cadastre-se no Tidia4:** http://tidia4.ufabc.edu.br

  Procurar: “PI-2018-Q2-Jesus”

Créditos (T-P-I)=(3-2-5)

Cada disciplina na UFABC é representada por três algarismos: T-P-I

- **T:** Número de horas semanais de aulas expositivas presenciais da disciplina (teóricas)
- **P:** Número médio de horas semanais de trabalho de laboratório, aulas práticas ou aulas de exercícios, realizadas em sala de aula (práticas)
- **I:** Estimativa de horas semanais adicionais de trabalhos necessárias para o bom aproveitamento da disciplina (estudos e trabalhos)

Calendário

Teoria

Laboratório

JUNHO							
Dom	Seg	Ter	Qua	Qui	Sex	Sab	
					1	2	
3	4	5	6	7	8	9	1
10	11	12	13	14	15	16	2
17	18	19	20	21	22	23	3
24	25	26	27	28	29	30	4

JULHO							
Dom	Seg	Ter	Qua	Qui	Sex	Sab	
1	2	3	4	5	6	7	5
8	9	10	11	12	13	14	6
15	16	17	18	19	20	21	7
22	23	24	25	26	27	28	8
29	30	31					9

9 - Revolução Constitucionalista

AGOSTO							
Dom	Seg	Ter	Qua	Qui	Sex	Sab	
			1	2	3	4	9
5	6	7	8	9	10	11	10
12	13	14	15	16	17	18	11
19	20	21	22	23	24	25	12
26	27	28	29	30	31		

20 - Feriado municipal - S B e recesso em S A

Agenda

Aula	Data	Conteúdo previsto
1	07/06	Ambiente(s) de programação
2	14/06	Modularização
3	21/06	Estruturas e seleção simples e composta (if - else)
4	28/06	Repetição (while, for)
5	05/07	Vetores - parte 1
6	12/07	Vetores - parte 2
7	19/07	Prova 1
8	26/07	Matrizes
9	02/08	Caracteres e strings
10	09/08	Exercícios de programação
11	16/08	Exercícios de programação
12	23/08	Prova 2

URL: http://professor.ufabc.edu.br/~jesus.mena/courses/bc0505-2q-2018/

Avaliação

Parte de Teoria: Apresentarei amanhã.

Parte de Laboratório: Duas provas.

- Prova 1 (50%): 19/07/2018
- Prova 2 (50%): 23/08/2018
- Bônus: 2 desafios de programação (10%)

- Prova Substitutiva: 24/08/2018
- Prova de Recuperação: Q3/2018

Avaliação

- **A:** nota ≥ 9
- **B:** 7,5 ≤ nota < 9
- **C:** 6 ≤ nota < 7,5
- **D:** 5,0 ≤ nota < 6
- **F:** nota < 5,0

Nota Teoria	Nota Prática	Conceito
A	A	A
	B	A
	C	B
	D	B
B	A	B
	B	B
	C	B
	D	C
C	A	B
	B	C
	C	C
	D	C
D	A	C
	B	C
	C	D
	D	D

Bibliografia

Bibliografia Básica

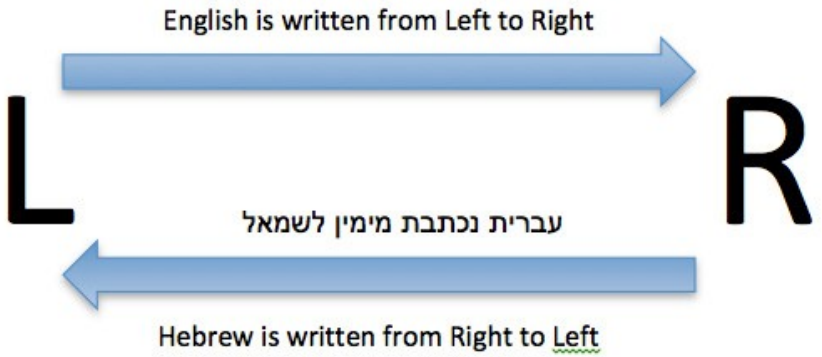
- Forbellone, A. L. V.; Eberspächer, H. F.; Lógica de Programação - A Construção de Algoritmos e Estruturas de Dados; 3ª edição, Editora Pearson Prentice-Hall, 2005
- Sebesta, R. W.; Conceitos de Linguagens de Programação; 5ª edição, Editora Bookman, 2003

Bibliografia Complementar

- Ascensio, A.F.; Campos, E.A., Fundamentos da Programação de Computadores, Pearson, 3a edição, 2012.
- Puga, S., Lógica de programação e estruturas de dados com aplicações em Java, Pearson Prentice-Hall, 2a edição, 2009.

Primeiro programa

Forma de leitura ‘padrão’?



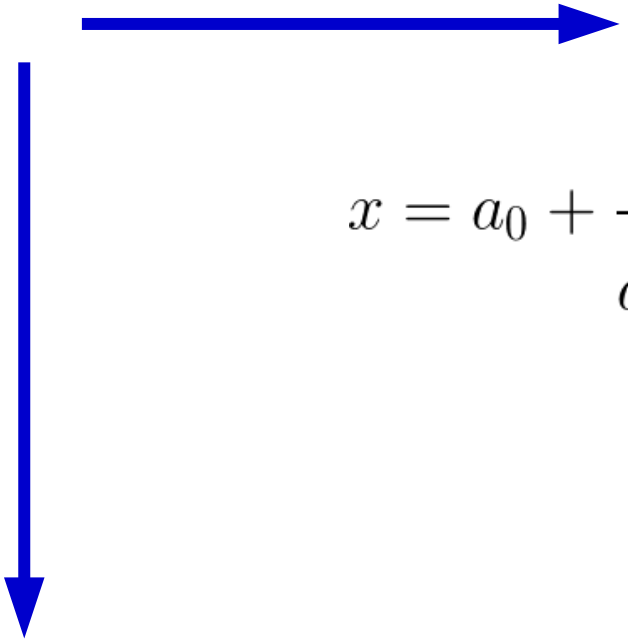
Hebraico

الجامعة الاتحادية ABC
إنها جامعة ممتازة.
البرمجة رائعة.

Árabe

Forma de leitura adotado nesta disciplina

- Instruções em sequência: de cima para baixo
- Leitura de esquerda para direita
- Estruturas aninhadas



$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + a_4}}}$$

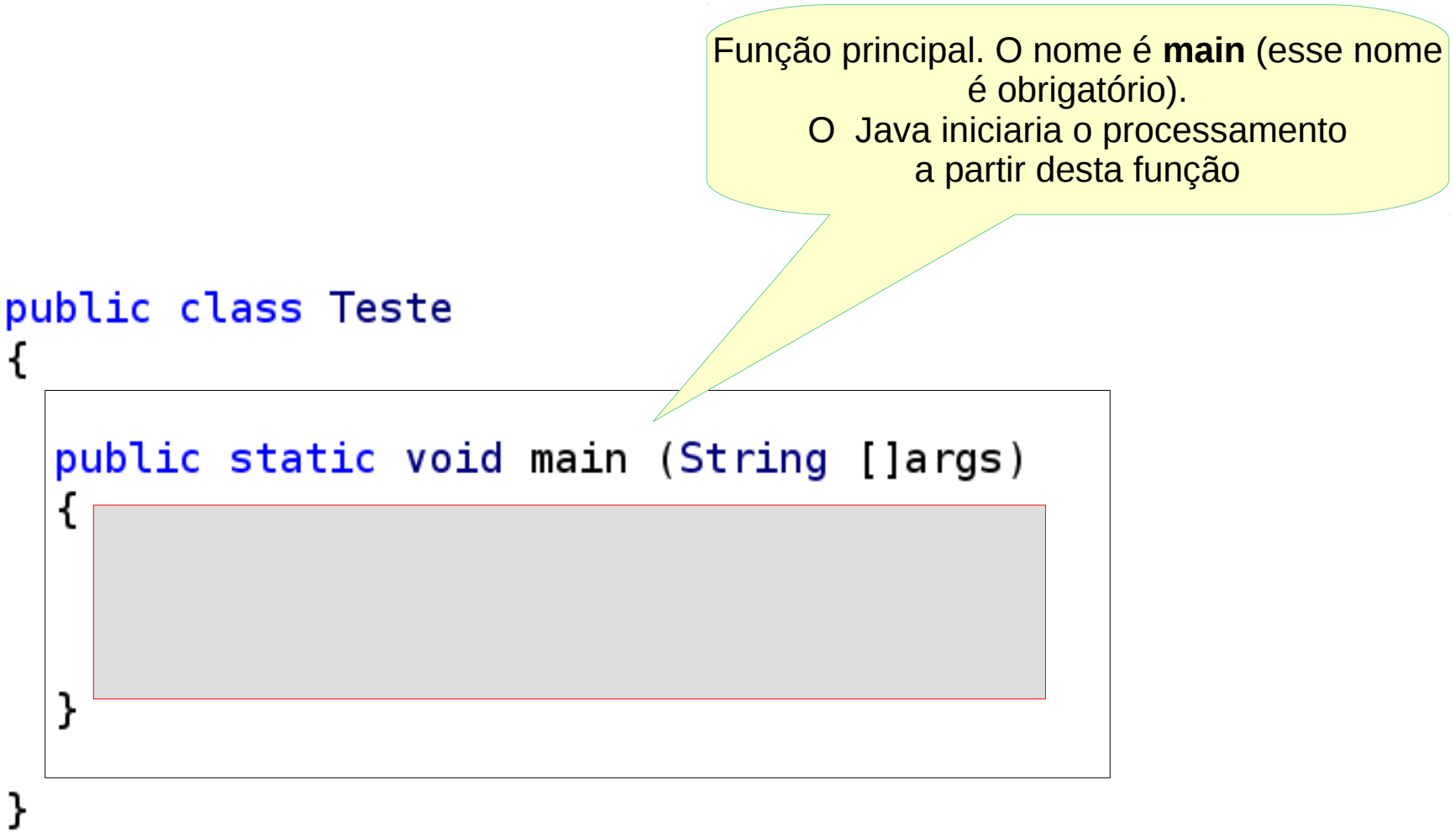
$$x = \frac{(1 * 3)}{(1 - (45 + \sqrt{34}))}$$

Primeiro programa em Java

Base do programa (classe)
Entenda como sendo um suporte
para o programa.
O nome é **Teste** (poderia ser qualquer nome)

```
public class Teste
{
    
}
```

Primeiro programa em Java



Primeiro programa em Java

```
public class Teste
{

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


   }

}
```

As chaves ({}) são obrigatórias.
A indentação (espaçamento horizontal) é recomendado para a leitura do humano)

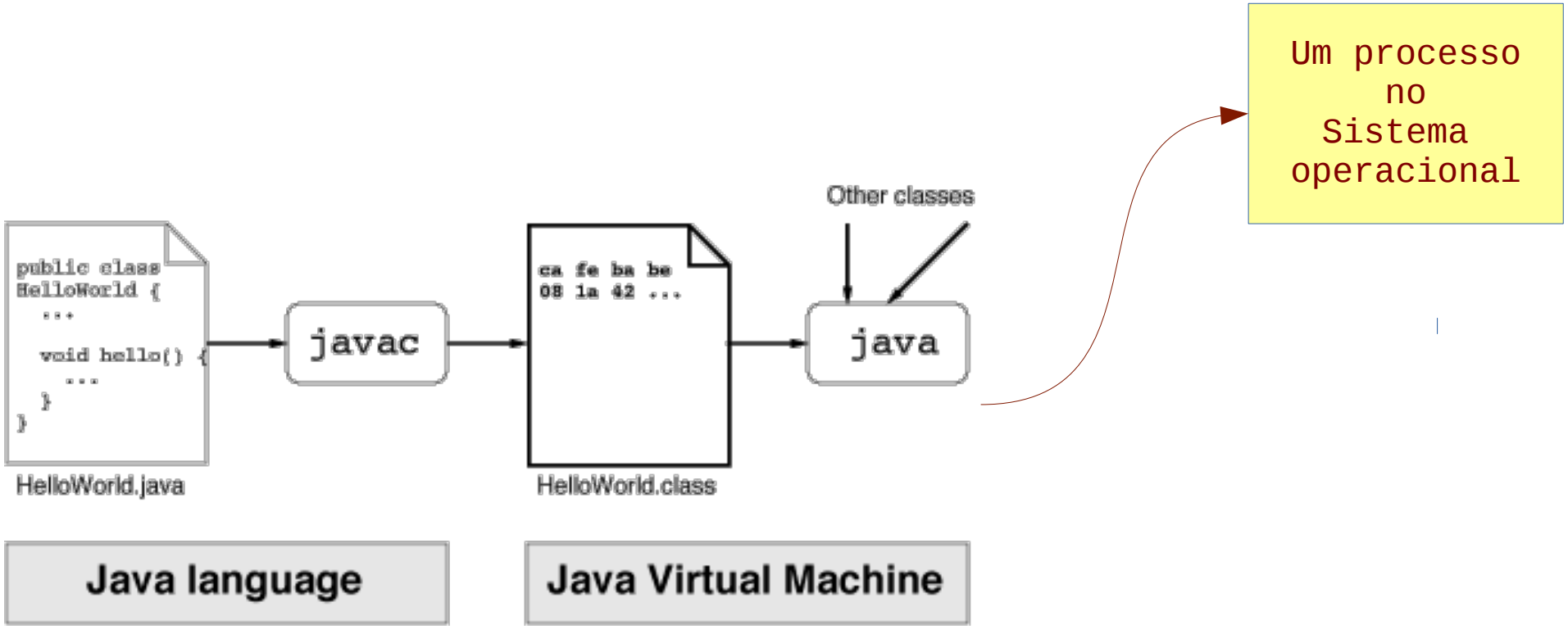
Formato do arquivo: .java

O programa é, na verdade, um arquivo de **texto**, mas é necessário ter o **sufixo .java**

Motivo: conveniência

Quando o arquivo de texto vira, de fato, em programa?

Compilador: javac (Java Compiler)



- Podemos fazer todo esse procedimento de forma **manual**.
- Mas existem **ambientes de programação** que auxiliam a tarefa.

Ambientes de programação

Ambientes de programação

Um ambiente de programação oferece **ferramentas** de gerenciamento e desenvolvimento de software.

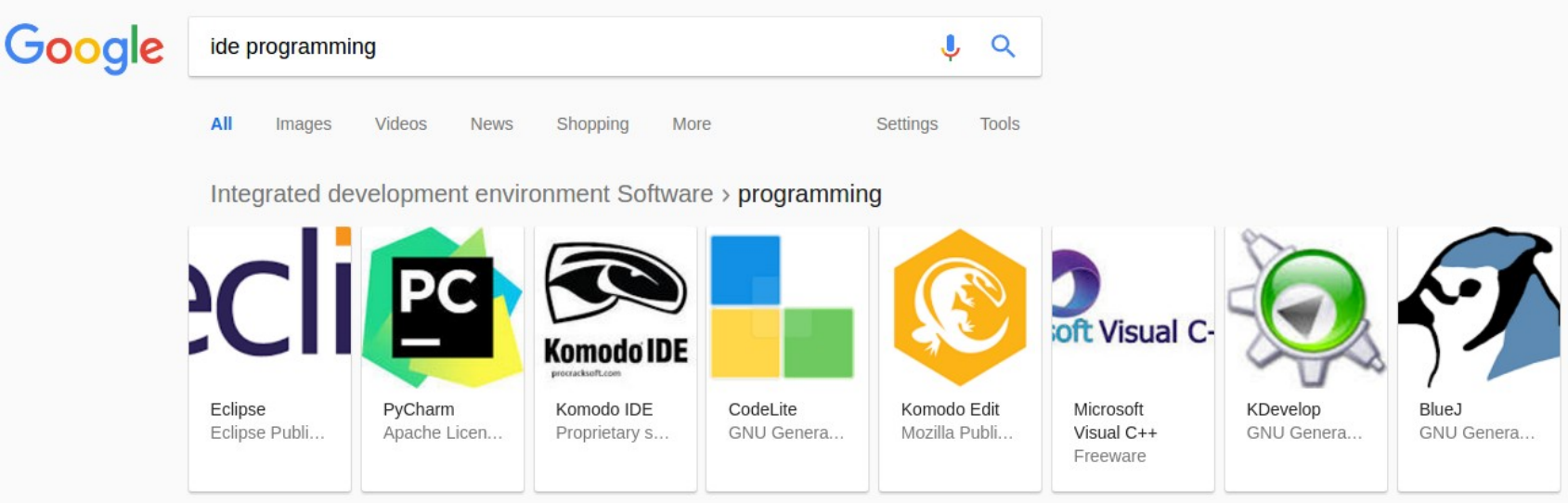
O objetivo é **agilizar** o desenvolvimento.

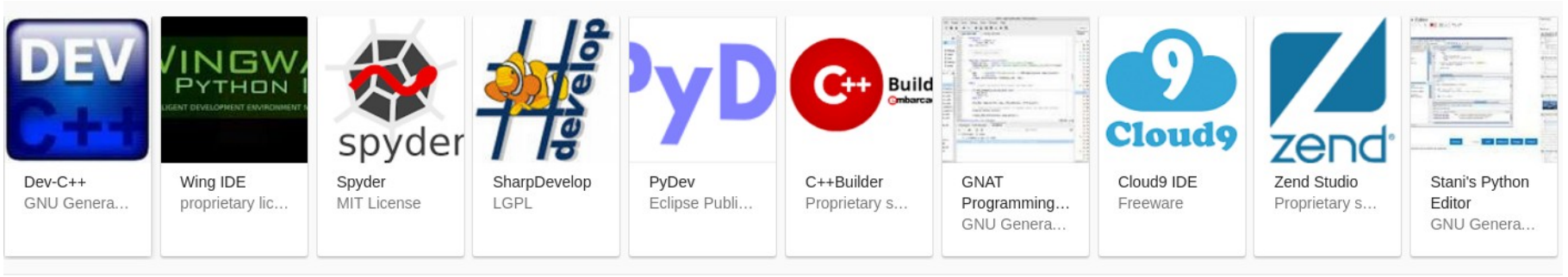
Um *Integrated Development Environment* (IDE) reune as características básicas para um desenvolvimento rápido de software.

Ferramentas comuns:

- Editor de código (editor de texto)
- Compilador
- Depurador

Qual IDE utilizar?





Ambientes de programação

IDE 1

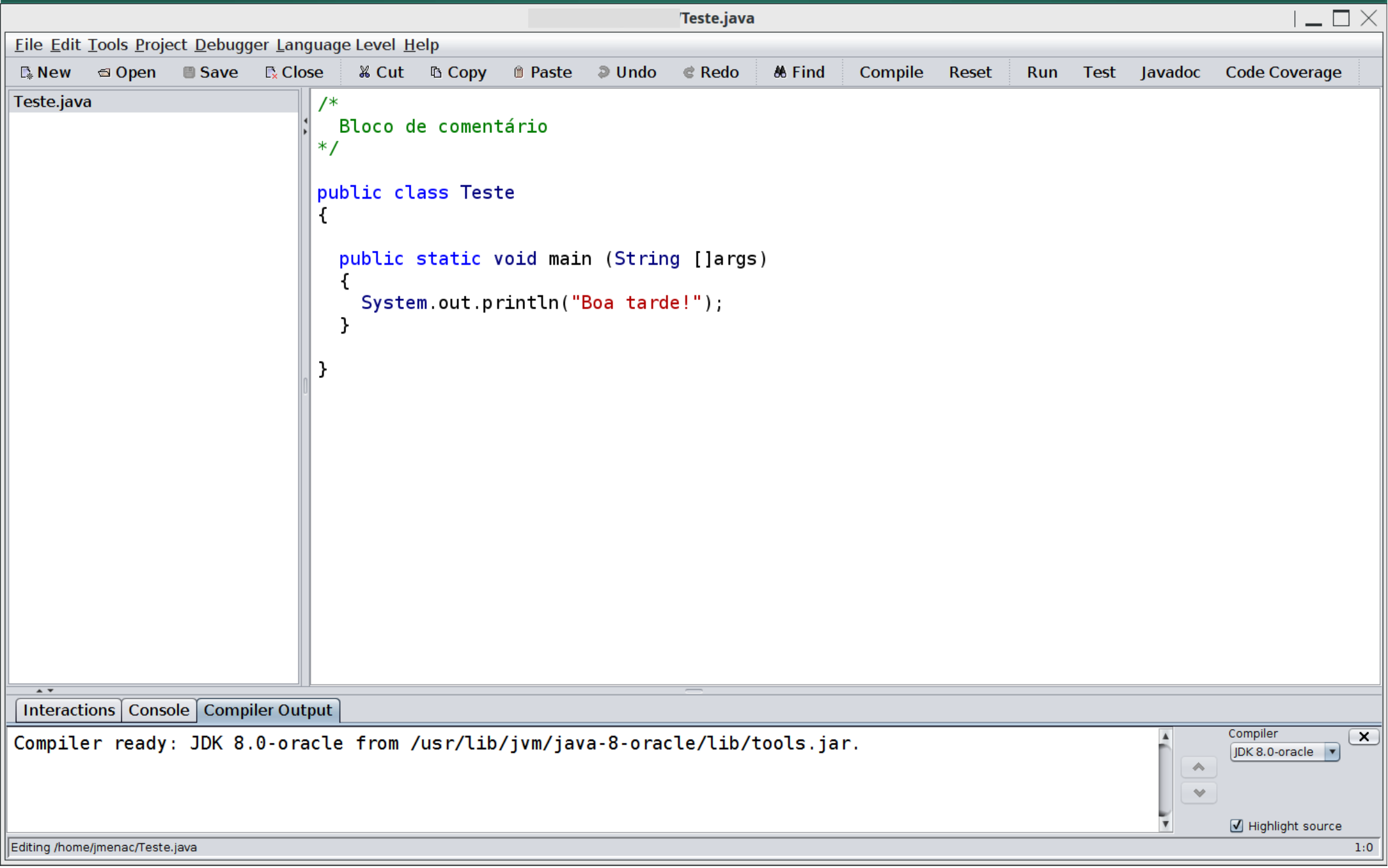
IDE 2

Em laboratório usaremos inicialmente IDEs ‘simples’

(1) Repl.it

https://repl.it/languages/java

(2) DrJava

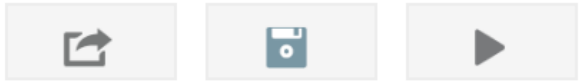

Teste.java
File Edit Tools Project Debugger Language Level Help
New Open Save Close Cut Copy Paste Undo Redo Find Compile Reset Run Test Javadoc Code Coverage
Teste.java
/*
Bloco de comentário
*/
public class Teste
{
public static void main (String []args)
{
System.out.println("Boa tarde!");
}
}
Interactions Console Compiler Output
Compiler ready: JDK 8.0-oracle from /usr/lib/jvm/java-8-oracle/lib/tools.jar.
Compiler
JDK 8.0-oracle
Highlight source
Editing /home/jmenac/Teste.java
1:0

Inteiros

Reais

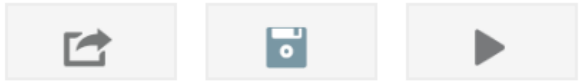
Data Type	Bits	Range
byte	8	-2^7 to 2^7-1 (-128 to 127)
short	16	-2^{15} to $2^{15}-1$ (-32,768 to 32,767)
int	32	-2^{31} to $2^{31}-1$ (-2,147,483,648 to 2,147,483,647)
long	64	-2^{63} to $2^{63}-1$ (-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807)
float	32	1.401298464324817070e-45 to 3.402823466385288600e+38 (positive or negative) (single-precision 32-bit IEEE 754 floating point)
double	64	4.940656458412465440e-324d to 1.797693134862315700e+308d (positive or negative) (double-precision 64-bit IEEE 754 floating point)
char	16 (unsigned)	0 to 65,535
boolean	1	true, false

Teste



```
class Main {

  public static void main(String[] args) {

    int t;
    t = 1 + 10;
    System.out.println("t");

  }

}
```

Teste



```
class Main {

  public static void main(String[] args) {

    int t;
    t = 1 + 10;
    System.out.println(t);

  }

}
```

Palavras “reservadas” que não podem ser usadas como nomes de variáveis

Java Keywords				
abstract	boolean	break	byte	case
catch	char	class	continue	default
do	double	else	extends	false
final	finally	float	for	if
implements	import	instanceof	int	interface
long	native	new	null	package
private	protected	public	return	short
static	super	switch	synchronized	this
throw	throws	transient	true	try
void	volatile	while		
Keywords that are reserved but not used by Java				
const	goto			