

Curso Programação Orientada a Objetos com Java

Capítulo: Apresentação da linguagem Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Objetivos

- Contextualizar Java
- Conhecer sintaxe e recursos básicos da linguagem Java
 - Estrutura de uma aplicação Java
 - Tipos básicos
 - Entrada e saída
 - Operadores
 - Estruturas de controle
- Aprender a utilização básica do Eclipse

Java - contextualização

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

O que é Java?

- Linguagem de programação (regras sintáticas)
- Plataforma de desenvolvimento e execução
 - Bibliotecas (API)
 - Ambientes de execução

Histórico

- Problemas resolvidos e motivo de seu sucesso:
 - Ponteiros / gerenciamento de memória
 - Portabilidade falha: reescrever parte do código ao mudar de SO
 - Utilização em dispositivos diversos
 - Custo
- Criada pela Sun Microsystems no meio da década de 1990
- Adquirida pela Oracle Corporation em 2010



Aspectos notáveis

- Código compilado para bytecode e executado em máquina virtual (JVM)
- Portável, segura, robusta
- Roda em vários tipos de dispositivos
- Domina o mercado corporativo desde o fim do século 20
- Padrão Android por muitos anos



Edições

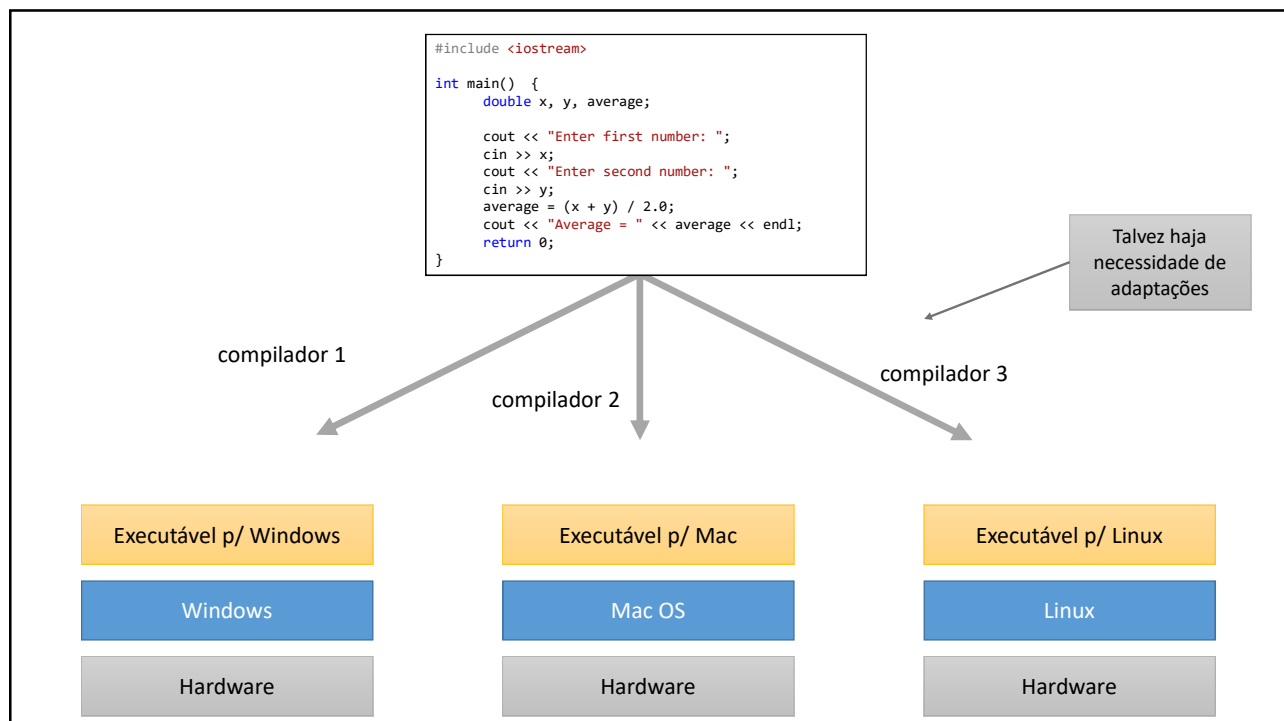
- Java ME - Java Micro Edition - dispositivos embarcados e móveis - IoT
 - <http://www.oracle.com/technetwork/java/javame>
- Java SE - Java Standard Edition - core - desktop e servidores
 - <http://www.oracle.com/technetwork/java/javase>
 - https://en.wikipedia.org/wiki/Java_version_history
 - JavaFX - desktop e RIA
- Java EE - Java Enterprise Edition - aplicações corporativas
 - <http://www.oracle.com/technetwork/java/javaee>
 - https://en.wikipedia.org/wiki/Java_EE_version_history

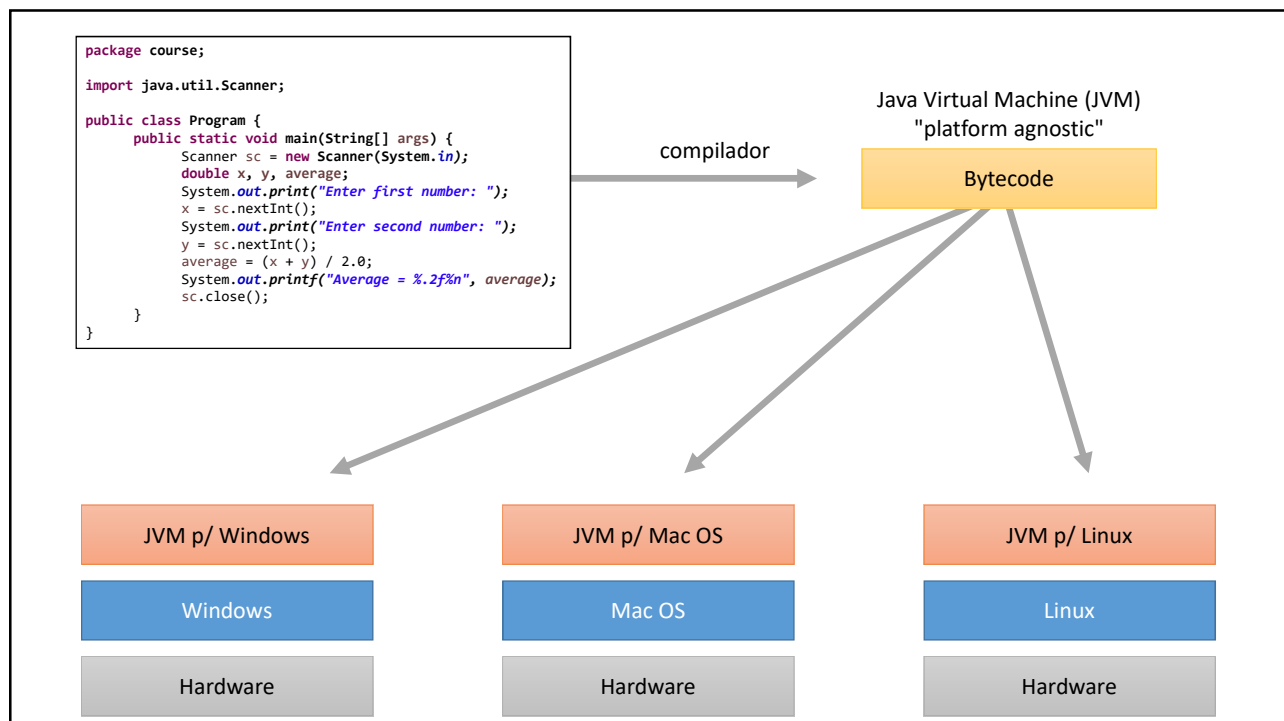
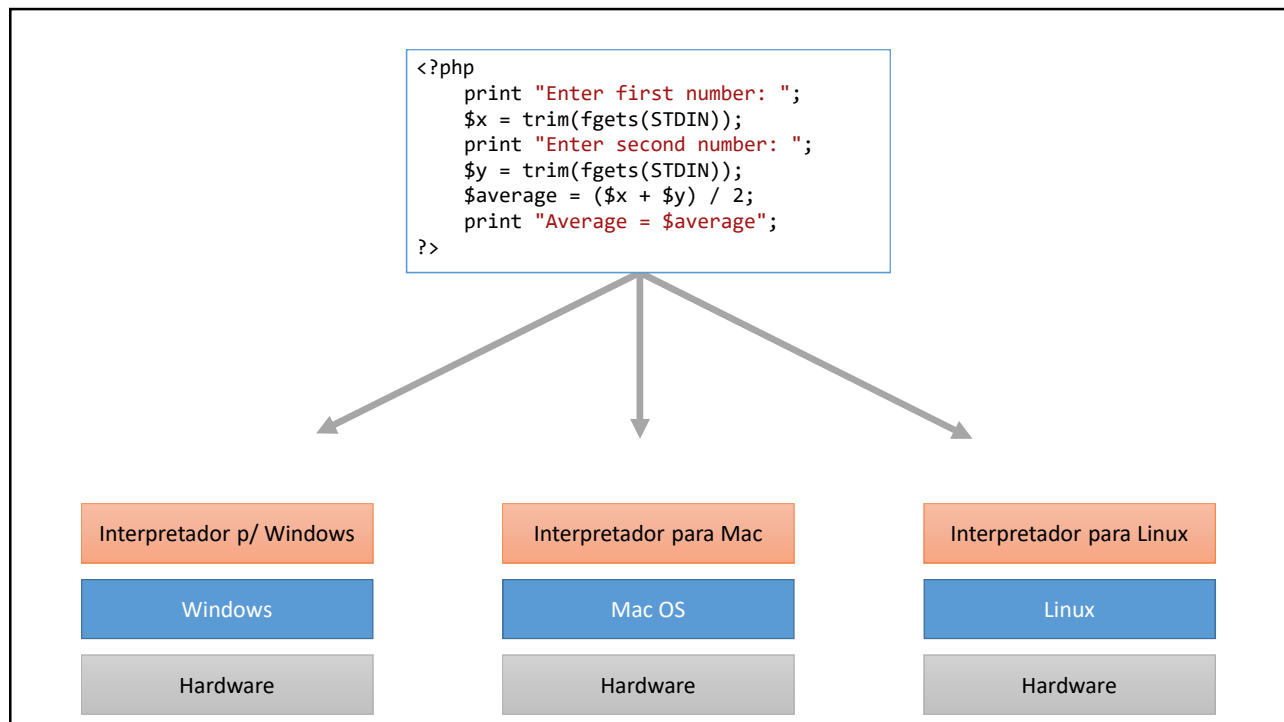
Plataforma Java SE

- Bibliotecas - API specification
 - <https://docs.oracle.com/javase/8/docs/api/>
 - <https://docs.oracle.com/javase/10/docs/api/>
- JVM - Java Virtual Machine
 - Máquina virtual do Java - necessário para executar sistemas Java
- <http://www.oracle.com/technetwork/java/javase/downloads>
 - JRE - Java Runtime Environment
 - Necessário para usuários finais executarem aplicações Java no computador
 - Server JRE
 - Necessário para executar sistemas Java em servidores
 - JDK - Java Development Kit
 - Necessário para desenvolvedores Java. Contém o JRE, mais ferramentas para desenvolvimento, depuração e monitoramento de projetos Java.

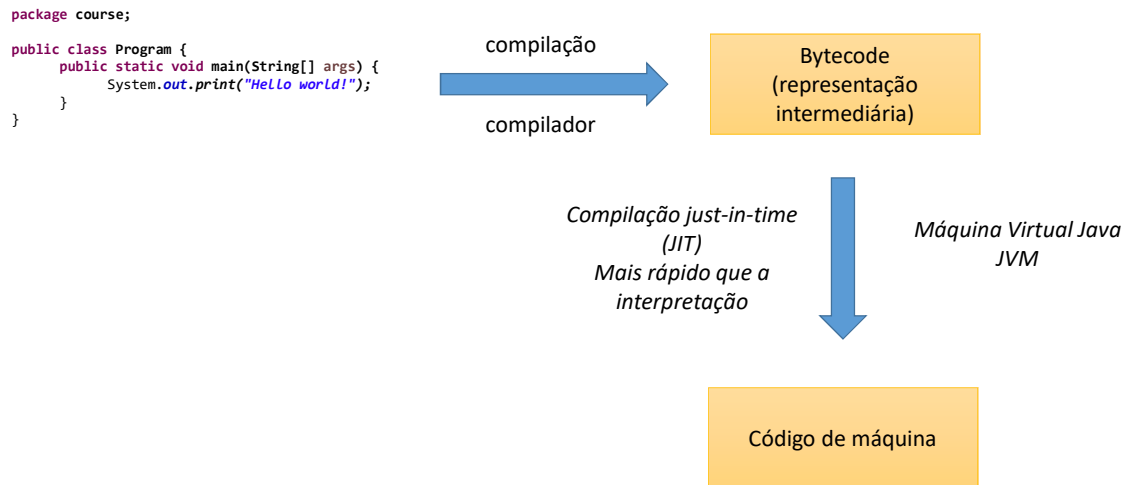
Compilação e interpretação

- Linguagens **compiladas**: C, C++
- Linguagens **interpretadas**: PHP, JavaScript
- Linguagens **pré-compiladas + máquina virtual**: Java, C#





Modelo de execução

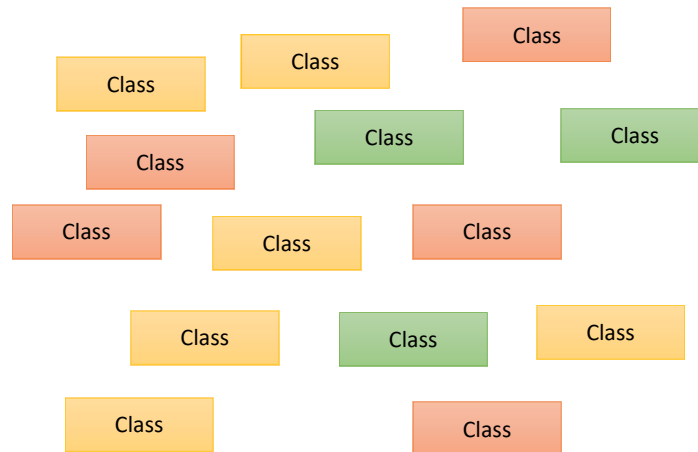


Estrutura de uma aplicação Java

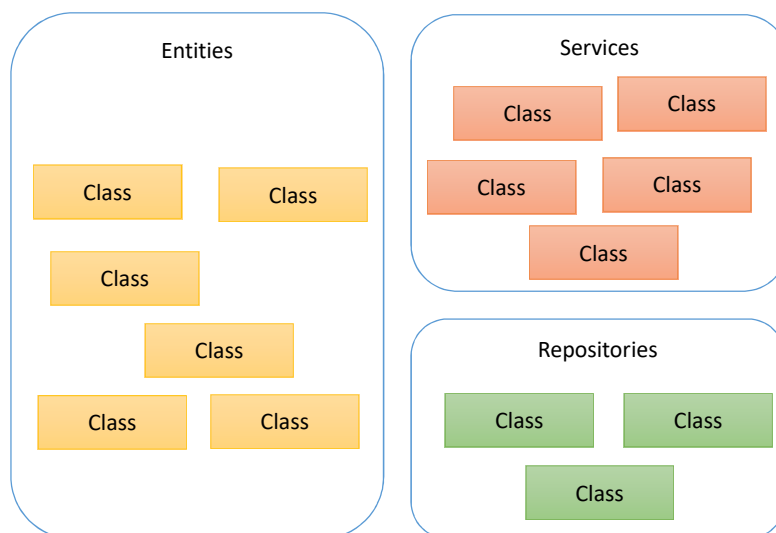
<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

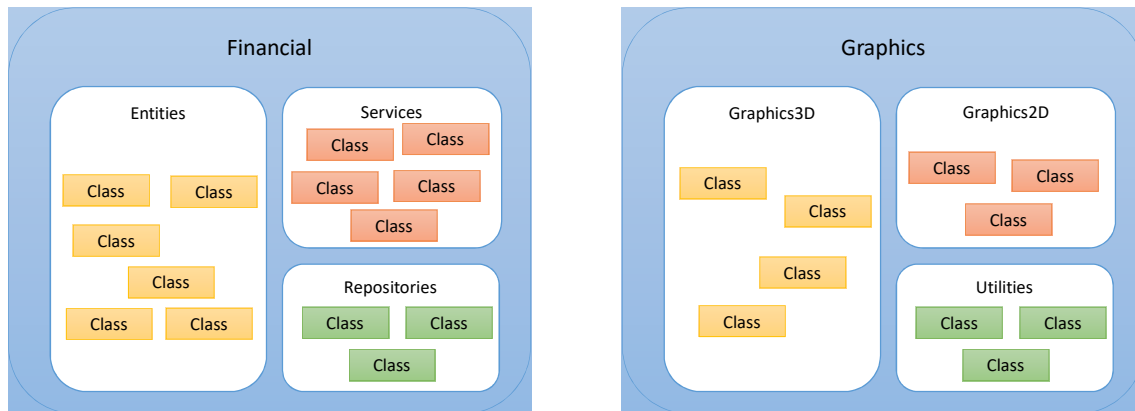
Uma aplicação é composta por classes



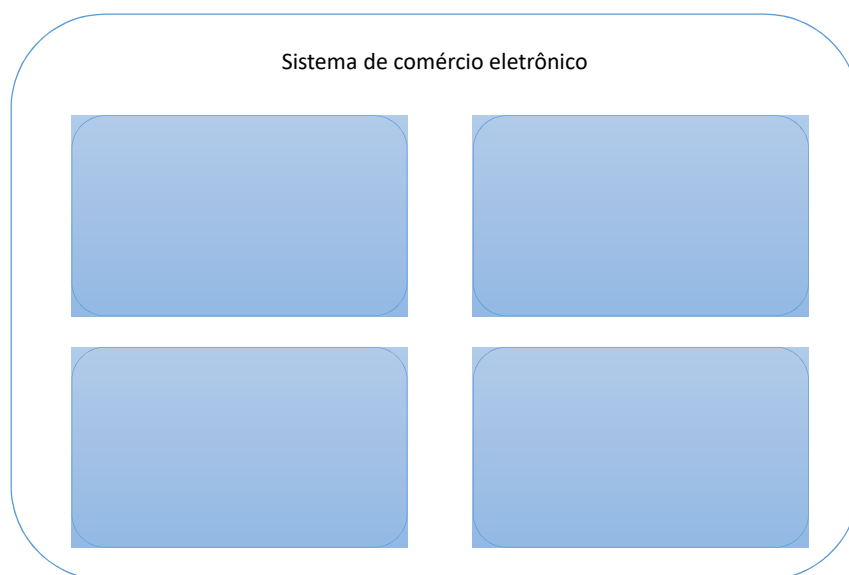
package = agrupamento LÓGICO de classes relacionadas



Módulo (Java 9+) = Agrupamento lógico de pacotes relacionados
Runtime = Agrupamento físico



Aplicação = Agrupamento de módulo relacionados



Instalação do Java e Eclipse no Windows

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Certifique-se de que seu Windows esteja devidamente licenciado e atualizado
 - Windows update
- Baixar e instalar o Java JDK
 - <http://www.oracle.com/technetwork/java/javase/downloads>
- Configurar variáveis de ambiente do sistema
 - Painel de Controle -> Variáveis de Ambiente
 - Variável JAVA_HOME: C:\Program Files\Java\jdk-10.0.1
 - Variável Path: **incluir** C:\Program Files\Java\jdk-10.0.1\bin
 - Testar no terminal de comando: java --version
- Baixar e descompactar o Eclipse
 - <https://www.eclipse.org/downloads/eclipse-packages/>
 - Testar: rodar o Eclipse e escolher um "workspace" (pasta onde você vai salvar seus projetos)

Primeiro programa em Java no Eclipse

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Window -> Perspective -> Open Perspective -> Java
- File -> New -> Java Project
- Package Explorer
 - JRE System Library
 - Pasta "src"
- Criar classe
 - Botão direito no projeto -> New -> Class
 - Escolher um nome para a classe (iniciar com letra Maiúscula)
 - Escolher um nome para o pacote (iniciar com letra Minúscula)
 - Selecionar public static void main
- Classe
 - package
 - Cláusula import: referências a outros pacotes
 - Classe
 - Método
 - public static void main(String[] args)
- Executar o programa: Botão direito no arquivo -> Run As -> Java Application
- Como fechar e reabrir o projeto? E o workspace?
- Dica: indentação automática: CTRL + SHIFT -> F

Trabalhando sem IDE

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Passos

- Crie a estrutura do projeto
- `javac course/*.java`
- `java course/Program`

Nota: para projetos grandes, usar uma ferramenta para automatização (Maven, Ant, Gradle).

Tipos primitivos em Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

- Java é uma linguagem **estaticamente tipada**
- Tipos primitivos em Java
- Tipos referência String e Object
- Variável não iniciada
- Valores padrão
- Overflow
- Padrão para float: sufixo "f"
- Padrão para char: aspas simples
- Padrão para string: aspas duplas
- Padrão para boolean: true, false
- Opção: inferência de tipos com palavra "var" (dentro de métodos)

Tipos primitivos em Java (tipo valor)

Type	Contains	Default	Size	Range
boolean	true or false	false	1 bit	NA
char	Unicode character	\u0000	16 bits	\u0000 to \uFFFF
byte	Signed integer	0	8 bits	-128 to 127
short	Signed integer	0	16 bits	-32768 to 32767
int	Signed integer	0	32 bits	-2147483648 to 2147483647
long	Signed integer	0	64 bits	-9223372036854775808 to 9223372036854775807
float	IEEE 754 floating point	0.0	32 bits	$\pm 1.4\text{E}-45$ to $\pm 3.4028235\text{E}+38$
double	IEEE 754 floating point	0.0	64 bits	$\pm 4.9\text{E}-324$ to $\pm 1.7976931348623157\text{E}+308$

String e Object

Tipo	Descrição
String	Uma cadeia de caracteres Unicode IMUTÁVEL (<i>segurança, simplicidade, thread safe</i>)
Object	Um objeto genérico (toda classe em Java é subclasse de object) getClass equals hashCode toString

Demo

```

boolean completed = false;
char gender = 'F';
char letter = '\u0041';
byte n1 = 126;
int n2 = 1000;
int n3 = 2147483647;
long n4 = 2147483648L;
float n5 = 4.5f;
double n6 = 4.5;
String name = "Maria Green";
Object obj1 = "Alex Brown";
Object obj2 = 4.5f;

System.out.println(completed);
System.out.println(gender);
System.out.println(letter);
System.out.println(n1);
System.out.println(n2);
System.out.println(n3);
System.out.println(n4);
System.out.println(n5);
System.out.println(n6);
System.out.println(name);
System.out.println(obj1);
System.out.println(obj2);

```

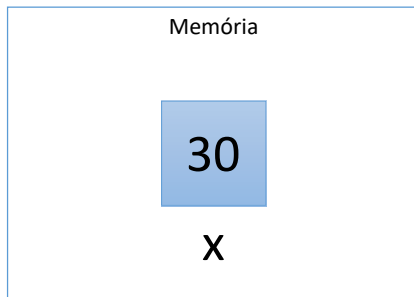
Valores padrão

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	"\u0000"
String (or any object)	null
boolean	false

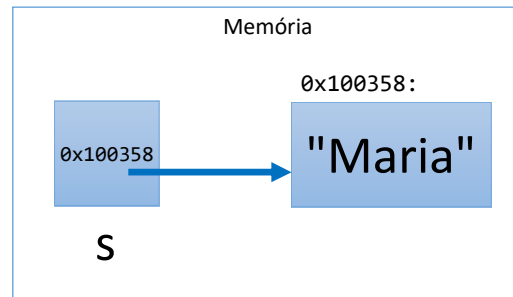
*Quando criamos um **array** ou um **objeto** com atributos desses tipos, esses são os valores atribuídos inicialmente aos dados.*

Tipo valor vs. tipo referência

```
int x = 30;
```



```
String s = "Maria";
```



Restrições e convenções para nomes

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Restrições para nomes de variáveis

- Não pode começar com dígito: use uma letra ou _
- Não usar acentos ou til
- Não pode ter espaço em branco
- Sugestão: use nomes que tenham um significado

Errado:

```
int 5minutes;  
int salário;  
int salario do funcionario;
```

Correto:

```
int _5minutes;  
int salario;  
int salarioDoFuncionario;
```

Convenções

- Camel Case: lastName
 - pacotes
 - atributos
 - métodos
 - variáveis e parâmetros
- Pascal Case: ProductService
 - classes

```
package entities;

public class Account {

    private String holder;
    private Double balance;

    public Account(String holder, Double balance) {
        this.holder = holder;
        this.balance = balance;
    }

    public String getHolder() {
        return holder;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public void withdraw(double amount) {
        balance -= amount;
    }
}
```

Conversão implícita e casting

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Conversão implícita entre tipos
- Casting: conversão explícita entre tipos COMPATÍVEIS

Exemplo 1

```
double a;  
float b;  
  
a = 5.0;  
b = (float)a;  
  
System.out.println(b);
```

Exemplo 2

```
double a;  
int b;  
  
a = 5.0;  
b = (int)a;  
  
System.out.println(b);
```

Exemplo 3

```
int a, b;  
double result;  
  
a = 5;  
b = 2;  
  
result = (double) a / b;  
  
System.out.println(result);
```

Saída de dados em Java

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Imprimir na saída padrão (console)

- Comandos
 - `System.out.println`
 - `System.out.print`
 - `System.out.printf`

Demo

```
char gender = 'F';
int age = 32;
double balance = 10.35784;
String name = "Maria";

System.out.print("Good morning!");
System.out.println("Good afternoon!");
System.out.println("Good night!");
System.out.println("-----");
System.out.println(balance);
System.out.printf("%.2f\n", balance);
System.out.printf("%.4f\n", balance);
Locale.setDefault(Locale.US);
System.out.printf("%.4f\n", balance);
System.out.println("-----");
System.out.printf("%s is %d years old, gender %c, and got balance = %.2f bitcoins\n", name, age,
gender, balance);
```

Exercício de fixação

Em um novo programa, inicie as seguintes variáveis:

```
String product1 = "Computer";
String product2 = "Office desk";

int age = 30;
int code = 5290;
char gender = 'F';

double price1 = 2100.0;
double price2 = 650.50;
double measure = 53.234567;
```

Em seguida, usando os valores das variáveis, produza a seguinte saída na tela do console:

```
Products:
Computer, which price is $ 2.100,00
Office desk, which price is $ 650,50

Record: 30 years old, code 5290 and gender: F

Measure with eight decimal places: 53,23456700
Rounded (three decimal places): 53,235
US decimal point: 53.235
```

(correção na próxima página)

```

package course;

import java.util.Locale;

public class Program {

    public static void main(String[] args) {

        String product1 = "Computer";
        String product2 = "Office desk";

        byte age = 30;
        int code = 5290;
        char gender = 'F';

        double price1 = 2100.0;
        double price2 = 650.50;
        double measure = 53.234567;

        System.out.println("Products:");
        System.out.printf("%s, which price is $ %.2f%n", product1, price1);
        System.out.printf("%s, which price is $ %.2f%n", product2, price2);
        System.out.println();
        System.out.printf("Record: %d years old, code %d and gender: %c%n", age, code, gender);
        System.out.println();
        System.out.printf("Measure with eight decimal places: %.8f%n", measure);
        System.out.printf("Rounded (three decimal places): %.3f%n", measure);
        Locale.setDefault(Locale.US);
        System.out.printf("US decimal point: %.3f%n", measure);
    }
}

```

Date

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Date

Representa um INSTANTE

Pacote java.util

<https://docs.oracle.com/javase/10/docs/api/java/util/Date.html>

Um objeto Date internamente armazena:

- O número de milissegundos desde a meia noite do dia 1 de janeiro de 1970 GMT (UTC)
 - *GMT: Greenwich Mean Time (time zone)*
 - *UTC: Coordinated Universal Time (time standard)*

SimpleDateFormat

• <https://docs.oracle.com/javase/10/docs/api/java/text/SimpleDateFormat.html>

• Define formatos para conversão entre Date e String

• dd/MM/yyyy -> 23/07/2018

• dd/MM/yyyy HH:mm:ss -> 23/07/2018 15:42:07

Padrão ISO 8601 e classe Instant

- Formato: yyyy-MM-ddTHH:mm:ssZ
- Exemplo: "2018-06-25T15:42:07Z"
- `Date y3 = Date.from(Instant.parse("2018-06-25T15:42:07Z"));`

```
SimpleDateFormat sdf1 = new SimpleDateFormat("dd/MM/yyyy");
SimpleDateFormat sdf2 = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
SimpleDateFormat sdf3 = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
sdf3.setTimeZone(TimeZone.getTimeZone("GMT"));

Date x1 = new Date();
Date x2 = new Date(System.currentTimeMillis());
Date x3 = new Date(0L);
Date x4 = new Date(1000L * 60L * 60L * 5L);

Date y1 = sdf1.parse("25/06/2018");
Date y2 = sdf2.parse("25/06/2018 15:42:07");
Date y3 = Date.from(Instant.parse("2018-06-25T15:42:07Z"));

System.out.println("x1: " + x1);
System.out.println("x2: " + x2);
System.out.println("x3: " + x3);
System.out.println("x4: " + x4);
System.out.println("y1: " + y1);
System.out.println("y2: " + y2);
System.out.println("y3: " + y3);
System.out.println("-----");
System.out.println("x1: " + sdf2.format(x1));
System.out.println("x2: " + sdf2.format(x2));
System.out.println("x3: " + sdf2.format(x3));
System.out.println("x4: " + sdf2.format(x4));
System.out.println("y1: " + sdf2.format(y1));
System.out.println("y2: " + sdf2.format(y2));
System.out.println("y3: " + sdf2.format(y3));
System.out.println("-----");
System.out.println("x1: " + sdf3.format(x1));
System.out.println("x2: " + sdf3.format(x2));
System.out.println("x3: " + sdf3.format(x3));
System.out.println("x4: " + sdf3.format(x4));
System.out.println("y1: " + sdf3.format(y1));
System.out.println("y2: " + sdf3.format(y2));
System.out.println("y3: " + sdf3.format(y3));
```

*Demo:
Criação e
impressão
de datas.*

Manipulando uma data com Calendar

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Somando uma unidade de tempo

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
  
Date d = Date.from(Instant.parse("2018-06-25T15:42:07Z"));  
  
System.out.println(sdf.format(d));  
  
Calendar cal = Calendar.getInstance();  
cal.setTime(d);  
cal.add(Calendar.HOUR_OF_DAY, 4);  
d = cal.getTime();  
  
System.out.println(sdf.format(d));
```

Obtendo uma unidade de tempo

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");

Date d = Date.from(Instant.parse("2018-06-25T15:42:07Z"));

System.out.println(sdf.format(d));

Calendar cal = Calendar.getInstance();
cal.setTime(d);
int minutes = cal.get(Calendar.MINUTE);
int month = 1 + cal.get(Calendar.MONTH);

System.out.println("Minutes: " + minutes);
System.out.println("Month: " + month);
```

Operadores aritméticos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

NOTAS:

1) * / % tem precedência maior que + -

2) Exemplos:

$3 + 4 * 2 \longrightarrow$ Resultado: 11

$(3 + 4) * 2 \longrightarrow$ Resultado: 14

3) Pode-se usar parêntesis à vontade

4) Exemplo com mod:

$17 \% 3 \longrightarrow$ Resultado: 2

$$\begin{array}{r} 17 \overline{) 3} \\ 2 \quad 5 \end{array}$$

Demo

```
int n1 = 3 + 4 * 2;
int n2 = (3 + 4) * 2;
int n3 = 17 % 3;
double n4 = 10.0 / 8.0;
```

```
double a = 1.0, b = -3.0, c = -4.0;
double x1 = (-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a);
```

```
System.out.println(n1);
System.out.println(n2);
System.out.println(n3);
System.out.println(n4);
System.out.println(x1);
```

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Operadores de atribuição

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores de atribuição

Operador	Exemplo	Significado
=	a = 10;	a RECEBE 10
+=	a += 2;	a RECEBE a + 2;
-=	a -= 2;	a RECEBE a - 2;
*=	a *= 2;	a RECEBE a * 2;
/=	a /= 2;	a RECEBE a / 2;
%=	a %= 3;	a RECEBE a % 3;

Demo

```
double n1 = 10;
double n2 = 30;
String s = "ABC";
System.out.println(n1);
System.out.println(n2);
System.out.println(s);

System.out.println("-----");

n1 *= 2;
n2 += n1;
s += "DEF";
System.out.println(n1);
System.out.println(n2);
System.out.println(s);
```

Operadores aritméticos / atribuição

Operador	Exemplo	Significado
++	a++; ou ++a;	a = a + 1;
--	a--; ou --a;	a = a - 1;

```
int a = 10;
a++;
System.out.println(a);

SAÍDA:
11
```

```
int a = 10;
int b = a++;
System.out.println(a);
System.out.println(b);

SAÍDA:
11
10
```

```
int a = 10;
int b = ++a;
System.out.println(a);
System.out.println(b);

SAÍDA:
11
11
```

Operadores comparativos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores comparativos

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

Demo

```
int a = 10;
boolean c1 = a < 10;
boolean c2 = a < 20;
boolean c3 = a > 10;
boolean c4 = a > 5;
System.out.println(c1);
System.out.println(c2);
System.out.println(c3);
System.out.println(c4);
System.out.println("-----");

boolean c5 = a <= 10;
boolean c6 = a >= 10;
boolean c7 = a == 10;
boolean c8 = a != 10;
System.out.println(c5);
System.out.println(c6);
System.out.println(c7);
System.out.println(c8);
```

Operadores lógicos

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores lógicos

Operador	Significado
&&	E
 	OU
!	NÃO

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

NOTAS:

1) Precedência: ! > && > ||

2) Pode-se usar parêntesis à vontade

3) Exemplos:

2 > 3 || 4 != 5 → Resultado: true
 !(2>3) && 4 != 5 → Resultado: true

Demo

```
boolean c1 = 2 > 3 || 4 != 5; // true
boolean c2 = !(2>3) && 4 != 5; // true
System.out.println(c1);
System.out.println(c2);

System.out.println("-----");

boolean c3 = 10 < 5; // false
boolean c4 = c1 || c2 && c3; // true
System.out.println(c4);
```

Operadores bitwise

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Operadores bitwise

Operador	Significado
&	Operação "E" bit a bit
 	Operação "OU" bit a bit
^	Operação "OU-exclusivo" bit a bit

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

C1	C2	C1 XOR C2
F	F	F
F	V	V
V	F	V
V	V	F

Demo

(89)	0101	1001	}	\swarrow \rightarrow \searrow	&: 0001 1000	(24)
(60)	0011	1100			: 0111 1101	(125)
					^: 0110 0101	(101)

```

int n1 = 89;
int n2 = 60;
System.out.println(n1 & n2);
System.out.println(n1 | n2);
System.out.println(n1 ^ n2);

```

Aplicação comum: verificar bit

(89)	0101	1001	}	&: 0000 0000	(0)
(32)	0010	0000			
(113)	0111	0001	}	&: 0010 0000	(32)
(32)	0010	0000			

```
package course;

import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int mask = 0b100000;
        int n = sc.nextInt();

        if ((n & mask) != 0) {
            System.out.println("6th bit is true!");
        }
        else {
            System.out.println("6th bit is false");
        }

        sc.close();
    }
}
```

Entrada de dados em Java - Parte 1

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Scanner

Uma das formas de se ler dados em Java é por meio de um objeto do tipo **Scanner**, o qual deve ser instanciado com o argumento **System.in**:

`Scanner sc = new Scanner(System.in);`

`import java.util.Scanner;`

faça `sc.close()` quando não precisar mais do objeto `sc`

```
int x = sc.nextInt();
double y = sc.nextDouble();
String z = sc.next();
```

ATENÇÃO:

Use `Locale.setDefault(Locale.US);`
ANTES de criar o Scanner para separador de decimais como ponto.

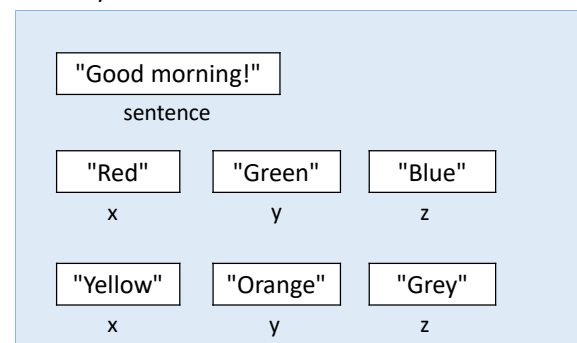
Checklist

- Ler um texto até a quebra de linha e armazenar em uma variável
- Ler três palavras, uma em cada linha, armazenando cada uma em uma variável
- Ler três palavras na mesma linha, separadas por espaço, armazenando cada uma em uma variável

Console:

```
Good morning!
Red
Green
Blue
Yellow Orange Grey
```

Memory:



Entrada de dados em Java - Parte 2

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

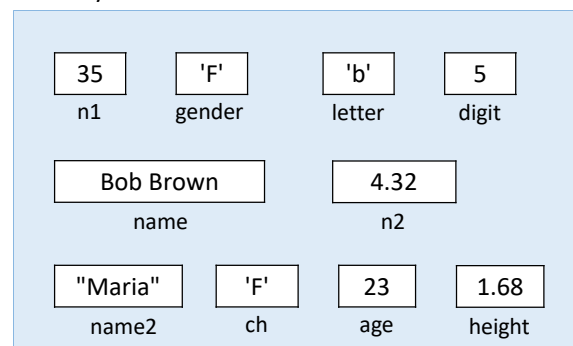
Checklist

- Ler um número inteiro
- Ler um texto até a quebra de linha
- Ler um caractere
- Ler um código composto de uma letra e um dígito (exemplo: "a3"). Armazenar os valores em uma variável tipo **char** e outra tipo **int**, respectivamente.
- Ler um número double
- Ler um nome (única palavra), sexo (caractere F ou M), idade (inteiro) e altura (double) na mesma linha, armazenando-os em quatro variáveis com os devidos tipos

Console:

```
35
Bob Brown
F
b5
4.32
Maria F 23 1.68
```

Memory:



Limpeza de stream de entrada (flush)

DETALHE IMPORTANTE:

Antes de fazer um `nextLine()`

- Se antes foi dada uma quebra de linha na stream de entrada sem o `nextLine()`
- Então é preciso fazer um `nextLine()` extra para "limpar" a quebra de linha que fica pendente na stream

Operação substring

```
String s = "abcdefgh";  
  
String sub1 = s.substring(3);  
String sub2 = s.substring(3, 5);
```

```
System.out.println(sub1);  
System.out.println(sub2);
```

SAÍDA:

```
defgh  
de
```

```

Locale.setDefault(Locale.US);
Scanner sc = new Scanner(System.in);

// 35
int n1 = sc.nextInt();
// Bob Brown
sc.nextLine();
String name = sc.nextLine();
// F
char gender = sc.next().charAt(0);
// b5
String s = sc.next();
char letter = s.charAt(0);
int digit = Integer.parseInt(s.substring(1));
// 4.32
double n2 = sc.nextDouble();
// Maria F 23 1.68
String name2 = sc.next();
char ch = sc.next().charAt(0);
int age = sc.nextInt();
double height = sc.nextDouble();

System.out.println(n1);
System.out.println(name);
System.out.println(gender);
System.out.println(letter);
System.out.println(digit);
System.out.println(n2);
System.out.println(name2);
System.out.println(ch);
System.out.println(age);
System.out.println(height);
sc.close();

```

Exercício de fixação

Fazer um programa para executar o seguinte procedimento de entrada de dados, e depois mostrar os dados na tela:

```

Enter your full name:
Alex Green
How many bedrooms are there in your house?
3
Enter product price:
500.50
Enter your last name, age and height (same line):
Green 21 1.73

```

SAÍDA ESPERADA (NÚMEROS REAIS COM DUAS CASAS DECIMAIS):

```

Alex Green
3
500.50
Green
21
1.73

```

(correção na próxima página)


```

package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter your full name:");
        String fullName = sc.nextLine();
        System.out.println("How many bedrooms are there in your house?");
        int bedrooms = sc.nextInt();
        System.out.println("Enter product price:");
        double price = sc.nextDouble();
        System.out.println("Enter your last name, age and height (same line):");
        sc.nextLine();
        String[] vect = sc.nextLine().split(" ");
        String lastName = vect[0];
        int age = Integer.parseInt(vect[1]);
        double height = Double.parseDouble(vect[2]);

        System.out.println(fullName);
        System.out.println(bedrooms);
        System.out.printf("%.2f%n", price);
        System.out.println(lastName);
        System.out.println(age);
        System.out.printf("%.2f%n", height);

        sc.close();
    }
}

```

Funções interessantes para String

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Formatar: toLowerCase(), toUpperCase(), trim()
- Recortar: substring(inicio), substring(inicio, fim)
- Substituir: Replace(char, char), Replace(string, string)
- Buscar: IndexOf, LastIndexOf
- str.Split(" ")

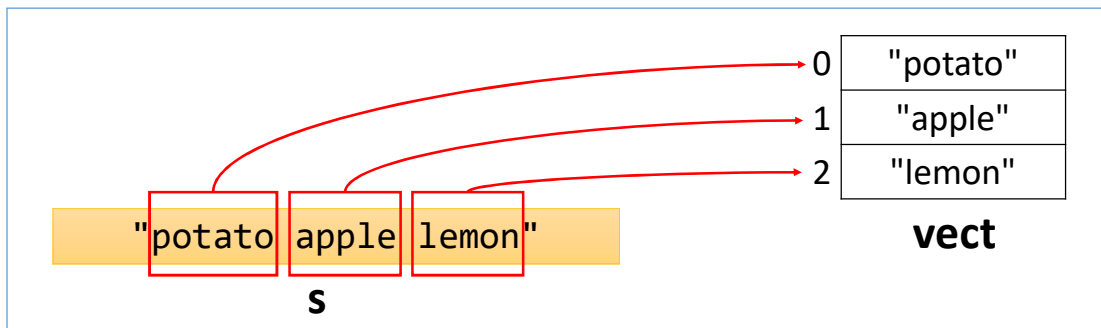
```
String original = "abcde FGHIJ ABC abc DEFG ";

String s01 = original.toLowerCase();
String s02 = original.toUpperCase();
String s03 = original.trim();
String s04 = original.substring(2);
String s05 = original.substring(2, 9);
String s06 = original.replace('a', 'x');
String s07 = original.replace("abc", "xy");
int i = original.indexOf("bc");
int j = original.lastIndexOf("bc");

System.out.println("Original: -" + original + "-");
System.out.println("toLowerCase: -" + s01 + "-");
System.out.println("toUpperCase: -" + s02 + "-");
System.out.println("trim: -" + s03 + "-");
System.out.println("substring(2): -" + s04 + "-");
System.out.println("substring(2, 9): -" + s05 + "-");
System.out.println("replace('a', 'x'): -" + s06 + "-");
System.out.println("replace('abc', 'xy'): -" + s07 + "-");
System.out.println("Index of 'bc': " + i);
System.out.println("Last index of 'bc': " + j);
```

Operação split

```
String s = "potato apple lemon";  
  
String[] vect = s.split(" ");  
String word1 = vect[0];  
String word2 = vect[1];  
String word3 = vect[2];
```



Estruturas condicionais

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

if-else

Simples

```
if ( condição ) {
    comando 1
    comando 2
}
```

Composta

```
if ( condição ) {
    comando 1
    comando 2
}
else {
    comando 3
    comando 4
}
```

Encadeamentos

```
if ( condição 1 ) {
    comando 1
    comando 2
}
else if ( condição 2 ) {
    comando 3
    comando 4
}
else if ( condição 3 ) {
    comando 5
    comando 6
}
else {
    comando 7
    comando 8
}
```

Nota: se o bloco de comandos possuir apenas um comando, as chaves são opcionais.

Demo #1

Enter an integer number:

10

Even!

Enter an integer number?

15

Odd!

Demo #2

What time is it?

10

Good morning!

$\text{time} < 12$

What time is it?

14

Good afternoon!

$12 \leq \text{time} < 18$

What time is it?

19

Good evening!

$\text{time} \geq 18$

Exercício de fixação

Fazer um programa para ler três números inteiros na mesma linha, separados por espaço em branco. Em seguida, mostrar na tela qual é o maior dos números lidos. Se houver empate, mostrar apenas um.

Exemplos:

Entrada	Saída
4 10 7	Higher = 10
Entrada	Saída
7 3 7	Higher = 7
Entrada	Saída
9 9 9	Higher = 9

(correção na próxima página)

```
package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        if (a > b && a > c) {
            System.out.println("Higher = " + a);
        }
        else if (b > c) {
            System.out.println("Higher = " + b);
        }
        else {
            System.out.println("Higher = " + c);
        }

        sc.close();
    }
}
```

Sintaxe alternativa: switch-case

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

switch-case

Estrutura opcional a vários if-else encadeados, **quando a condição envolve o teste do valor de uma variável.**

Sintaxe:

```
var minhaVariavel = (...);

switch (minhaVariavel) {
    case 1:
        System.out.println("Caso 1");
        break;
    case 2:
        System.out.println("Caso 2");
        break;
    default:
        System.out.println("Caso padrão");
}
```

```
Scanner sc = new Scanner(System.in);

int x = sc.nextInt();
String day;

if (x == 1) {
    day = "Sunday";
}
else if (x == 2) {
    day = "Monday";
}
else if (x == 3) {
    day = "Tuesday";
}
else if (x == 4) {
    day = "Wednesday";
}
else if (x == 5) {
    day = "Thursday";
}
else if (x == 6) {
    day = "Friday";
}
else if (x == 7) {
    day = "Saturday";
}
else {
    day = "Invalid value";
}

System.out.println("Day: " + day);
```

```
Scanner sc = new Scanner(System.in);

int x = sc.nextInt();
String day;

switch (x) {
    case 1:
        day = "Sunday";
        break;
    case 2:
        day = "Monday";
        break;
    case 3:
        day = "Tuesday";
        break;
    case 4:
        day = "Wednesday";
        break;
    case 5:
        day = "Thursday";
        break;
    case 6:
        day = "Friday";
        break;
    case 7:
        day = "Saturday";
        break;
    default:
        day = "Invalid value";
}

System.out.println("Day: " + day);
```

Expressão condicional ternária

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Expressão condicional ternária

Estrutura opcional ao if-else quando se deseja decidir um **VALOR** com base em uma condição.

Sintaxe:

(`condição`) ? `valor_se_verdadeiro` : `valor_se_falso`

Exemplos:

(`2 > 4`) ? `50` : `80`  **80**

(`10 != 3`) ? `"Maria"` : `"Alex"`  **"Maria"**

Demo

```
double preco = 34.5;
double desconto;
if (preco < 20.0) {
    desconto = preco * 0.1;
}
else {
    desconto = preco * 0.05;
}
```

```
double preco = 34.5;
double desconto = (preco < 20.0) ? preco * 0.1 : preco * 0.05;
```

Escopo e inicialização

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Checklist

- Escopo de uma variável: é a região do programa onde a variável é válida, ou seja, onde ela pode ser referenciada.
- Uma variável não pode ser usada se não for iniciada.
- Falaremos de escopo de métodos no Capítulo 5

Demo

```
double price = sc.nextDouble();  
  
if (price > 100.0) {  
    double discount = price * 0.1;  
}  
  
System.out.println(discount);
```

Comentários em Java (básico)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

```
package course;

import java.util.Locale;
import java.util.Scanner;

/*
Este programa calcula as raízes de uma equação do segundo grau
Os valores dos coeficientes devem ser digitados um por linha
*/
public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        double a, b, c, delta;

        System.out.println("Digite os valores dos coeficientes:");
        a = sc.nextDouble();
        b = sc.nextDouble();
        c = sc.nextDouble();

        delta = b * b - 4 * a * c; // cálculo do valor de delta
    }
}
```

Funções (sintaxe)

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Funções

- Representam um processamento que possui um significado
 - `Math.sqrt(double)`
 - `System.out.println(string)`
- Principais vantagens: modularização, delegação e reaproveitamento
- Dados de entrada e saída
 - Funções podem receber dados de entrada (parâmetros ou argumentos)
 - Funções podem ou não retornar uma saída
- Em orientação a objetos, funções em classes recebem o nome de "métodos"

Problema exemplo

Fazer um programa para ler três números inteiros e mostrar na tela o maior deles.

Exemplo:

```
Enter three numbers:  
5  
8  
3  
Higher = 8
```

```
package course;  
  
import java.util.Scanner;  
  
public class Program {  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter three numbers:");  
        int a = sc.nextInt();  
        int b = sc.nextInt();  
        int c = sc.nextInt();  
  
        if (a > b && a > c) {  
            System.out.println("Higher = " + a);  
        } else if (b > c) {  
            System.out.println("Higher = " + b);  
        } else {  
            System.out.println("Higher = " + c);  
        }  
  
        sc.close();  
    }  
}
```

```

package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter three numbers:");
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();

        int higher = max(a, b, c);

        showResult(higher);

        sc.close();
    }

    public static int max(int x, int y, int z) {
        int aux;
        if (x > y && x > z) {
            aux = x;
        } else if (y > z) {
            aux = y;
        } else {
            aux = z;
        }
        return aux;
    }

    public static void showResult(int value) {
        System.out.println("Higher = " + value);
    }
}



```

Debugging com Eclipse

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Tópicos

- Teclas
 - CTRL+SHIFT+B - marcar/desmarcar breakpoint
 -  - iniciar/continuar o debug
 - F6 - executar um passo (pula função)
 - F5 - executar um passo (entra na função)
 - F7 - sair do método em execução
 -  - parar debug
- Perspectiva Debug
 - Aba importante: Variables

Estrutura repetitiva while

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura "enquanto"

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

V: executa e volta

F: pula fora

Comparação com if simples

```
if ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

**V: executa
F: pula fora**

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

Regra:

**V: executa e volta
F: pula fora**

Problema exemplo

Digitar um número e mostrar sua raiz quadrada com três casas decimais, depois repetir o procedimento. Quando o usuário digitar um número negativo (podendo inclusive ser na primeira vez), mostrar uma mensagem "Negative number" e terminar o programa.

```
Enter a number: 25  
5.000  
Enter another number: 10  
3.162  
Enter another number: 9  
3.000  
Enter another number: -4  
Negative number
```

Estrutura repetitiva for

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura "para"

Executa somente
na primeira vez

V: executa e volta
F: pula fora

Executa toda vez depois
de voltar

```
for ( início ; condição ; incremento ) {  
    comando 1  
    comando 2  
}
```

Problema exemplo

Digitar um número N e depois N valores inteiros. Mostrar a soma dos N valores digitados.

```
How many integer numbers are you going to enter? 3  
Value #1: 10  
Value #2: 7  
Value #3: 8  
Sum = 25
```

```
package course;

import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("How many integer numbers are you going to enter: ");
        int n = sc.nextInt();

        int sum = 0;
        for (int i = 1; i <= n; i++) {
            System.out.print("Value #" + i + ": ");
            int x = sc.nextInt();
            sum += x;
        }

        System.out.println("Sum = " + sum);

        sc.close();
    }
}
```

Exercício de fixação - estruturas repetitivas while e for

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Exercício de fixação

Fazer um programa para ler um número N (se for digitado um valor não positivo, mostrar uma mensagem e ler novamente). Em seguida, N valores inteiros. Mostrar o maior dentre os N números digitados. Veja exemplo:

Exemplo:

```
Enter N: 0
N must be positive! Try again: -2
N must be positive! Try again: 4
Value #1: 5
Value #2: 4
Value #3: 10
Value #4: 2
Higher = 10
```

(correção na próxima página)

Dica

Integer.*MIN_VALUE*

Integer.*MAX_VALUE*

```

package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter N: ");
        int n = sc.nextInt();
        while (n <= 0) {
            System.out.print("N must be positive! Try again: ");
            n = sc.nextInt();
        }

        int higher = Integer.MIN_VALUE;
        for (int i = 1; i <= n; i++) {
            System.out.print("Value #" + i + ": ");
            int x = sc.nextInt();
            if (x > higher) {
                higher = x;
            }
        }

        System.out.println("Higher = " + higher);
        sc.close();
    }
}

```

Estrutura repetitiva do-while

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

Estrutura "faça-enquanto"

```
do {  
    comando 1  
    comando 2  
} while ( condição );
```

Regra:

V: volta

F: pula fora

Problema exemplo

Digitar um número e mostrar sua raiz quadrada, depois perguntar ao usuário se ele deseja repetir o procedimento (y/n). Caso ele responda 'y', repetir o procedimento.

```
Enter a number: 9  
Square root = 3.000  
Repeat (y/n)? y  
Enter a number: 10  
Square root = 3.162  
Repeat (y/n)? n
```

```
package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        char resp;
        do {
            System.out.print("Enter a number: ");
            double n = sc.nextDouble();
            double sq = Math.sqrt(n);
            System.out.printf("Square root = %.3f%n", sq);

            System.out.print("Repeat (y/n)? ");
            resp = sc.next().charAt(0);
        } while (resp != 'n');

        sc.close();
    }
}
```

Exercícios de reforço

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

1) Deseja-se fazer um programa para calcular a nota final de um estudante, considerando as três notas que ele tirou nos três trimestres de seu curso. Caso a nota final do estudante seja menor que 60.00, deve-se mostrar uma mensagem "FAILED" também.

Exemplos:

```
Enter three student scores:
22.00 30.10 25.50
Final score = 77.60
```

```
Enter three student scores:
15.00 20.00 20.00
Final score = 55.00
FAILED
```

```
package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter three student scores: ");
        double score1 = sc.nextDouble();
        double score2 = sc.nextDouble();
        double score3 = sc.nextDouble();

        double finalScore = score1 + score2 + score3;

        System.out.printf("Final score: %.2f\n", finalScore);
        if (finalScore < 60.0) {
            System.out.println("FAILED");
        }

        sc.close();
    }
}
```


2) Fazer um programa para ler a quantidade de glicose no sangue de uma pessoa e depois mostrar na tela a classificação desta glicose de acordo com a tabela de referência ao lado.

Classificação	Glucose
Normal	glucose <= 100
High	100 < glucose <= 140
Diabetes	glucose > 140

Exemplos:

Enter glucose measure: 90.0
Normal

Enter glucose measure: 140.0
High

Enter glucose measure: 143.2
Diabetes

```
package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter glucose measure: ");
        double glucose = sc.nextDouble();

        if (glucose <= 100.0) {
            System.out.println("Normal");
        } else if (glucose <= 140.0) {
            System.out.println("High");
        } else {
            System.out.println("Diabetes");
        }

        sc.close();
    }
}
```

3) Fazer um programa para ler dois números inteiros e mostrar a soma de todos os números ímpares entre eles, inclusive eles próprios.

Exemplos:

Enter two integer numbers:

-10

2

Sum of odd numbers = -24

Enter two integer numbers:

14

3

Sum of odd numbers = 48

```
package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter two integer numbers: ");
        int x = sc.nextInt();
        int y = sc.nextInt();

        int begin = x, end = y;

        if (y < x) {
            begin = y;
            end = x;
        }

        int sum = 0;
        for (int i = begin; i <= end; i++) {
            if (i % 2 != 0) {
                sum += i;
            }
        }

        System.out.println("Sum of odd numbers = " + sum);

        sc.close();
    }
}
```

4) Leia números inteiros até que um 0 (zero) seja lido. Para cada valor lido, o programa deve mostrar o quadrado do valor.

Exemplo 1:

```
Enter an integer value: 4
16
Enter an integer value: 1
1
Enter an integer value: 10
100
Enter an integer value: 0
```

Exemplo 2:

```
Enter an integer value: 0
```

```
package course;

import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter an integer value: ");
        int n = sc.nextInt();

        while (n != 0) {
            int square = n * n;
            System.out.println(square);
            System.out.print("Enter an integer value: ");
            n = sc.nextInt();
        }

        sc.close();
    }
}
```

5) O curso de Computação Gráfica do IFTM é trimestral, sendo que as notas do ano são divididas em três partes: 30, 35 e 35, totalizando 100 pontos no ano. O professor de programação está querendo automatizar o processo de fechamento de notas. Faça um programa para somar as notas dos três trimestres de todos alunos da turma.

Exemplo:

```
Enter number of students: 3
Student #1: 20.00 31.00 30.00
Final score: 81.00
Student #2: 25.00 31.10 30.22
Final score: 86.32
Student #3: 19.25 29.00 18.31
Final score: 66.56
```

```
package course;

import java.util.Locale;
import java.util.Scanner;

public class Program {
    public static void main(String[] args) {

        Locale.setDefault(Locale.US);
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of students: ");
        int n = sc.nextInt();
        for (int i = 1; i <= n; i++) {
            System.out.print("Student #" + i + ": ");
            double score1 = sc.nextDouble();
            double score2 = sc.nextDouble();
            double score3 = sc.nextDouble();
            double finalScore = score1 + score2 + score3;
            System.out.printf("Final score: %.2f\n", finalScore);
        }

        sc.close();
    }
}
```