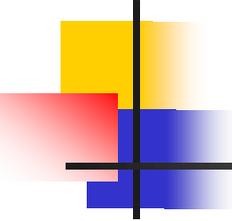


Sintaxe Básica de Java - Parte 1

Universidade Católica de Pernambuco
Ciência da Computação

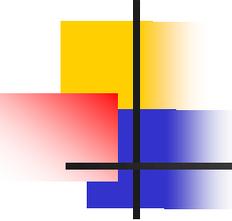
Prof. Márcio Bueno
pooite@marciubeno.com

Fonte: Material da Prof^a Karina Oliveira



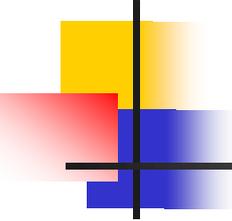
Estrutura de Programa

- Um programa Java fica dentro de uma classe.
- Os programas começam com o método (função) `main()`:
 - `public static void main(String[] args)`
 - `args` => passagem de argumentos na linha de comando para o programa Java.
 - A execução da aplicação Java continua até que o método `main()` termine de executar todas as suas instruções.
- Linguagem estruturada em blocos.
 - Blocos são delimitados por chaves “{” e “}”.
- Instruções são terminadas por ponto-e-vírgula.



Estrutura de Programa

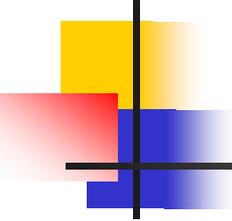
- Java é *case sensitive*: faz a diferenciação entre maiúsculas e minúsculas.
- Comentários
 - `/*` Este é um comentário de várias linhas. `*/`
 - `//` Este é um comentário de uma única linha.
 - `/**` Este é um comentário no padrão Javadoc `*/`



Estrutura de Programa

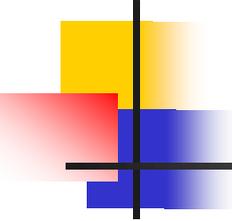
- Exemplo:

```
/*  
 * Meu primeiro programa Java.  
 * Autor: Karina Oliveira  
 */  
  
public class HelloWorld {  
    public static void main(String[ ] args) {  
        // Imprime texto na tela.  
        System.out.println("Hello World!");  
    }  
}
```



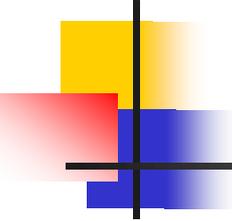
Estrutura de Programa

- Compilando e executando o primeiro programa Java usando o J2SDK - Java 2 Software Development Kit.
 1. Editar o programa em um editor de textos qualquer (notepad, por exemplo).
 2. Salvar o programa com o mesmo nome utilizado para a classe + a extensão java
Exemplo: `public class HelloWorld` ⇒ `HelloWorld.java`
 3. Abrir sessão DOS e ir para o diretório onde foi gravado o arquivo com o primeiro programa Java.
 4. Configurar a variável de ambiente path
`set path=C:\j2sdk1.4.2\bin;%path%`



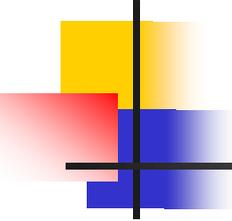
Estrutura de Programa

- Compilando e executando o primeiro programa Java usando o J2SDK - Java 2 Software Development Kit (Cont.).
 5. Configurar a variável de ambiente classpath
set
classpath=.;C:\j2sdk1.4.2\jre\lib\rt.jar;%classpath%
 5. Compilar o programa:
C:\> javac HelloWorld.java
 6. Executar o programa:
C:\> java HelloWorld



Ferramentas para Desenvolvimento

- NetBeans
 - <http://www.netbeans.org/>
- JBuilder
 - <http://www.borland.com/br/products/jbuilder/>
- Eclipse
 - <http://www.eclipse.org>
- JCreator
 - <http://www.jcreator.com/download.htm>



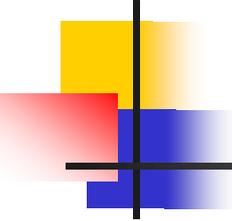
Tipos de Dados

■ Inteiros

- byte = inteiro com sinal (1 byte)
- short = inteiro com sinal (2 bytes)
- **int** = inteiro com sinal (4 bytes)
- long = inteiro com sinal (8 bytes)

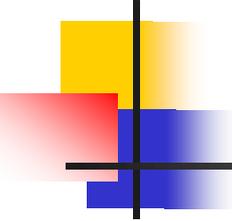
■ Ponto Flutuante

- float = ponto flutuante (4 bytes)
- **double** = ponto flutuante (8 bytes)



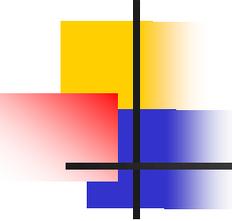
Tipos de Dados

- `char` = caractere (2 bytes)
 - Caracteres Especiais
 - `\'` → apóstrofo
 - `\"` → aspas
 - `\\` → barra invertida
 - `\b` → backspace
 - `\n` → nova linha
 - `\t` → tabulação
- `boolean` = `true` ou `false` (1 bit)
- `String` = Uma seqüência de caracteres



Tipos de Dados

- Valores Literais
 - boolean → true / false
 - **int** → exemplo: 100
 - float → exemplo: 1.5F
 - **double** → exemplo: 1.5
 - char → exemplo: 'a'
 - String → exemplos: "a...a", null, ""



Declaração de Variáveis

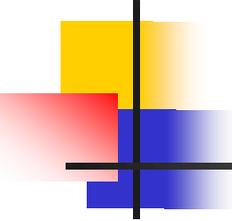
- **Sintaxes:**

- `<tipo> <nome da variável>;` ou
- `<tipo> <nome1>, <nome2>, ..., <nomeN>;` ou
- `<tipo> <nome> = <valor inicial>;`

- **Exemplos:**

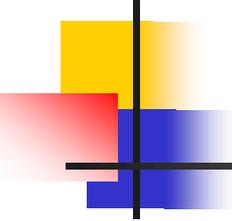
- `int valor = 10;`
- `float peso, altura;`
- `String nome = "CATOLICA";`

- **OBS 1:** Podem ser declaradas em qualquer ponto de um método.
- **OBS 2:** Variáveis locais devem ser inicializadas.



Nomes de Variáveis

- O primeiro caractere de um nome de variável deverá ser uma letra ou um sublinhado (“_”).
- Os caracteres subsequentes podem ser letras, sublinhados ou números.
- Palavras reservadas não podem ser usadas para nomear variáveis (ex.: import, int, double, ...)
- Exemplos de nomes válidos: minhaVariavel, VAR1, _teste
- Exemplos de nomes inválidos: 10valor, int, tes-te, @nome
- OBS: teste é diferente de TESTE



Declaração de Constantes

- Para definir uma constante utiliza-se a palavra reservada *final*

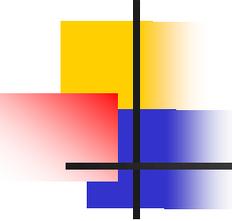
- Exemplo:

```
final double PI = 3.14;
```

- Para definir uma constante de classe são utilizadas as palavras reservadas *static final*

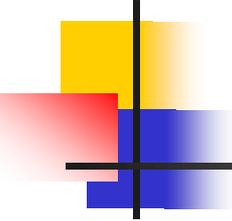
- Exemplo:

```
static final double PI = 3.14;
```



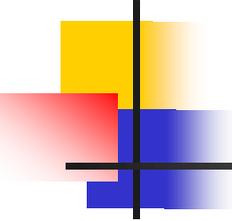
Operadores

- **Operadores Matemáticos**
 - Aritméticos binários: +, -, *, /, %
 - Aritméticos unários: +, -
 - Incremento: ++
 - Decremento: --
- **Operadores de Atribuição**
 - Atribuição: =
 - Atribuição com operação: *=, /=, %=, +=, -=
- **OBS: Os operadores + e += podem ser usados para concatenação de strings.**



Operadores

- **Operadores Relacionais e Lógicos**
 - Quando usados retornam sempre um valor booleano (**true** ou **false**).
 - Igualdade: ==
 - Diferença: !=
 - Menor, menor ou igual: <, <=
 - Maior, maior ou igual: >, >=
 - Negação: !
 - 'e' lógico: &&
 - 'ou' lógico: ||



Operadores

- Exponenciação

- Java não tem operador para exponenciação
- Necessário usar o método `pow` da classe `Math` da biblioteca `java.lang`

- Exemplo:

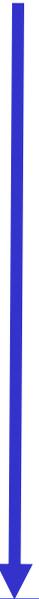
```
double y = Math.pow(x, a);
```

Operadores

■ Precedência de operadores

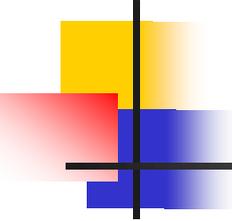
- ++, -- (usados como prefixos); +, - (unários), !
- *, /, %
- +, -
- <, >, <=, >=
- ==, !=
- &&
- ||
- =, *=, /=, %=, +=, -=
- ++, -- (usados como sufixos)

maior



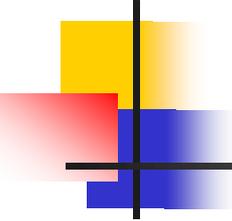
menor

OBS: O uso de parênteses pode alterar esta ordem de precedência.



Entrada de Dados

- Pode-se utilizar a classe *JOptionPane*
- Produz janelas para interação com o usuário
- Permite a entrada de dados do usuário através do uso do método *showInputDialog*
- Exemplo: Programa para entrar com o nome de um aluno e ao final apresentar o nome digitado (próximo slide)



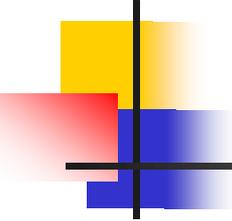
Entrada de Dados

- OBS 1: A seguinte linha de código deve ser adicionada ao início do programa:

```
import javax.swing.JOptionPane;
```

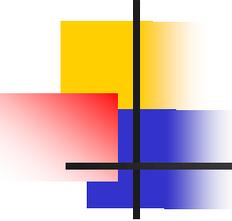
Veremos adiante detalhes sobre a importação de classes e pacotes em Java.

- OBS 2: Os dados são passados para o programa no formato String. Portanto, conversões devem ser feitas caso o formato desejado na entrada seja um outro tipo diferente, como int, double, etc.



Entrada de Dados

- Para cada um dos tipos primitivos existe uma classe Java associada com métodos úteis para manipular valores destes tipos (Consultar documentação da API Java):
 - boolean - Boolean
 - char - Character
 - short - Short
 - int - Integer
 - long - Long
 - float - Float
 - double - Double



Entrada de Dados

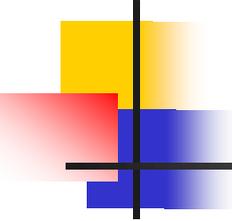
- Exemplo:

- Classe Integer - método para converter String para inteiro.

```
String entrada =  
JOptionPane.showInputDialog("Digite o valor:");  
int a = Integer.parseInt(entrada);
```

- Classe Float - Float.parseFloat
- Classe Boolean - Boolean.parseBoolean
- Conversão de String para char:

```
char valor =  
JOptionPane.showInputDialog("Digite o valor:").charAt(0);
```



Saída de Dados

- Comandos de impressão na tela

- Saída Padrão:

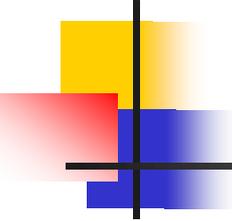
```
System.out.print("Exemplo");
```

```
System.out.println("O resultado é = " + valor);
```

```
System.out.println(a + b)
```

- Usando JOptionPane:

```
JOptionPane.showMessageDialog(null, "O resultado  
é = " + valor);
```



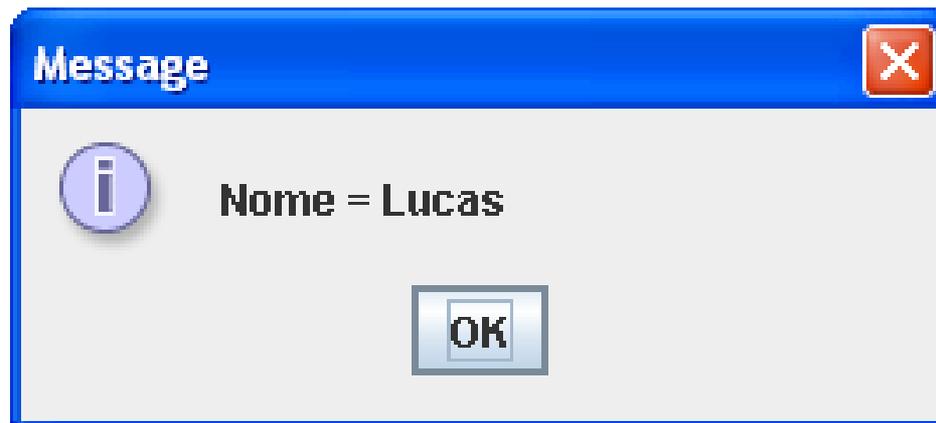
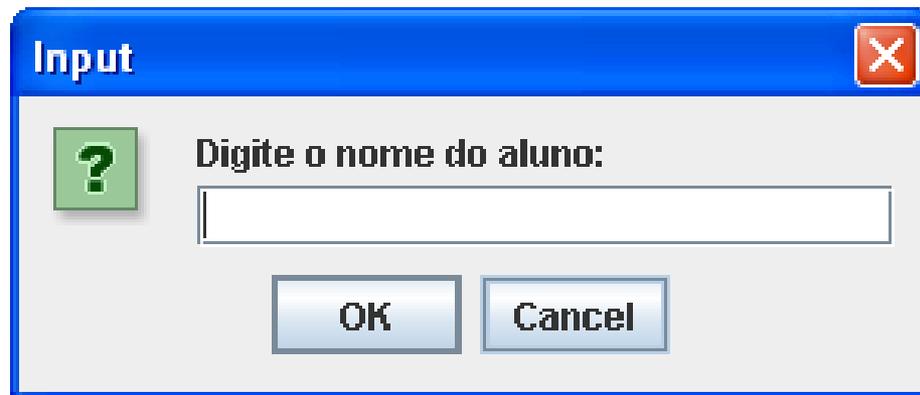
Entrada e Saída de Dados

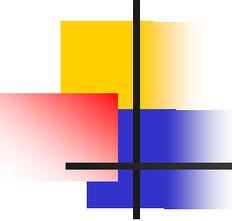
- Exemplo - Código Fonte:

```
import javax.swing.JOptionPane;
public class Exemplo {
    public static void main(String args[]) {
        String retorno =
        JOptionPane.showInputDialog("Digite o nome do aluno:");
        System.out.println("Nome = " + retorno);
        JOptionPane.showMessageDialog(null, "Nome = " + retorno);
    }
}
```

Entrada e Saída de Dados

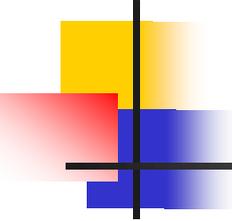
- Exemplo - Resultado:





Estilo de Codificação Java

- Estilo baseado no *Java Code Conventions*
- Identificadores
 - Identificadores são os nomes que o programador utiliza para nomear suas classes, métodos e variáveis.
 - Nomes de classe com a primeira letra maiúscula, exemplos:
 - `Teste`
 - `MinhaClasse`
 - Nomes de pacotes em letras minúsculas, exemplos:
 - `java.lang`
 - `java.pacoteTeste`



Estilo de Codificação Java

■ Identificadores (Cont.)

- Nomes de variáveis iniciando com letra minúscula, exemplos:
 - `variavel`
 - `minhaVariavel`
- Nomes de constantes todas em letras maiúsculas, exemplos:
 - `PI`
 - `CONSTANTE_EXEMPLO`
- Convenção para `{ e }`.
- Documentar cabeçalho de classes, atributos e métodos como comentários no estilo Javadoc
`/** ... */`