

| | |
|---------------------------|--|
| Objetivos | a. Apresentar conceitos da linguagem Java b. Iniciar a ligação entre o mundo “estruturado-procedimental” (linguagem C) e o mundo “orientado a objetos” (Java) |
| Referências gerais | W3 SCHOOLS. <i>Java Tutorial</i> . Disponível em < https://www.w3schools.com/java/default.asp >. Acesso em 21 fev 2024. DEV.JAVA. <i>Learn Java</i> . Disponível em < https://dev.java/learn/ >. Acesso em 21 fev 2024. VS CODE. <i>Getting Started with Java in Visual Studio Code</i> . Disponível em < https://code.visualstudio.com/docs/java/java-tutorial >. Acesso em 21 fev 2024. |

Objetivo específico desta aula prática: apresentar os conceitos da divisão do código Java em diferentes classes, organizando-as por meio de um *projeto*.

Suporte teórico 1. A estrutura geral de um arquivo de código Java simples (arquivo único) é:

| Código-exemplo | Observações |
|--|---|
| <pre>// Classe Programa (mesmo nome do arquivo .java) public class Programa { // ponto de início da execução ("programa principal") public static void main(String[] args) { // declarações de variáveis int x; // realização das tarefas previstas x = 5; System.out.println("Valor inicial: " + x); x = x + 10; System.out.println("Depois da alteração: " + x); } // final do método 'main' } // final da classe 'Programa'</pre> | <p>O nome da classe dever ser o mesmo do arquivo .java.</p> <p>Execução tem início no método main.</p> <p>Pode haver mais de uma variável.</p> <p>Ações são codificadas <u>dentro</u> do metodo main.</p> <p>No exemplo, println é uma ação que está programada em outro código Java, System.out, importado (automaticamente neste caso).</p> |

Atividade 1. Analise o programa Java escrito no arquivo da pasta Versao_1_..., o qual implementa as ações abaixo diretamente no main.

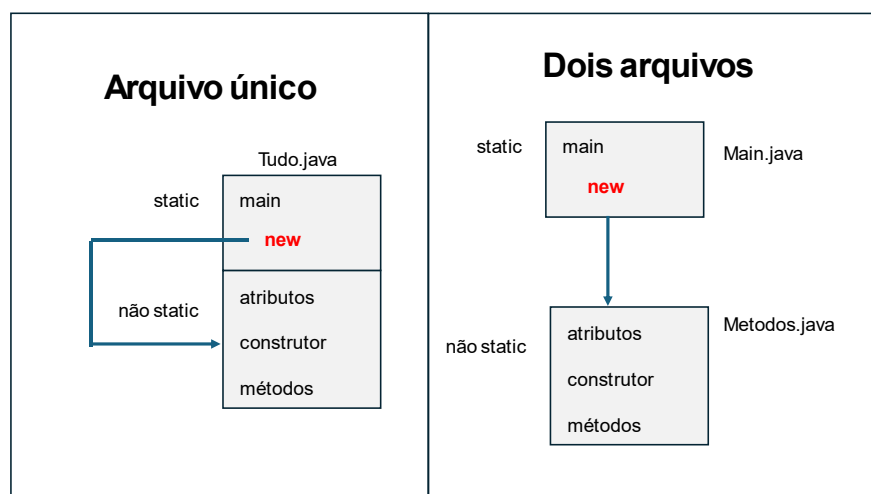
| Opção do menu | Ação correspondente |
|---------------|---|
| 1 | Receber (via teclado) dois argumentos inteiros (n1 e n2, com n2 > n1) e apresentar todos os números pares existentes entre eles. |
| 2 | Apresentar os dez primeiros caracteres do alfabeto, em maiúsculas (<u>dica</u> : na tabela ASCII, o caracter 'A' corresponde ao valor 65). |
| 3 | Receber (via teclado) um número inteiro qualquer entre 1 e 50 e exibir os números múltiplos de 3 existentes entre esse inteiro e 99 (inclusive). |
| 4 | Receber (via teclado) três valores do tipo int, correspondentes a “horas” (0 a 23), “minutos” (0 a 59) e “segundos” (0 a 59), calcular o valor equivalente em segundos e exibir esse valor. |
| 5 | Receber (via teclado) um valor do tipo int, correspondente a “minutos” (0 a 1439), calcular e exibir o correspondente valor em “horas” (0 a 23) e “minutos” (0 a 59). |

Suporte teórico 2. Uma classe Java permite separar o programa principal (main) das ações que ele realiza. Estas ações podem ser programadas em *métodos*. A estrutura geral de uma classe Java é:

| Código-exemplo | Observações |
|---|--|
| <pre>// Classe Programa (mesmo nome do arquivo .java) public class Programa { // atributos int x; // Construtor para a classe Exemplo public Programa() { x = 5; } // Método para alterar o valor do atributo public void alteraAtributo(int valor) { x = x + valor; } // ponto de início da execução ("programa principal") public static void main(String[] args) { Programa myObj = new Programa(); System.out.println("Antes da alteração: " + myObj.x); myObj.alteraAtributo(10); System.out.println("Depois da alteração: " + myObj.x); } // final do método 'main' } // final da classe 'Programa'</pre> | <p>O nome da classe dever ser o mesmo do arquivo .java.</p> <p>Pode haver mais de um atributo.</p> <p>O método construtor tem o mesmo nome da classe; não retorna valor (nem mesmo void). É útil para inicializar os valores dos atributos.</p> <p>Pode haver vários métodos na classe.</p> <p>Se houver um método main, é aí que a execução terá início.</p> <p>A instanciação de myObj aciona o construtor da classe Programa e permite acessar seu atributos e métodos (note o uso do <i>operador de partição .</i>).</p> |

Atividade 2. Modifique o programa Java da pasta Versao_2_..., deslocando os códigos de cada ação do menu para um método respectivo.

Suporte teórico 3. A separação entre o programa principal (main) e as ações que ele realiza (métodos) pode ser feita utilizando uma classe Java apenas para o main e outra(s) para os métodos.



Atividade 3. Separe o programa Java da pasta Versao_3_... em dois: o primeiro conterá apenas o main, enquanto o outro conterá os *métodos* que realizam as ações previstas..

Atividade 4. Reúna os programas Java da pasta Versao_4_... sob o conceito de um projeto.