

Atividade 05 – PESQUISA: Classes Wrappers

1. Na tecnologia Java, defina o que são e para que servem as Classes Wrappers.

As Classes Wrappers em Java encapsulam tipos primitivos de dados, como `int`, `float`, `double` e `boolean`, em objetos. Cada tipo primitivo tem sua classe correspondente no pacote `java.lang`, fornecendo métodos úteis para trabalhar com esses tipos de dados como objetos. Elas permitem que esses tipos primitivos sejam tratados como objetos em situações onde os objetos são necessários, como em coleções (`ArrayList`, `HashMap`, etc.), manipulação de strings, reflexão e outros recursos que requerem objetos em vez de tipos primitivos. Além disso, as Classes Wrappers fornecem métodos para converter tipos primitivos em strings e vice-versa, realizar operações matemáticas e lógicas, e fornecer constantes úteis e métodos estáticos. Em resumo, elas são usadas para fornecer uma interface orientada a objetos para tipos primitivos em Java.

Essas classes estão localizadas no pacote `java.lang` e são automaticamente importadas em programas Java, eliminando a necessidade de importá-las explicitamente. Por exemplo, a classe `Integer` é um wrapper para o tipo primitivo `int`, e a classe `Double` para `double`. A principal vantagem é a possibilidade de utilizar tipos primitivos em situações onde objetos são esperados, como em métodos que trabalham com a `Collections Framework`. Além disso, fornecem métodos adicionais que não estão disponíveis para os tipos primitivos, como métodos para comparação, conversão e formatação de valores.

2. De forma sucinta, comente a aplicação das seguintes classes:

a. Integer;

A classe Integer é usada para encapsular valores inteiros (int) em objetos. Ela fornece métodos para realizar operações matemáticas, conversão de tipos e manipulação de números inteiros de forma mais flexível. É amplamente aplicada em situações onde valores inteiros precisam ser tratados como objetos, como em coleções genéricas e algoritmos que operam em objetos.

b. Boolean;

A classe Boolean é usada para encapsular valores booleanos (true/false) em objetos. Ela fornece métodos para manipulação de valores booleanos, como operações lógicas e conversão para outros tipos de dados. É comumente utilizada em estruturas de controle condicional, como if e while.

c. Character;

A classe Character é usada para encapsular caracteres individuais em objetos. Ela fornece métodos para manipulação de caracteres, como conversão de maiúsculas para minúsculas, verificação de caracteres alfabéticos, numéricos, etc. É muito utilizada em operações que envolvem caracteres, como manipulação de strings, análise de texto e formatação.

d. Double;

A classe Double é usada para encapsular valores de ponto flutuante (double) em objetos. Ela fornece métodos para realizar operações matemáticas, conversão de tipos e manipulação de números de ponto flutuante. É aplicada em cálculos matemáticos mais precisos e em situações onde números decimais requerem tratamento como objetos.

e. Byte;

A classe Byte é usada para encapsular valores de bytes em objetos. Ela fornece métodos para manipulação de valores de bytes, como operações

aritméticas e conversão para outros tipos de dados. É muito utilizada em operações de baixo nível e manipulação de arquivos binários.

f. Short;

A classe Short é usada para encapsular valores inteiros curtos (short) em objetos. Ela fornece métodos para realizar operações matemáticas, conversão de tipos e manipulação de números inteiros curtos. Assim como o Byte, também é aplicada em operações de baixo nível, mas possui uma faixa maior de valores possíveis.

g. Float;

A classe Float é usada para encapsular valores de ponto flutuante de precisão simples (float) em objetos. Ela fornece métodos para realizar operações matemáticas, conversão de tipos e manipulação de números de ponto flutuante de precisão simples. É aplicada em situações que demandam menor consumo de memória e quando a precisão dupla não é necessária.

h. Long.

A classe Long é usada para encapsular valores inteiros longos (long) em objetos. Ela fornece métodos para realizar operações matemáticas, conversão de tipos e manipulação de números inteiros longos. É aplicada em situações onde números inteiros muito grandes precisam ser tratados como objetos, como timestamps ou IDs extensos.

3. Considere as classes citadas no item “2”. Escolha 3 destas classes e construa um código simples que mostre um exemplo de sua aplicação para elas. Copie e cole o código de teste abaixo.

```
4. public class TesteWrappers {
5.
6.     public static void main(String[] args) {
7.
8.         // Exemplo de uso da classe Integer
9.         Integer quantidadeProdutos = 50;
10.        Integer quantidadeVendida = 20;
11.        Integer restante = quantidadeProdutos - quantidadeVendida;
12.        System.out.println("Estoque restante: " + restante);
13.
14.        // Exemplo de uso da classe Boolean
15.        Boolean temEstoque = restante > 0;
16.        System.out.println("Há estoque disponível? " + temEstoque);
17.
18.        // Exemplo de uso da classe Character
19.        Character letra = 'A';
20.        System.out.println("Letra: " + letra);
21.
22.        // Exemplo de uso da classe Double
23.        Double precoProduto = 99.99;
24.        Double desconto = 0.15;
25.        Double precoComDesconto = precoProduto * (1 - desconto);
26.        System.out.println("Preço: " + precoComDesconto);
27.
28.        // Exemplo de uso da classe Byte
29.        Byte byteValor = 100;
30.        System.out.println("Valor em byte: " + byteValor);
31.
32.        // Exemplo de uso da classe Long
33.        Long distancia = 1000000L;
34.        System.out.println("Distância: " + distancia);
35.    }
36.
37. }
```