JAVA

JAVA – CEFET

Prof. Gustavo Guedes

Atualização: 16/03/2011

Classes Wrapper

- O Java não considera os dados de tipos primitivos como objetos. Por exemplo, os dados numéricos, booleanos e de caracteres são tratados na forma primitiva por questões de eficiência. O Java fornece as classes wrapper para manipular os elementos de dados primitivos como objetos. Tais elementos de dados são “embalados” em um objeto criado ao seu redor. Cada tipo de dado primitivo Java tem uma classe wrapper correspondente no pacote java.lang. Cada objeto da classe wrapper encapsula um único valor primitivo.

Classes Wrapper

Primitive Data Type	Wrapper Class
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

Classes Wrapper

- Resumindo, Wrapper classes é uma designação para as classes do java que encapsulam os tipos primitivos, ou seja, (Integer, Double, Character, Short, etc). Vale ressaltar que Wrapper (invólucro, empacotador) é uma designação genérica para todo o objeto que encapsula outro ou para todo objeto que encapsula um tipo primitivo, mas no contexto atual, estamos utilizando o conceito para os tipos primitivos.

- O Integer, por exemplo, é um objeto que contém um int. O Integer atua como uma caixa (box) para o int que está lá dentro. Se você quiser criar uma lista de int, não consegue, pois List só permite conter objetos e int não é um objeto. Então, os wrappers servem para utilizar os primitivos como objetos sempre que isso é necessário (como colocar em uma lista, por exemplo).

Wrapper classes

- List lista = new ArrayList();
- lista.add(new Integer(2));

- Integer i1 = (Integer) lista.get(0);
- int i2 = i.intValue();

- O java 1.5 trouxe o conceito de autoboxing e unboxing (auto-encaixotamento e auto-desencaixotamento).

Wrapper classes

- List lista = new ArrayList();
- lista.add(2); //encaixota “2” no wrapper para utilizar como objeto.
- int i2 = (Integer)lista.get(0);
- Integer x = 2; //ok

Wrapper classes

- …main{
- Integer x = 5;
- Integer y = 5;
- teste(y);
- syso(y);
- syso(x);
- }
- public static void teste (Integer x) {
- x=7;
- }

Wrapper classes

- List lista = new ArrayList();
- lista.add(2); //encaixota 2 no wrapper para utilizar como objeto.
- int i2 = (Integer)lista.get(0);

- Existem métodos nessas classes que podem ser utilizados para transformação de números sendo representados como String para os respectivos primitivos, ex:
- Integer.parseInt("1000") que recebe como argumento uma String, retorna essa String como int. Assim como Double.parseDouble("15.5") retorna a String "15.5" como 15.5, ou seja, o double representado pela String.

Dicas e exercícios

- String str = "*Para ser um campeão*, *você tem* que acreditar em si mesmo quando ninguém mais acredita."

Crie uma classe Utils com um método chamado contaPalavras que recebe como argumento uma String. Conte o número de palavras que existe no texto e retorne um int representado essa quantidade. Dica: utilize o split.

Crie um outro método chamado verificaQuando que recebe como argumento uma String. Esse método verifica se existe a palavra "quando" nesse texto. Dica: utilize o indexOf e o toLowerCase.

Crie um outro método chamado verificaSite que recebe como argumento uma String. Esse método deve retornar true caso o texto comece com "http". Dica: utilize o startsWith.

Crie um novo método chamado verificaExtensao que recebe um String e retorna true caso a String termine com "jpg". Dica: utilize o endsWith.

Crie um novo método chamado comparacao que recebe 2 strings como argumento e retorna um int representando quantos caracteres existem iguais nas mesmas posiçoes, ex: amarelo, amxrelo retornaria 6. casa e asac retornaria 0. Dica: use toCharArray().

Dicas e exercícios

- String str = "*Para ser um campeão*, *você tem* que acreditar em si mesmo quando ninguém mais acredita."

Ainda no exemplo acima, podemos utilizar o método replaceAll("você", "nós") para sobrescrever toda palavra "você" pela palavra "nós". Isso retorna uma nova String.

Dessa forma, ao fazer

String str2 = str. replaceAll("você", "nós")

System.out.println(str); Imprime:

"*Para ser um campeão*, *você tem* que acreditar em si mesmo quando ninguém mais acredita."

System.out.println(str); Imprime:

"*Para ser um campeão*, *nós tem* que acreditar em si mesmo quando ninguém mais acredita."

Dicas e exercícios

- String str = "*A casa amarela*."

Podemos utilizar o método str.substring(2, 4). Esse método me retorna uma nova string contendo os caracteres da posição 2 até a 4, exclusivo, ou seja, a posição 4 não entra.

Saída no console: "ca"

Isso pode ser utilizado em conjunto com o indexOf(string). Dessa forma, ao dizer str.indexOf("casa"), temos o retorno 2.