

Atividade 05 – PESQUISA: Classes Wrappers

Discente: Marlene Vasconcelos Moraes de Oliveira

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

As Classes Wrappers são classes que representam os tipos primitivos no Java, a fim de tratá-los como objetos e encapsular um dado primitivo em um objeto de classe Wrapper. Isso permite que o uso dos tipos de dados primitivos em locais onde apenas objetos são aceitos, como em coleções.

2. De forma sucinta, comente a aplicação das seguintes classes:
 - a. Integer: classe wrapper para o tipo primitivo int, que fornece métodos para manipular os valores inteiros de 2^{31} até $2^{31}-1$.
 - b. Boolean: classe wrapper para o tipo primitivo boolean, que fornece métodos para manipular os valores booleanos (verdadeiro e falso).
 - c. Character: classe wrapper para o tipo primitivo char, que fornece métodos para manipular caracteres.
 - d. Double: classe wrapper para o tipo primitivo double, que fornece métodos para manipular os valores de ponto flutuante de uma precisão de 64 bit.
 - e. Byte: classe wrapper para o tipo primitivo byte, que fornece métodos para manipular os valores inteiros de -128 até 127.
 - f. Short: classe wrapper para o tipo primitivo short, que fornece métodos para manipular os valores inteiros de -32768 até 32768.
 - g. Float: classe wrapper para o tipo primitivo float, que fornece métodos para manipular os valores de ponto flutuante de uma precisão de 32 bit.
 - h. Long: classe wrapper para o tipo primitivo long, que fornece métodos para manipular os valores inteiros de 2^{63} até $2^{63}-1$.
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.

Obs.: Não é preciso fazer exemplos complexos; todos eles podem ser feitos dentro de um mesmo método main.

```
public class Main {
    public static void main(String[] args) {
        String[] dados = { "Marlene", "29", "1234.56", "true" };

        Pessoa pessoa = new Pessoa();
        pessoa.setNome(dados[0]);
        pessoa.setIdade(Short.parseShort(dados[1]));
        pessoa.setSalario(Double.valueOf(dados[2]));
        pessoa.setMaior_18(Boolean.parseBoolean(dados[3]));

        System.out.println(pessoa.toString());
    }
}

// Objeto Pessoa

//Optei por encurtar o num de linhas para colar aqui

class Pessoa {
    private String nome;
    private short idade;
    private double salario;
    private boolean maior_18;

    public Pessoa() { }

    public String getNome() { return nome; }

    public void setNome(String nome) { this.nome = nome; }

    public short getIdade() { return idade; }

    public void setIdade(short idade) { this.idade = idade; }

    public double getSalario() { return salario; }

    public void setSalario(double salario) { this.salario = salario; }

    public boolean isMaior_18() { return maior_18; }

    public void setMaior_18(boolean maior_18) { this.maior_18 = maior_18; }

    @Override
    public String toString() { return "[Pessoa]\nNome: " + getNome() + "\nIdade: " +
    getIdade() + "\nSalario: " + getSalario() + "\nMaior de idade: " + isMaior_18(); }
}
```