



Compartilhar o seu link com: luciorocha @ professores.utfpr.edu.br

Mabyly <  Cópia de POCO4A - Aula 5 - Exercícios >

Roberto Furlani Neto:  Cópia de POCO4A - Aula 5 - Exercícios

Pedro Lucas dos Reis e Thiago Tieghi:  Cópia de POCO4A - Aula 5 - Exercícios

João Pedro de Paula:<  Cópia de POCO4A - Aula 5 - Exercícios >

João Pedro Cavani Meireles<  Cópia de POCO4A - Aula 5 - Exercícios >

Luis Mendes e Alexandre Aparecido: <  Cópia de POCO4A - Aula 5 - Exercícios >

Guilherme Ramalho:  Cópia de POCO4A - Aula 5 - Exercícios

Rafael Kendy Naramoto Lopes <  Cópia de POCO4A - Aula 5 - Exercícios >

Henrique Cois <  Cópia de POCO4A - Aula 5 - Exercícios >

Thales Alves <  Aula 5 - Exercícios >

Deivid da Silva Galvao <  Cópia de POCO4A - Aula 5 - Exercícios >

Séfora e Carlos  Cópia de POCO4A - Aula 5 - Exercícios

Filipe Augusto Parreira Almeida  Cópia de POCO4A - Aula 5 - Exercícios

Marcos Tadao Shoji  Cópia de POCO4A - Aula 5 - Exercícios

Matheus Hirata Vanzela <  Cópia de POCO4A - Aula 5 - Exercícios >

João Vitor N. Yoshida <  Cópia de POCO4A - Aula 5 - Exercícios >

Daniel Martins de Carvalho <  Cópia de POCO4A - Aula 5 - Exercícios >

Isabella Melo Almeida:  Aula 5

Thiago Cristovão de Souza and Matheus Mazali Maeda: <
 Cópia de POCO4A - Aula 5 - Exercícios >

Raphael Uematsu  Cópia de POCO4A - Aula 5 - Exercícios

Julio Farias<  Cópia de POCO4A - Aula 5 - Exercícios >

Angélica <  Cópia de POCO4A - Aula 5 - Exercícios >

Felipe F B Lorusso <[Aula 5 - Exercícios](#)>

Gabriel Takeshi/Plinio<  Cópia de POCO4A - Aula 5 - Exercícios >

Gustavo Nunes  Cópia de POCO4A - Aula 5 - Exercícios

Alexandre Olah  Aula 5 - Exercícios

Felipe Antonio Magro:  Cópia de POCO4A - Aula 5 - Exercícios

Vitor Hugo Leite Arruda de Oliveira <  Cópia de POCO4A - Aula 5 - Exercícios >

Rodrigo Leandro Benedito:  Cópia de POCO4A - Aula 5 - Exercícios

Victor Ramos Bernardes  Cópia de POCO4A - Aula 5 - Exercícios

Lucas Viana:  Cópia de POCO4A - Aula 5 - Exercícios

- 1) (Online) Acesse o link: <https://codeboard.io/projects/341634>
 - a) TODO1: Classe Estudante: Crie um construtor sobrecarregado que aceite 2 (dois) argumentos diferentes.
 - b) TODO2: Classe Estudante: Crie 2 (dois) métodos sobrecarregados que imprimam o nome do estudante.

- c) TODO3: Classe Estudante: Utilize o método toString para imprimir o estado de execução do objeto.
- d) TODO4: Classe Estudante: Crie um atributo idade de acesso public. Ilustre a chamada desse atributo na classe Principal.
- e) TODO5: Classe Estudante: Crie um atributo endereco de acesso private. Ilustre a chamada desse atributo na classe Principal.

```
public class Estudante {
    private String nome;
    private String RA;
    public int idade;
    private String endereco;

    public Estudante(){
        this.nome="";
        this.RA="";
        this.idade=0;
        this.endereco="";
    }
    public Estudante(String nome, String RA){
        this.nome=nome;
        this.RA = RA;
        this.idade=0;
        this.endereco="";
    }
    public Estudante(String nome, String RA, int idade){
        this.nome=nome;
        this.RA=RA;
        this.idade=idade;
        this.endereco="";
    }
    }

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

    public void setEndereco(String endereco){ this.endereco = endereco; }

    // @Override
    public String toString(){
        return this.nome + " " +
            this.RA + " " +
            this.idade + " " +
            this.endereco;
    }
}
```

```

}

//-----
/*
DONE1: Classe Estudante: Crie um construtor sobrecarregado que
    aceite 2 (dois) argumentos diferentes.
DONE2: Classe Estudante: Crie 2 (dois) métodos sobrecarregados que modificam
    o atributo nome.
DONE3: Classe Estudante: Utilize o método toString para imprimir o
    estado de execução do objeto.
DONE4: Classe Estudante: Crie um atributo idade de acesso public.
    Ilustre a chamada desse atributo na classe Principal.
DONE5: Classe Estudante: Crie um atributo endereco de acesso private.
    Ilustre a chamada desse atributo na classe Principal.
*/

public class Principal {

    public Principal(){

    }

    public static void main(String[] args) {
        new Principal();
        Estudante aluno1 = new Estudante("CARLOS","2271931",21);
        Estudante aluno2 = new Estudante();
        Estudante aluno3 = new Estudante("CARLOS","2271931");
    }
}

```

```
}
```

- 2) (NetBeans) Utilize o código-fonte do exercício anterior. No mesmo projeto:
- Implemente uma Classe Temperatura que realize a conversão de temperatura de graus Celsius para graus Fahrenheit, e vice-versa, de acordo com as seguintes fórmulas:

$$\text{celsius} = 5.0 / 9.0 * (\text{fahrenheit} - 32)$$

$$\text{fahrenheit} = 9.0 / 5.0 * \text{celsius} + 32$$

- 3) (NetBeans) Utilize o código-fonte do exercício anterior. No mesmo projeto:
- Implemente uma Classe Notas que leia 3 (três) notas e retorne a menor nota. Dica: utilize o método static Math.min

- 4) (NetBeans) Utilize o código-fonte do exercício anterior. No mesmo projeto:
- Implemente uma Classe Sorteio que gere um número inteiro aleatório. A seguir, receba um número do usuário e informe se o número recebido é maior, menor ou igual ao número recebido.

```
SecureRandom numeroAleatorio = new SecureRandom();  
  
resultado = MIN + numeroAleatorio.nextInt(MAX);
```

- 5) (NetBeans) Utilize o código-fonte do exercício anterior. No mesmo projeto, implemente:
- a) Classe Principal: implemente um método que meça a temperatura de 5 (cinco) alunos.
 - b) Classe Principal: implemente um método que imprima a menor nota dos 5 (cinco) alunos.
 - c) Classe Principal: implemente um método que escolha aleatoriamente um dos 5 (cinco) alunos de acordo com o seu número de matrícula (inteiro). A seguir, mostre a temperatura e a soma das notas do aluno escolhido.