

Nome: Emerson Serrou Camy  
CPF: 034824681-17  
Curso: Ciência da Computação  
Semestre: 4º

### GABARITO ATIVIDADE 03

#### Questão 01

import java.util.Date;

// Declaração da Classe Abstract BasePessoa.

abstract class BasePessoa

```
{  
    private int codigo;  
    private String nome;  
    private String endereco;  
    private String telefone;  
    private Date dataNascimento;  
    private String rg;  
    private String cpf;  
    private Date dataInsercao;
```

// Construtor da classe BasePessoa, com os atributos solicitados.

public BasePessoa(int codigo, String nome, String endereco, String telefone, Date dataNascimento, String rg, String cpf, Date dataInsercao)

```
{  
    this.codigo = codigo;  
    this.nome = nome;  
    this.endereco = endereco;  
    this.telefone = telefone;  
    this.dataNascimento = dataNascimento;  
    this.rg = rg;  
    this.cpf = cpf;  
    this.dataInsercao = dataInsercao;  
}  
}
```

//Classe derivada Professor, deriva de BasePessoa.

class Professor extends BasePessoa

```
{  
    private String registro;  
    private Date dataContratacao;  
    //Construtor da classe Professor.  
    //Chama o construtor da classe base e inicializa os atributos adicionais.  
    public Professor(int codigo, String nome, String endereco, String telefone, Date dataNascimento, String rg, String cpf, Date dataInsercao, String registro, Date dataContratacao)  
    {  
        super(codigo, nome, endereco, telefone, dataNascimento, rg, cpf, dataInsercao);  
        this.registro = registro;  
        this.dataContratacao = dataContratacao;
```

```

    }
}

//Classe derivada Aluno, deriva de BasePessoa.
class Aluno extends BasePessoa
{
    private String matricula;
    private Date dataMatricula;
    //Construtor da classe Aluno.
    //Chama o construtor da classe base e inicializa os atributos adicionais.
    public Aluno(int codigo, String nome, String endereco, String telefone, Date
dataNascimento, String rg, String cpf, Date dataInsercao, String matricula, Date
dataMatricula)
    {
        super(codigo, nome, endereco, telefone, dataNascimento, rg, cpf, dataInsercao);
        this.matricula = matricula;
        this.dataMatricula = dataMatricula;
    }
}

//Classe principal.
public class Main
{
    public static void main(String[] args)
    {
        Date dataAtual = new Date();

        // Instanciando um objeto Professor e um objeto Aluno.
        Professor professor = new Professor(1, "Luizão", "Av. Ceara 1234", "67-1234-5678",
dataAtual, "1234567", "123456789", dataAtual, "01", dataAtual);
        Aluno aluno = new Aluno(2, "Emerson", "Mato Grosso 5678", "67-8765-4321",
dataAtual, "7654321", "987654321", dataAtual, "02", dataAtual);
    }
}

```

#### Questão 02

Alternativa B.

Justificativa – De acordo com o livro How to Program Java, de Paul Deitel e Harvey Deitel, a alternativa B está correta, porque a propriedade ID da classe Produto está com apenas um getter e não possui um setter definido, fazendo com que o valor do ID não possa ser alterado após a inicialização do objeto, pois a torna uma propriedade somente de leitura.

#### Questão 03

A - Verdadeiro, porque a classe Calculadora é uma classe pública, que pode ser instanciada em outras classes usando a linha de código “Calculadora calc = new Calculadora ()”, sendo que o objeto calc é uma instancia da classe Calculadora e pode ser utilizado em operações matemáticas e chamar seus métodos.

B - Verdadeiro, pois a palavra reservada abstract é utilizada em Java para indicar que uma classe ou método é abstrato, portanto não influencia na funcionalidade da classe, pois no código fornecido essa palavra-chave não é utilizada e a classe Calculadora funciona corretamente.

C - Falso, porque o código fornecido tem seus métodos concretos (que não são declarados como abstract) e implementados com corpo, mas se fossem declarados como abstract, o compilador apresentaria algum erro, visto que métodos abstract não possuem uma implementação completa.

D - Falso, porque o uso da palavra abstract indica que o método ou classe não pode ser instanciado diretamente, no código fornecido a classe Calculadores e seus métodos não estão marcados como abstract, por isso que eles podem ser chamados diretamente e também podem ser instanciados.

E - Falso, porque os métodos foram implementados corretamente, respeitando a sintaxe adequada para que as operações matemáticas sejam realizadas.

#### Questão 04

Alternativa C.

Justificativa – Porque de acordo com o livro "Programação Orientada a Objetos", de Robert Lafore, um objeto é definido como "uma entidade que possui um estado e um conjunto definido de operações definidas para funcionar nesse estado".

#### Questão 05

Alternativa B.

Justificativa – Porque de acordo com o livro "Programação Orientada a Objetos" de Robert Lafore, é afirmado que "a reutilização de código é uma das principais vantagens da programação orientada a objetos", o autor explica que a principal vantagem do uso de classes, é que elas podem ser potencialmente reutilizáveis, podendo ser utilizadas em diferentes projetos e aplicações.