Nome: Marcos Marton Magalhães
CPF: 065.952.401-57
Curso: Engenharia da Computação
Semestre: 10° Semestre

GABARITO ATIVIDADE 03

Questão 01

```
import java.util.LocalDate;

public abstract class BasePessoa {
    protected int codigo;
    protected String nome;
    protected String endereco;
    protected String telefone;
    protected LocalDate dataNascimento;
    protected String rg;
    protected String cpf;
    protected LocalDate dataInsercao;

    public BasePessoa(int codigo, String nome, String endereco, String telefone, LocalDate
dataNascimento, String rg, String cpf, LocalDate 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;
  }

    public abstract String obterInformacoes();
}

public class Professor extends BasePessoa {
    private String registro;
    private LocalDate dataContratacao;

    public Professor(int codigo, String nome, String endereco, String telefone, LocalDate
dataNascimento, String rg, String cpf, Date dataInsercao, String registro, LocalDate
dataContratacao) {
        super(codigo, nome, endereco, telefone, dataNascimento, rg, cpf, dataInsercao);
        this.registro = registro;
        this.dataContratacao = dataContratacao;
```

```
    }

    public String obterInformacoes() {
        return "Professor - Nome: " + nome + ", Registro: " + registro + ", Contratação:       " +
dataContratacao;
    }
}

public class Aluno extends BasePessoa {
    private String matricula;
    private LocalDate dataMatricula;

    public Aluno(int codigo, String nome, String endereco, String telefone, LocalDate
dataNascimento, String rg, String cpf, LocalDate dataInsercao, String matricula, LocalDate
dataMatricula) {
        super(codigo, nome, endereco, telefone, dataNascimento, rg, cpf, dataInsercao);
        this.matricula = matricula;
        this.dataMatricula = dataMatricula;
    }

    public String obterInformacoes() {
        return "Aluno - Nome: " + nome + ", Matrícula: " + matricula + ", Data de Matrícula: " +
dataMatricula;
    }
}

public class Main {
    public static void main(String[] args) {
        Professor professor = new Professor(1, "João Silva", "Rua A", "123456789", new
LocalDate.of(1980, 5, 10), "1234567", "12345678901", new LocalDate.of(2023, 8, 23), "123",
new LocalDate.of(2020, 1, 15));
        Aluno aluno = new Aluno(2, "Maria Santos", "Rua B", "987654321", new
LocalDate.of(2000, 3, 20), "7654321", "98765432101", new LocalDate.of(2023, 8, 23), "456",
new LocalDate.of(2023, 2, 10));

        System.out.println(professor.obterInformacoes());
        System.out.println(aluno.obterInformacoes());
    }
}
```

Questão 02

Alternativa B

Verdadeiro. A propriedade '*setId()*' na classe '*Classe*' pode não existir, pois ela não é definida explicitamente na classe '*Classe*', mas sim na classe '*BaseIdentificador*'.

Questão 03

Alternativa A
Verdadeiro. O programador pode criar uma instância da classe Calculadora.

Alternativa B
Verdadeiro. A classe atual não possui nenhum método marcado como abstract.

Alternativa C
Verdadeiro. Os métodos da classe podem ser declarados como abstract, o compilador não emitirá um erro, desde que a classe seja marcada como abstract também.

Alternativa D
Falso. Quando uma classe é declarada como abstract, isso indica que ela não pode ser instanciada diretamente.

Alternativa E
Falso. A classe parece estar escrita corretamente em termos de sintaxe e funcionalidade.

Questão 04
Alternativa C

Um objeto é uma instância de uma classe e representa uma entidade do mundo real ou conceitual. Ele tem um estado, que é determinado pelos valores de seus atributos, e um conjunto de métodos (operações) definidos pela classe que podem ser invocados para interagir com o objeto e alterar seu estado.

Questão 05
Alternativa B

Uma das principais vantagens da programação orientada a objetos é a capacidade de reutilizar código através do conceito de herança e criação de classes. Quando você define uma classe, ela pode ser usada como base para criar outras classes derivadas, aproveitando a funcionalidade existente e estendendo-a conforme necessário.