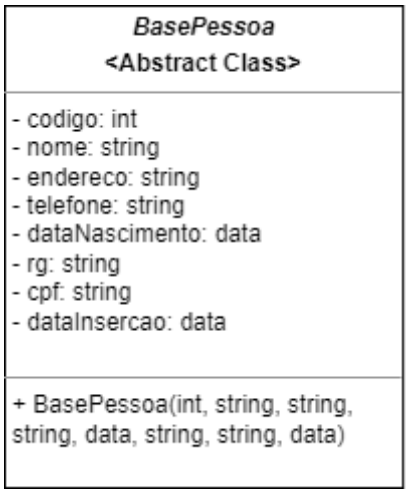
UNIVERSIDADE ANHANGUERA-UNIDERP
UNIDADE MATRIZ
Avenida Ceará nº 333 – Bairro Miguel Couto
Campo Grande - MS, CEP 79003-010, Telefone: (67) 3348-8000

PROGRAMAÇÃO ORIENTADA A OBJETOS

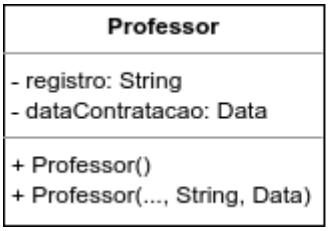
ATIVIDADE 03

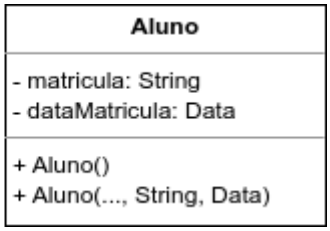
QUESTÃO 01

Crie um programa que atenda a especificação dos diagramas abaixo.



As classes abaixo devem ser derivadas da classe BasePessoa. Atenção para o construtor das classes Professor e Aluno.





QUESTÃO 02

Considere o trecho de código abaixo.

```
public class Classe
{
	privateint id;
	publicint ID { get { return this.id; } set { this.id = value; } }
	public Classe() { }
}

public class Subclasse
{
	private int id;
	public int ID { get { return this.id; } set { this.id = value; } }

	private Classe classe;
	public Classe Classe { get { return this.classe; } set { this.classe =
value; }	}
	public Subclasse() : base() { this.classe = new Classe(); }
}

public class Produto
{
	private int id;
	public int ID { get { return this.id; } }

	private Subclasse subclasse;
	public Subclasse Subclasse { get { return this.subclasse; } set {
this.subclasse = value; } }
	public Produto(int chave) { this.id = chave; }
}
```

Marque a alternativa correta nas afirmações abaixo.

A) Todas as instâncias de classe são inicializadas corretamente.
B) A propriedade ID da classe Produto, por não ter set, pode ser considerada somente leitura.
C) É correto considerar o construtor da classe "Subclasse" como compilável, segundo os conceitos de Orientação a Objeto.
D) É correto afirmar que a classe "Produto" herda atributos e propriedades das outras classes, devido ao relacionamento associativo.
E) O código está semanticamente correto, e compilará sem erros ou avisos.

QUESTÃO 03

Observe o código abaixo, e marque Verdadeiro ou Falso nas afirmações a seguir.

```
public class Calculadora
{
	public int Somar(int numero1, int numero2)
	{
		return numero1 + numero2;
	}

	public int Subtrair(int numero1, int numero2)
	{
		return numero1 - numero2;
	}

	public int Multiplicar(int numero1, int numero2)
	{
		return numero1 * numero2;
	}

	public int Dividir(int numero1, int numero2)
	{
		return numero1 / numero2;
	}
}
```

A) Se necessário, o programador pode criar uma instância dessa classe, facilitando assim sua operação, bastando adicionar a outra classe a linha de código:
`Calculadora calc = new Calculadora()`
B) Um possível uso da palavra reservada *abstract* não influencia a funcionalidade da classe.
C) Os métodos poderiam ser declarados como *abstract*. Assim como está, o compilador irá informar que o código está errado.
D) O uso da palavra *abstract*, tanto na classe como nos métodos, representa que a mesma pode ser chamada diretamente, sem que seja necessário criar uma instância para utilizá-la.
E) Baseado na semântica e funcionalidade do código, a classe está escrita incorretamente, e compilará com erros e avisos.

QUESTÃO 04

Considerando a definição de um objeto, na Programação Orientada a Objetos, marque a alternativa correta nas afirmações que explicam a definição.

A) Um objeto é uma rotina de programação contida em uma classe que pode ser chamada diversas vezes possibilitando assim reuso de código de programação.
B) Um objeto é um conjunto de atributos primitivos tipados contido em uma classe.
C) Um objeto é uma entidade que possui um estado e um conjunto definido de operações definidas para funcionar nesse estado.
D) Um objeto é um elemento de uma classe que representa uma operação (a implementação de uma operação).
E) Um objeto é uma porção de código que resolve um problema muito específico, parte de um problema maior.

QUESTÃO 05

O aumento da produtividade de desenvolvimento e a capacidade de compartilhar o conhecimento adquirido, representa uma vantagem no uso de projetos orientados a objeto.

Marque a alternativa correta nas afirmações que explicam esse fato.

A) um objeto pode ser chamado por objetos de classe diferente da sua.
B) as classes podem ser potencialmente reutilizáveis.
C) as classes devem ser concretas ou abstratas.
D) todo método pode ser derivado naturalmente das operações de sua classe.
E) o encapsulamento impossibilita equívocos de código.