



Disciplina Programação Modular	Departamento Ciência da Computação	Turno Manhã	Período 9º
Professor Hugo de Paula (hugo@pucminas.br)			
Matrícula:	Aluno:		

## Exercício 2

1. Considere o código abaixo:

```
public class Instanciadora
{
    private static int id = 0;
    private String tipoInstancia;

    public Instanciadora(String tipoRequerido)
    {
        id = id + 1;
        tipoInstancia = tipoRequerido;
    }

    public String toString()
    {
        return "Tipo: " + tipoInstancia + ", Id: " + id + "; ";
    }

    public static void main(String [] args)
    {
        Instanciadora[] instancias = new Instanciadora[2];
        instancias[0] = new Instanciadora("Prod");
        instancias[1] = new Instanciadora("Obj");
        for (int i = 0; i < 2; i++)
            System.out.print(instancias[i]);
        System.out.println();
    }
}
```

Assinale a saída do programa:

- (a) Tipo: Prod, Id: 0; Tipo: Obj, Id: 1;                      (d) Tipo: Prod, Id: 2; Tipo: Obj, Id: 2;  
(b) Tipo: Prod, Id: 1; Tipo: Prod, Id: 2;  
(c) Tipo: Prod, Id: 1; Tipo: Obj, Id: 2;                      (e) Tipo: Obj, Id: 2; Tipo: Obj, Id: 2;
2. Instâncias de classes são inicializadas por construtores. Assinale a afirmação correta sobre construtores de classes:

- (a) Uma classe pode ter mais de um construtor, somente se eles possuírem diferentes números de argumentos.

Matrícula:	Aluno:
------------	--------

- (b) Uma classe pode ter mais de um construtor, mas apenas um deles pode ser do tipo **void**.
- (c) Uma classe pode ter mais de um construtor, desde que eles possuam um número diferente de argumentos, ou os argumentos possuam tipos diferentes.
- (d) Uma classe pode ter mais de um construtor, desde que todos possuam nomes diferentes.
- (e) Uma classe pode ter mais de um construtor, desde que eles sejam públicos e não estáticos.

3. Utilizando o TDD, atualize a classe **Data** desenvolvida no Exercício 1 para que seja possível escrever a data por extenso, a partir do método:

```
public String porExtenso()
```

Por exemplo, (**new** Data(15/11/1889)).porExtenso() deve retornar **15 de novembro de 1889**. Faça o uso adequado de atributos estáticos e finais. Escreva a nova classe Data na notação UML.

4. No Exercício 1 foi solicitado que se desenvolvesse uma classe **Conta** para representar uma conta bancária. Ela deveria possuir no mínimo o número da conta e o saldo, além de outros atributos que se julgassem necessários. Utilizando a abordagem de desenvolvimento dirigido por teste, complemente a classe **Conta** desenvolvida anteriormente, adicionando aos atributos um identificador contador, utilizando membros estáticos. Lembre-se de encapsular os dados. Desenvolva uma classe aplicação para ilustrar o funcionamento da classe Conta. Atualize a notação UML da classe Conta.
5. Apesar do método destrutor **public void** finalize() poder ser utilizado para resolver pendências como um arquivo aberto ou uma conexão de banco de dados não finalizada, porque não devemos nos basear no destrutor para liberar recursos limitados de armazenamento ou de acesso a banco?