

**P3**PUC/RS – Faculdade de Informática – LAPRO II
Prof. Marcelo Cohen Turma: 128 Data: 20/11/2000**Nome legível do aluno(a):** _____

1. (2,5 pt) Escreva uma classe para representar um veículo em movimento. O construtor deverá receber os seguintes dados: descrição, velocidade (m/s), peso (kg). Crie métodos para setar e retornar cada um dos dados. Faça consistência de todos os dados de entrada, usando tratamento de exceções (inclusive no construtor). O tratamento deve simplesmente exibir uma mensagem de erro na tela.

```
class Veic
{
    private String descr;
    private float vel,peso;

    public Veic(String descr, float vel, float peso)
    {
        try
        {
            this.descr = descr;
            this.vel = vel;
            this.peso = peso;
        }
        catch (Exception e)
        {
            System.out.println("Erro nos dados de entrada!");
        }
    }

    void setDescr(String descr)
    {
        try
        {
            this.descr = descr;
        }
        catch (Exception e)
        {
            System.out.println("Erro!");
        }
    }

    // idem para velocidade e peso

    String getDescr()
    {
        return descr;
    }

    // idem para velocidade e peso
}
```

2. (1,5 pt) Reescreva a classe acima, de forma que as exceções não sejam tratadas aqui, mas sim passadas para quem chamou cada método.

```
class Veic2
{
    // mesmas declarações de dados
```

```

public Veic2(String descr, float vel, float peso) throws Exception
{
    this.descr = descr;
    this.vel = vel;
    this.peso = peso;
}

public setDescr(String descr) throws NullPointerException
{
    this.descr = descr;
}

// idem para os demais
}

```

3. (2 pt) Escreva uma classe que receba três descrições (strings) via linha de comando (através do args[]) e instancie 3 veículos (com a classe criada na questão 2), tratando as possíveis exceções. Se não forem digitadas as 3 descrições, deverá ser gerada uma exceção para o programa chamador (usualmente, o prompt do MS-DOS ou o Shell).

```

class TestaVeic
{
    public void main(String args[]) throws IndexOutOfBoundsException
    {
        try
        {
            Veic2 v1 = new Veic2(args[0],10,20);
            Veic2 v2 = new Veic2(args[1],20,30);
            Veic3 v3 = new Veic2(args[2],30,40);
        }
        catch (IndexOutOfBoundsException e)
        {
            System.out.println("Faltam parametros!!!");
            throw (e);
        }
        catch (Exception e)
        {
            System.out.println("Erro nos argumentos de criação!");
        }
    }
}

```

Responda as questões 4 e 5 considerando o código das classes a seguir:

```

import java.io.*;
import java.io.Serializable;

class Dados implements Serializable
{
    private int dado;

    public Dados(int n)
    {
        dado = n;
        System.out.println(dado);
    }

    public void imp(int x)
    {
        if (x == dado) System.out.println("Dado = "+dado);
        else System.out.println("Não é esse");
    }
}

```

```

class TestaSerializable
{
    public static void main(String args[]) {
        try
        {
            FileOutputStream f = new FileOutputStream("teste");
            ObjectOutputStream os = new ObjectOutputStream(f);
            for(int i=0; i<5; i++)
                os.writeObject(new Dados(i*10));
            os.close();
        }
        catch(IOException e)
        {
            System.out.println("Erro de gravacao");
            System.exit(0);
        }
        Dados aux;
        ObjectInputStream is=null;
        try {
            FileInputStream f = new FileInputStream("teste");
            is = new ObjectInputStream(f);
            while (true) {
                aux = (Dados) (is.readObject());
                aux.imp(20);
            }
        } catch(ClassNotFoundException e) {
            System.out.println("ERRO DE FORMATO");
            System.exit(0);
        } catch(EOFException e) {
            try {
                if (is != null) is.close();
                System.out.println("Fim");
            } catch(IOException x){ }
        } catch(IOException e) {
            System.out.println("ERRO DE LEITURA");
            System.exit(0);
        }
    }
}

```

4. (2 pt) Qual das alternativas abaixo apresenta o resultado **correto** que será exibido na tela do terminal após a execução do programa?

- | | | | | |
|---------------|------------|------------|------------|------------|
| a) Não é esse | b) 0 | c) 0 | d) 0 | e) 0 |
| Não é esse | 10 | 10 | 10 | 10 |
| Dado = 20 | 20 | 20 | 20 | 20 |
| Não é esse | 30 | 30 | 30 | 30 |
| Não é esse | 40 | 40 | 40 | 40 |
| | Não é esse | Não é esse | 0 | 0 |
| | Não é esse | Não é esse | Não é esse | Não é esse |
| | Dado = 20 | Dado = 20 | 10 | 10 |
| | Não é esse | Não é esse | Não é esse | Não é esse |
| | Não é esse | Não é esse | 20 | 20 |
| | | Fim | Dado = 20 | Dado = 20 |
| | | | 30 | 30 |
| | | | Não é esse | Não é esse |
| | | | 40 | 40 |
| | | | Não é esse | Não é esse |
| | | | | Fim |

5. (2 pt) Sobre o programa apresentado é possível afirmar:

- I- O programa se utiliza do recurso de serialização para armazenar 5 instâncias da classe *Dado* em 5 arquivos diferentes.
- II- O programa encerra sua execução normalmente quando ocorre uma exceção do tipo *EOFException*.

III- Quando ocorre um erro de leitura no arquivo, a classe exibe a mensagem “*ERRO DE FORMATO*”.

Estão corretas as alternativas:

- a) Apenas a I
- b) Apenas a I e a II
- c) Apenas a I e a III
- d) Apenas a II e a III
- e) Todas