



UAST

Unidade Acadêmica
de Serra Talhada - PE

Reitor 2016



MPOO

site: <https://sites.google.com/site/profricodemery/mpoo>

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery

Data: 04 / 04 / 2022

Nota: _____

Aluno: _____

1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA) – PARTE 1

Instruções gerais:

Utilize o diretório D: para salvar as implementações. Salve as implementações a cada modificação, caso aconteça alguma falha de energia o trabalho será preservado. Lembre-se que uma vez removido o arquivo do eclipse, seu conteúdo será perdido.

A prova é prática e deverão ser devolvidas tanto a prova impressa quanto os códigos-fonte implementados em Java. A Nota máxima desta prova é de 10,0 pontos.

Pontuação da Prova: (i) Projeto Eclipse e executável: 1,0 ponto; (ii) Classe, atributos, métodos construtor e acesso, encapsulamento: 2,0 pontos; (iii) Herança e Polimorfismo de Objetos: 1,5 ponto; (iv) instâncias e chamadas de métodos: 1,0 ponto; (v) Definição de métodos e suas implementações: 3,0 pontos, (vi) Cardinalidade por ArrayList: 1,0 ponto, (vii) Dependência: 0,5 ponto

1) No Eclipse limpe todos os projetos existentes.

- Crie um novo projeto chamado **NomeSobrenome**, o qual deverá ter uma pasta de pacotes chamada **sistemaPetShop** contendo todos os arquivos necessários para as respectivas questões.

- Ao finalizar a prova **exporte o projeto** contendo toda a codificação do projeto (arquivos texto e *bytecodes*) e envie pelo AVA.

2) O sistema descrito abaixo é modelado no APÊNDICE A, deverá ser implemente em Java.



a) É descrição do sistema:

- Observe os encapsulamentos dos atributos.
- Observe os relacionamentos entre as classes.
- Os animais do sistema estão armazenados em uma de BaseDados. Faça o devido uso de ArrayList.
 - o sistema deverá usar os métodos de ArrayList nos métodos de BaseDados
- Observe os métodos do diagrama de classes:
 - Quando necessário, faça a utilização de métodos já definidos. Não se deve realizar redundância de codificação.
 - Um mesmo animal pode ser adicionado diversas vezes, mas a princípio assume-se que são indivíduos diferentes. Cada indivíduo é indicado por um id do pet shop.
 - Utilize o método `toString()` de Animal para retornar os dados de um animal.

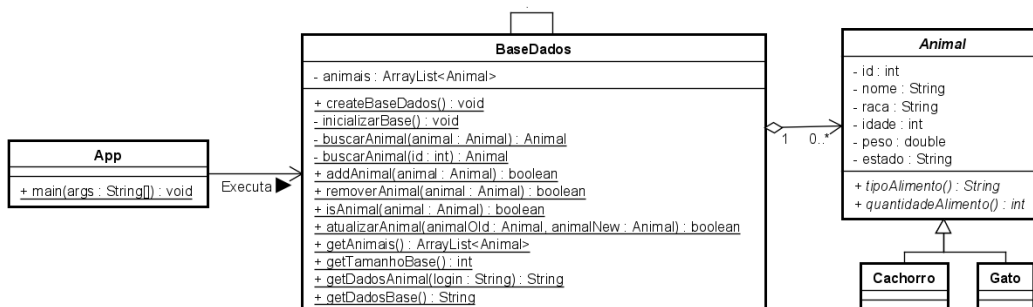
b) Proponha uma solução para receitar a quantidade de comida de um animal a partir do estado de “desnutricao” versus peso, a partir das informações de alimentação, das quais são dadas levando em consideração o APÊNDICE B (quantidade de comida versus peso do animal).

c) Deve-se ter em App:

- O cachorro REX, doberman, 1 ano de idade, 10kg e desnutrido;
- O gato PIXANO, angorá, 1 ano de idade, 2kg e desnutrido;
- Outro gato: PIXANO, angorá, 1 ano de idade, 3kg e saudável;
- Exiba a receita do animal REX.

| |  |  |
|----|--|--|
| | kg | g |
| 1 | 75 | |
| 2 | 130 | |
| 3 | 175 | |
| 4 | 215 | |
| 5 | 255 | |
| 6 | 295 | |
| 8 | 365 | |
| 10 | 430 | |
| 12 | 495 | |
| 14 | 555 | |
| 16 | 615 | |
| 18 | 670 | |
| 20 | 725 | |
| 25 | 855 | |

APÊNDICE A



APÊNDICE B

```
public class MPOO {  
    void saudacao(){  
        System.out.println("Boa Prova!");  
    }  
  
    public static void main(String[] args) {  
        new MPOO().saudacao();  
    }  
}
```
