



UAST

Unidade Acadêmica
de Serra Talhada - PE

Reide 2006



MPOO

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

Disciplina: Modelagem e Programação Orientada a Objetos (MPOO)

Profº: Richarlyson D'Emery

Data: 11 / 04 / 2022

Nota: _____

Aluno: _____

1ª VERIFICAÇÃO DE APRENDIZAGEM (PRÁTICA) – PARTE 2

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) Classes, atributos, métodos construtor e acesso, encapsulamento: 3,0 pontos; (ii) Herança: 1,0 ponto; (iii) Polimorfismo de Objetos: 0,5 ponto; (iv) interface: 1,0 ponto; (v) Composição: 1,0 ponto; (vi) Cardinalidade por ArrayList na agregação: 0,5 ponto, (vii) Cardinalidade por ArrayList na composição: 1,5 pontos, (viii) Dependências: 0,5 ponto; (ix) instâncias e chamadas de métodos: 1,0 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. (Também poderá ser utilizada a codificação da 1ª VA Parte 1)
- 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:

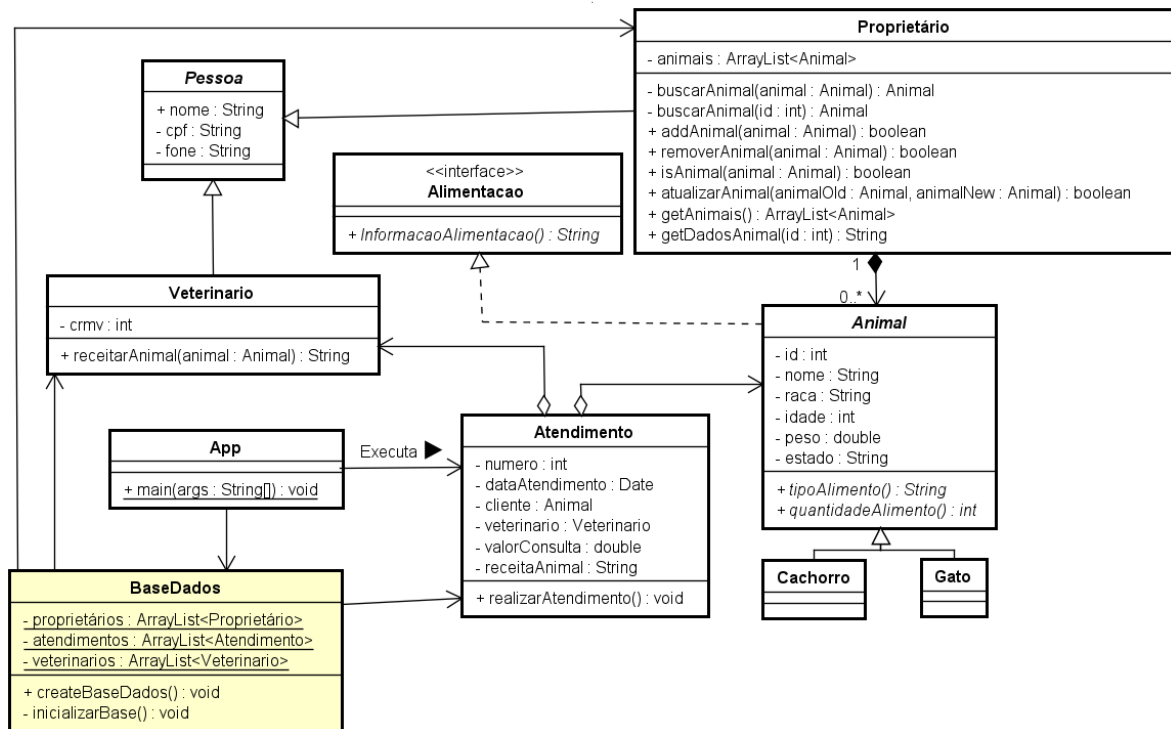
- Observe os encapsulamentos dos atributos.
- Observe os relacionamentos entre as classes.
- Um animal do sistema só existe se pertencer a um proprietário.
- Um proprietário poderá ter diversos animais
- Proprietários, Veterinários e Atendimentos estão armazenados em uma de BaseDados. Faça o devido uso de ArrayList e seus métodos.
- Deve-se considerar:
 - 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.
 - O atendimento de um animal por um veterinário só será realizado se o animal possuir o estado de “desnutricao”, então este receitará o animal a partir das informações de alimentação, das quais são dadas levando em consideração o APÊNDICE B (quantidade de comida versus peso do animal).
 - O método **public void informacaoAlimentacao()** deve informar a alimentação a partir do tipo e da quantidade do alimento para um tipo de um animal. Exemplo:

```
Console
App [Java Application] C:\Program Files\Java\
Tipo Alimentação: ração
Quantidade Alimentação: 75
```

b) Deve-se ter em App:

- um veterinário de dados: José, CPF: 111.111.111-11, fone: (87)99999-9999 e crmv: 1234
- 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;
- Todos os animais pertencem a Maria, CPF: 222.222.222-22, fone: (87)88888-8888.
- Dois atendimentos, sendo um para cada animal que necessita de atendimento. A data de atendimento para ambos é de 11/04/2022 a ser atendido pelo veterinário José e valor de Consulta é de R\$ 100 para ambos
- Exiba a receita do animal REX.

APÊNDICE A



APÊNDICE B

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

```

public class MPOO {
    void saudacao(){
        System.out.println("Boa Prova!");
    }

    public static void main(String[] args) {
        new MPOO().saudacao();
    }
}

```