

Disciplina: Java

Profª Kátia Livia Zambon

## **Lista 2 de exercícios – Terceiro Bimestre (2017)**

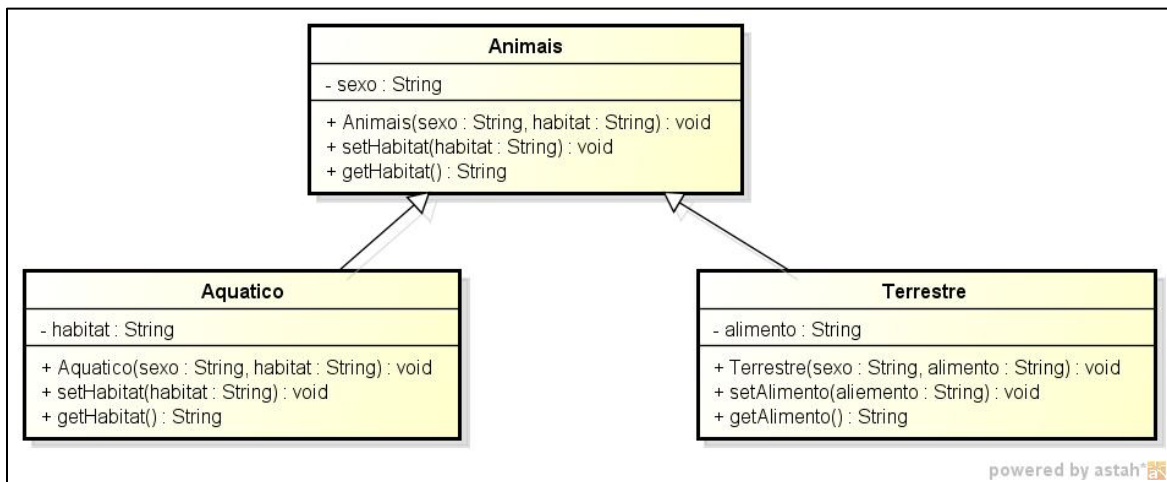
Conteúdo: Herança

1. Desenhe o diagrama de classe para se orientar e crie as seguintes classes:

- Uma **classe Pessoa** com atributos nome e sobrenome. Cada um desses atributos deve ter métodos para acessá-los e alterá-los. A classe Pessoa ainda deve ter um método `getNomeCompleto` que não possui parâmetro de entrada e que retorna a concatenação do atributo nome com o atributo sobrenome separados por um espaço. A classe deve também possuir um construtor sem parâmetros e um outro construtor que recebe como parâmetros o nome e o sobrenome da pessoa e altera esses atributos;
- Crie uma **subclasse** de Pessoa chamada **Funcionário**, que possui os atributos matrícula e salário com seus métodos de acesso e alteração (getters e setters). Todo funcionário recebe seu salário em duas parcelas, 60% na primeira parcela e 40% na segunda parcela. Crie dois métodos `getSalarioPrimeiraParcela` e `getSalarioSegundaParcela` que retorna o valor correto de cada parcela;
- Crie uma **subclasse** de Funcionário chamada **Professor**, mas que recebe seu salário em uma única parcela, diferentemente dos demais funcionários. Crie os métodos que achar necessário para professor;
- Por fim, crie uma classe que instancia três objetos, uma Pessoa, um Funcionário e um Professor. Insira o nome e sobrenome dos objetos exibindo-os no console, além de preencher seus salários (de quem tiver) exibindo as respectivas parcelas.

Fonte: CECIERJ, curso de Java, 2011.

2. Observe o diagrama de classe e os códigos do projeto. Dados os códigos abaixo, desenhe o diagrama de classe correspondente e responda as perguntas:



The screenshot shows an IDE with a project named **ProjZoo**. The project structure on the left includes **Pacotes de Códigos-fonte** (containing **projzoo** with **Animais.java**, **Aquatico.java**, **MainZoo.java**, and **Terrestre.java**) and **Pacotes de Teste** (containing **Bibliotecas** and **Bibliotecas de Testes**). The main editor shows the source code of **Animais.java**:

```

1 package projzoo;
2
3
4 public class Animais {
5     private final String sexo;
6
7     Animais(String sexo)
8     {
9         this.sexo=sexo;
10        System.out.println("Animal super criado");
11    }
12
13    public String getSexo()
14    {
15        return this.sexo;
16    }
17 }

```

**final:** tem que ser inicializada na classe pelo construtor.

Quando é aplicado na classe, não permite estende-la, nos métodos impede que o mesmo seja sobrescrito (override@) na subclasse, e nos valores de variáveis não pode ser alterado depois que já tenha sido atribuído um valor.

## Classe Terrestre

```

1  package projzoo;
2
3  public class Terrestre extends Animais{
4      String alimento;
5
6      Terrestre(String sexo, String alimento)
7      {
8          super(sexo);
9          this.alimento=alimento;
10         System.out.println("Animal terrestre criado");
11     }
12
13     public void setAlimento(String alimento)
14     {
15         this.alimento=alimento;
16     }
17
18     public String getAlimento()
19     {
20         return this.alimento;
21     }
22 }

```

## Classe Aquatico

```

1  package projzoo;
2
3  public class Aquatico extends Animais{
4      private String habitat;
5
6      Aquatico(String sexo, String habitat)
7      {
8          super(sexo);
9          this.habitat=habitat;
10         System.out.println("Animal aquatico criado");
11     }
12
13     public void setHabitat(String habitat)
14     {
15         this.habitat=habitat;
16     }
17
18     public String getHabitat()
19     {
20         return this.habitat;
21     }
22 }

```

## Classe MainZoo – contém a classe principal:

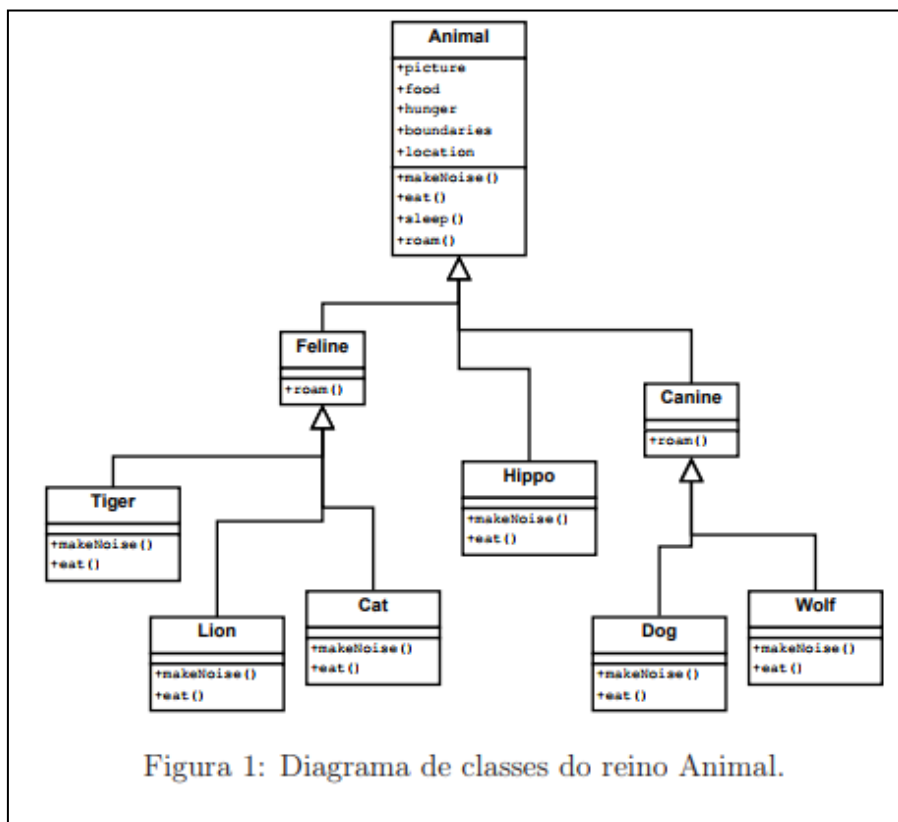
```

1  package projzoo;
2  public class MainZoo {
3
4      public static void main(String[] args) {
5
6          Aquatico turtle = new Aquatico("femea", "oceano");
7          Terrestre giraffe = new Terrestre("macho", "folhas");
8          System.out.println("Alimento predileto das girafas: "+giraffe.getAlimento());
9          System.out.println("Habitat das turtles: "+turtle.getHabitat());
10         System.out.println("Sexo da turtle: "+turtle.getSexo());
11         System.out.println("Sexo da giraffe: "+giraffe.getSexo());
12         giraffe.setAlimento("copa da arvore");
13         System.out.println("Alimento predileto das girafas: "+giraffe.getAlimento());
14     }
15 }

```

- a) Qual a execução do programa MainZoo?
- b) Considerando as instâncias realizadas no exercício anterior na classe principal, qual seria a execução das linhas de comando abaixo?
- ```
System.out.println("Alimento da turtle: "+turtle.getAlimento());
System.out.println("Habitat da giraffe: "+giraffe.getHabitat());
```
- c) A linha de comando abaixo apresenta algum erro? Por quê?
- ```
Aquatico turtle = new Aquatico("oceano");
```
- d) No construtor da classe Terrestre e Aquatico por que utilizamos a palavra "super"?
- e) O que o conceito de herança propõe e o que auxiliou na modelagem do ProjZoo?

3. Considere a imagem abaixo e responda:



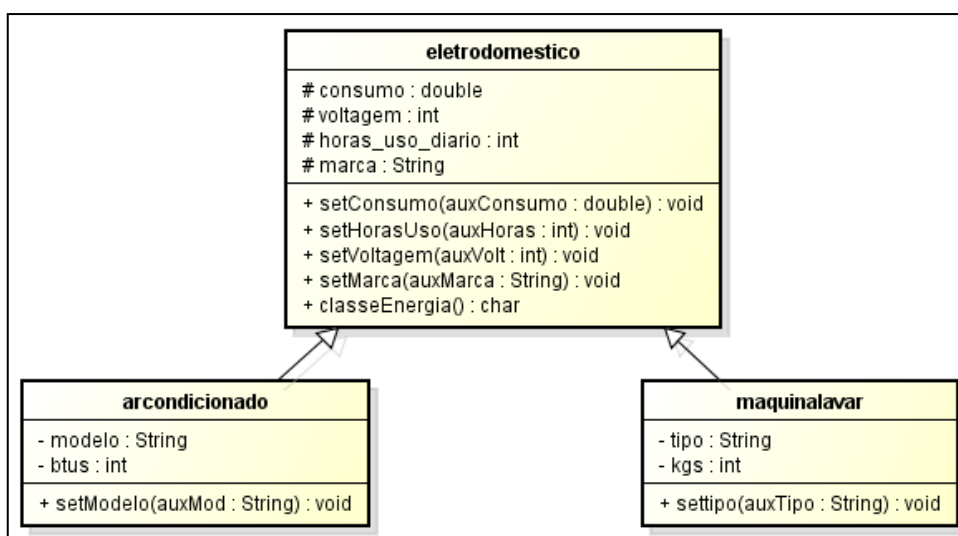
- a) Quais são as superclasses?
- b) Quais são as subclasses?
- c) Os atributos da classe Animal possuem visibilidade *public*. Se modificarmos para *private*, quais classes poderão acessá-los?
- d) Os atributos da classe Animal possuem visibilidade *public*. Se modificarmos para *protected*, quais classes poderão acessá-los?

4. Desenvolva um projeto que contenha as seguintes classes:

- Ingresso() – possui o atributo valor (double) e um método imprimeValor() que retorna o valor do ingresso;
- Normal() – que herda Ingresso() e contém um construtor que apresenta "Ingresso Normal";
- Vip() – que herda Ingresso() e possui o atributo adicional (double). Crie um método que retorne o valor do ingresso Vip (com o adicional incluído);
- Camarote – que herda a classe Vip(), possui o atributo localização (String) e o atributo adicional de camarote (double). Crie métodos que atribuam e apresentem esta localização e o preço do camarote.

Adaptado de: [http://professor.unisinos.br/mraeder/lab1/exercicios\\_her\\_pol.pdf](http://professor.unisinos.br/mraeder/lab1/exercicios_her_pol.pdf)

5. Observe o diagrama de Classe abaixo para resolver a atividade descrita:



Crie um novo projeto. Atenção: você terá 4 classes, uma delas contém a classe principal que você poderá nomear de testa\_eletros, por exemplo.

Modele a classe eletrodomestico com os atributos descritos no Diagrama de classe

Crie um construtor para esta classe com uma mensagem "objeto eletrodoméstico ok!"

Programa os métodos a seguir:

- setConsumo( ) – deve ser maior que zero
- setHorasUso( ) – deve ser maior que zero
- setVoltagem( ) – só pode receber os valores 110V ou 220V
- setMarca( ) – pode receber qualquer marca: LG, Electrolux, Elgin, Midea etc.
- classeEnergia( ) vocês deve pesquisar quais são as faixas de consumo que classificam os eletrodomésticos de A a E. Se possível associar aos aparelhos.

( <http://www.inmetro.gov.br/consumidor/tabelas.asp>)

Modele a classe arcondicionado criando um construtor que receba a quantidade de BTUs que o aparelho tem.

Programa o método setModelo( ) que deve receber se o aparelho é split ou wall (de parede)

Modele a classe maquina lavar que contém um construtor que inicializa a quantidade de roupas que a máquina lava (em kgs)

Programe o método `setTipo()` que recebe a string que caracteriza a máquina como somente lava ou lava e seca.

Instancie objetos para todas as classes em uma nova classe que contenha a principal, por exemplo, `testa_eletros.java`.

Utilize os métodos já feitos e, se necessário, crie outros para apresentar os dados de cada aparelho ou desempenhar outras funções.

Crie uma subclasse de eletrodomestico de sua preferência que herda a classe eletrodomestico. Crie os atributos característicos e implemente os métodos.

Instancie objetos de todas as classes criadas.