PROF.(A):Tiago de Almeida Lopes
ALUNO(A): Vinícius dos Santos Andrade
DATA: 20/08/2024

Lista de Exercícios 04 – Orientação a Objetos Avançada em Java

1. Criar uma interface AnimalIF com os métodos comer, moverse, dormir;

```java
public interface AnimalIF {

   void comer(float quantidade);
   void moverse(float distancia);
   void dormir(int horas);
}
```

2. Criar uma classe abstrata AnimalAB que implementa a interface AnimalIF e define os métodos abstratos assinado na interface;

```
public abstract class AnimalTerresteAB implements AnimalIF {

   protected AnimalTerreste animalTerreste;
   protected HabitatTerrestre habitatTerrestre;
   protected String nome;
   protected int idade;
   protected int qtdMembros;
   protected float velocidadeMaxima;
   protected float comidaIngerida;
   protected float distanciaPercorrida;
   protected int horasDormidas;

   public AnimalTerresteAB() {
   }

public AnimalTerresteAB(HabitatTerrestre habitatTerrestre,
                        String nome,
                        int idade,
                        int qtdMembros,
                        float velocidadeMaxima,
                        float comidaIngerida,
                        float distanciaPercorrida,
                        int horasDormidas) {
       this.habitatTerrestre = habitatTerrestre;
       this.nome = nome;
       this.idade = idade;
       this.qtdMembros = qtdMembros;
       this.velocidadeMaxima = velocidadeMaxima;
       this.comidaIngerida = comidaIngerida;
       this.distanciaPercorrida = distanciaPercorrida;
       this.horasDormidas = horasDormidas;
   }

public AnimalTerresteAB(HabitatTerrestre habitatTerrestre,
                        float comidaIngerida,
                        float distanciaPercorrida,
                          int horasDormidas) {
       this.habitatTerrestre = habitatTerrestre;
       this.comidaIngerida = comidaIngerida;
       this.distanciaPercorrida = distanciaPercorrida;
       this.horasDormidas = horasDormidas;
   }



   @Override
   public void comer(float quantidade) {
       this.comidaIngerida += quantidade;
   }
```

```java
@Override
public void moverse(float distancia) {
    this.distanciaPercorrida += distancia;
}

@Override
public void dormir(int horas) {
    this.horasDormidas += horas;
}

public int getIdade() {
    return idade;
}

public AnimalTerreste getAnimalTerreste() {
    return animalTerreste;
}

public HabitatTerrestre getHabitatTerrestre() {
    return habitatTerrestre;
}

public String getNome() {
    return nome;
}

public int getQtdMembros() {
    return qtdMembros;
}

public float getComidaIngerida() {
    return comidaIngerida;
}

public float getDistanciaPercorrida() {
    return distanciaPercorrida;
}

public int getHorasDormidas() {
    return horasDormidas;
}

public float getVelocidadeMaxima() {
    return velocidadeMaxima;
}

public void setVelocidadeMaxima(float velocidadeMaxima) {
    this.velocidadeMaxima = velocidadeMaxima;
}

public void setNome(String nome) {
    this.nome = nome;
}
```

```java
    public void setIdade(int idade) {
        this.idade = idade;
    }

    public void setQtdMembros(int qtdMembros) {
       this.qtdMembros = qtdMembros;
    }

@Override
public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null) return false;
        if (!(o instanceof AnimalTerresteAB that)) return false;

        return Objects.equals(this.animalTerreste, that.animalTerreste) &&
                Objects.equals(this.habitatTerrestre, that.habitatTerrestre) &&
                String.valueOf(this.nome).equals(that.nome) &&
                this.idade == that.idade &&
                this.qtdMembros == that.qtdMembros &&
                Float.compare(that.velocidadeMaxima, this.velocidadeMaxima) == 0 &&
                Float.compare(that.comidaIngerida, this.comidaIngerida) == 0 &&
               Float.compare(that.distanciaPercorrida, this.distanciaPercorrida) == 0 &&
                this.horasDormidas == that.horasDormidas;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int hash = 1;

    hash *= prime + ((nome == null) ? 0 : nome.hashCode());
    hash *= prime + ((animalTerreste == null) ? 0 : animalTerreste.hashCode());
    hash *= prime + ((habitatTerrestre == null) ? 0 : habitatTerrestre.hashCode());
    hash *= prime + ((idade == 0) ? 0 : idade);
    hash *= prime + ((qtdMembros == 0) ? 0 : qtdMembros);
    hash *= prime + ((velocidadeMaxima == 0) ? 0 : Float.floatToIntBits(velocidadeMaxima));
    hash *= prime + ((comidaIngerida == 0) ? 0 : Float.floatToIntBits(comidaIngerida));
    hash *= prime + ((distanciaPercorrida == 0) ? 0 : Float.floatToIntBits(distanciaPercorrida));
    hash *= prime + ((horasDormidas == 0) ? 0 : horasDormidas);

        if (hash < 0) hash = -hash;

        return hash;
}
```

```java
@Override
public String toString() {
       return "{" +
                "\"animalTerreste\":\"" + animalTerreste + "\"," +
                "\"habitatTerrestre\":\"" + habitatTerrestre + "\"," +
                "\"nome\":\"" + nome + "\"," +
                "\"idade\":" + idade + "," +
                "\"qtdMembros\":" + qtdMembros + "," +
                "\"comidaIngerida\":\"" + comidaIngerida + "\"," +
                "\"distanciaPercorrida\":" + distanciaPercorrida + "," +
                "\"horasDormidas\":" + horasDormidas +
                "}";
   }
}
```

3. Crie as classes concretas Cachorro, Gato, Elefante, Leão que herdam da classe AnimalAB e sobrescreva os métodos abstratos comer, moverse, dormir; As ações desses métodos consistem em alterar o estado interno do objeto através dos atributos de instância que representam a quantidade de comida ingerida ao comer, a quantidade de caminho percorrido ao moverse, e a quantidade de horas ao dormir;

```
public class Cachorro extends AnimalTerresteAB {

    public Cachorro(HabitatTerrestre habitatTerrestre,
                    float comidaIngerida,
                    float distanciaPercorrida,
                    int horasDormidas) {

        super(habitatTerrestre,
                comidaIngerida,
                distanciaPercorrida,
                horasDormidas);

        this.animalTerreste = CACHORRO;
    }

    public Cachorro(HabitatTerrestre habitatTerrestre,
                    String nome,
                    int idade,
                    int qtdMembros,
                    float velocidadeMaxima,
                    float comidaIngerida,
                    float distanciaPercorrida,
                    int horasDormidas) {

        super(habitatTerrestre,
                nome,
                idade,
                qtdMembros,
                velocidadeMaxima,
                comidaIngerida,
                distanciaPercorrida,
                horasDormidas);

        this.animalTerreste = CACHORRO;
    }

    @Override
    public void comer(float quantidade) {
        super.comer(quantidade);
        System.out.println("Cachorro comendo ... ");
    }
```

```java
 @Override
 public void moverse(float distancia) {
       super.moverse(distancia);
       System.out.println("Cachorro se movendo ... ");
   }

   @Override
   public void dormir(int horas) {
       super.dormir(horas);
       System.out.println("Cachorro dormindo ... ");
   }

   @Override
   public String toString() {
       return "{\n" +
               "  \"nome\": \"" + getNome() + "\",\n" +
               "  \"animalTerreste\": \"" + animalTerreste + "\",\n" +
               "  \"habitatTerrestre\": \"" + getHabitatTerrestre() + "\",\n" +
               "  \"idade\": " + getIdade() + ",\n" +
               "  \"qtdMembros\": " + getQtdMembros() + ",\n" +
               "  \"velocidadeMaxima\": " + getVelocidadeMaxima() + ",\n" +
               "  \"comidaIngerida\": " + getComidaIngerida() + ",\n" +
               "  \"distanciaPercorrida\": " + getDistanciaPercorrida() + ",\n" +
               "  \"horasDormidas\": " + getHorasDormidas() + "\n" +
               "}";
   }
}
```

```java
package animais_terrestres;

import classes_abstratas.AnimalTerresteAB;
import enums.habitats.HabitatTerrestre;

import static enums.tipoAnimal.AnimalTerreste.ELEFANTE;

public class Elefante extends AnimalTerresteAB {

   public Elefante(HabitatTerrestre habitatTerrestre,
                   String nome,
                   int idade,
                   int qtdMembros,
                   float velocidadeMaxima,
                   float comidaIngerida,
                   float distanciaPercorrida,
                   int horasDormidas) {

       super(habitatTerrestre,
               nome,
               idade,
               qtdMembros,
               velocidadeMaxima,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = ELEFANTE;
   }

   public Elefante(HabitatTerrestre habitatTerrestre,
                   float comidaIngerida,
                   float distanciaPercorrida,
                   int horasDormidas) {

       super(habitatTerrestre,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = ELEFANTE;
   }

   @Override
   public void comer(float quantidade) {
       super.comer(quantidade);
       System.out.println("Elefante comendo ... ");
   }

@Override
public void moverse(float distancia) {
       super.moverse(distancia);
       System.out.println("Elefante se movendo ... ");
}
```

```java
    @Override
    public void dormir(int horas) {
        super.dormir(horas);
        System.out.println("Elefante dormindo...");
    }

    @Override
    public String toString() {
        return "{\n" +
                "  \"nome\": \"" + getNome() + "\",\n" +
                "  \"animalTerreste\": \"" + animalTerreste + "\",\n" +
                "  \"habitatTerrestre\": \"" + getHabitatTerrestre() + "\",\n" +
                "  \"idade\": " + getIdade() + ",\n" +
                "  \"qtdMembros\": " + getQtdMembros() + ",\n" +
                "  \"velocidadeMaxima\": " + getVelocidadeMaxima() + ",\n" +
                "  \"comidaIngerida\": " + getComidaIngerida() + ",\n" +
               "  \"distanciaPercorrida\": " + getDistanciaPercorrida() + ",\n" +
                "  \"horasDormidas\": " + getHorasDormidas() + "\n" +
                "}";
    }
}
```

```java
package animais_terrestres;

import classes_abstratas.AnimalTerresteAB;
import enums.habitats.HabitatTerrestre;

import static enums.tipoAnimal.AnimalTerreste.GATO;

public class Gato extends AnimalTerresteAB {

   public Gato(HabitatTerrestre habitatTerrestre,
               String nome,
               int idade,
               int qtdMembros,
               float velocidadeMaxima,
               float comidaIngerida,
               float distanciaPercorrida,
               int horasDormidas) {

       super(habitatTerrestre,
               nome,
               idade,
               qtdMembros,
               velocidadeMaxima,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = GATO;
   }

   public Gato(HabitatTerrestre habitatTerrestre,
               float comidaIngerida,
               float distanciaPercorrida,
               int horasDormidas) {

       super(habitatTerrestre,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = GATO;
   }

   @Override
   public void comer(float quantidade) {
       super.comer(quantidade);
       System.out.println("Gato comendo ... ");
   }

   @Override
   public void moverse(float distancia) {
       super.moverse(distancia);
       System.out.println("Gato se movendo ... ");
   }
```

```java
    @Override
    public void dormir(int horas) {
        super.dormir(horas);
        System.out.println("Gato dormindo ... ");
    }

    @Override
    public String toString() {
        return "{\n" +
                "  \"nome\": \"" + getNome() + "\",\n" +
                "  \"animalTerreste\": \"" + animalTerreste + "\",\n" +
                "  \"habitatTerrestre\": \"" + getHabitatTerrestre() + "\",\n" +
                "  \"idade\": " + getIdade() + ",\n" +
                "  \"qtdMembros\": " + getQtdMembros() + ",\n" +
                "  \"velocidadeMaxima\": " + getVelocidadeMaxima() + ",\n" +
                "  \"comidaIngerida\": " + getComidaIngerida() + ",\n" +
                "  \"distanciaPercorrida\": " + getDistanciaPercorrida() + ",\n" +
                "  \"horasDormidas\": " + getHorasDormidas() + "\n" +
                "}";
    }
}
```

```java
package animais_terrestres;

import classes_abstratas.AnimalTerresteAB;
import enums.habitats.HabitatTerrestre;

import static enums.tipoAnimal.AnimalTerreste.LEAO;

public class Leao extends AnimalTerresteAB {

   public Leao(HabitatTerrestre habitatTerrestre,
               String nome,
               int idade,
               int qtdMembros,
               float velocidadeMaxima,
               float comidaIngerida,
               float distanciaPercorrida,
               int horasDormidas) {

       super(habitatTerrestre,
               nome,
               idade,
               qtdMembros,
               velocidadeMaxima,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = LEAO;
   }

   public Leao(HabitatTerrestre habitatTerrestre,
               float comidaIngerida,
               float distanciaPercorrida,
               int horasDormidas) {

       super(habitatTerrestre,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       this.animalTerreste = LEAO;
   }

   @Override
   public void comer(float quantidade) {
       super.comer(quantidade);
       System.out.println("Leão comendo ... ");
   }
```

```java
    @Override
    public void moverse(float distancia) {
        super.moverse(distancia);
        System.out.println("Leão se movendo ... ");
    }

    @Override
    public void dormir(int horas) {
        super.dormir(horas);
        System.out.println("Leão dormindo ... ");
    }

    @Override
    public String toString() {
        return "{\n" +
                "  \"nome\": \"" + getNome() + "\",\n" +
                "  \"animalTerreste\": \"" + animalTerreste + "\",\n" +
                "  \"habitatTerrestre\": \"" + getHabitatTerrestre() + "\",\n" +
                "  \"idade\": " + getIdade() + ",\n" +
                "  \"qtdMembros\": " + getQtdMembros() + ",\n" +
                "  \"velocidadeMaxima\": " + getVelocidadeMaxima() + ",\n" +
                "  \"comidaIngerida\": " + getComidaIngerida() + ",\n" +
                "  \"distanciaPercorrida\": " + getDistanciaPercorrida() + ",\n" +
                "  \"horasDormidas\": " + getHorasDormidas() + "\n" +
                "}";
    }
}
```

4. Crie uma classe Peixe e Pombo. De quem vamos herdar? Um peixe nada e um pombo voa então os métodos nadar e voar devem estar na classe abstrata Animal? Não. Então criem uma classe abstrata AnimalMarinhoAB ,AnimalVoadorAB, AnimalTerrestreAB que implementa a classe abstrata AnimalAB para representar a classe abstrata para animais marinhos e aéreos “que voam”.

```java
package classes_abstratas;

import enums.habitats.HabitatAquatico;
import enums.tipoAnimal.AnimalAquatico;

import java.util.Objects;

public abstract class AnimalMarinhoAB implements AnimalIF {

   protected AnimalAquatico animalAquatico;
   protected HabitatAquatico habitatAquatico;
   protected String nome;
   protected int idade;
   protected float profundidadeMaxima;
   protected float comidaIngerida;
   protected float distanciaPercorrida;
   protected int horasDormidas;

   public AnimalMarinhoAB() {
   }

   public AnimalMarinhoAB(HabitatAquatico habitatAquatico,
                          String nome,
                          int idade,
                          float profundidadeMaxima,
                          float comidaIngerida,
                          float distanciaPercorrida,
                          int horasDormidas) {
       this.habitatAquatico = habitatAquatico;
       this.nome = nome;
       this.idade = idade;
       this.profundidadeMaxima = profundidadeMaxima;
       this.comidaIngerida = comidaIngerida;
       this.distanciaPercorrida = distanciaPercorrida;
       this.horasDormidas = horasDormidas;
   }
```

```java
public AnimalMarinhoAB(HabitatAquatico habitatAquatico,
                       float comidaIngerida,
                       float distanciaPercorrida,
                       int horasDormidas) {
    this.habitatAquatico = habitatAquatico;
    this.comidaIngerida = comidaIngerida;
    this.distanciaPercorrida = distanciaPercorrida;
    this.horasDormidas = horasDormidas;
}

@Override
public void comer(float quantidade) {
    this.comidaIngerida += quantidade;
}

@Override
public void moverse(float distancia) {
    this.distanciaPercorrida += distancia;
}

@Override
public void dormir(int horas) {
    this.horasDormidas += horas;
}

public void setIdade(int idade) {
    this.idade = idade;
}

public void setNome(String nome) {
    this.nome = nome;
}

public void setProfundidadeMaxima(int profundidadeMaxima) {
    this.profundidadeMaxima = profundidadeMaxima;
}

public int getIdade() {
    return idade;
}

public AnimalAquatico getAnimalAquatico() {
    return animalAquatico;
}

public HabitatAquatico getHabitatAquatico() {
    return habitatAquatico;
}

public String getNome() {
    return nome;
}
```

```java
public float getComidaIngerida() {
    return comidaIngerida;
}

public float getDistanciaPercorrida() {
    return distanciaPercorrida;
}

public int getHorasDormidas() {
    return horasDormidas;
}

public float getProfundidadeMaxima() {
    return profundidadeMaxima;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (o == null) return false;
    if (!(o instanceof AnimalMarinhoAB that)) return false;

    return Objects.equals(this.animalAquatico, that.animalAquatico) &&
            Objects.equals(this.habitatAquatico, that.habitatAquatico) &&
            String.valueOf(this.nome).equals(that.nome) &&
            this.idade == that.idade &&
            Objects.equals(this.profundidadeMaxima, that.profundidadeMaxima) &&
            Objects.equals(this.comidaIngerida, that.comidaIngerida) &&
            Objects.equals(this.distanciaPercorrida, that.distanciaPercorrida) &&
            this.horasDormidas == that.horasDormidas;
}

@Override
public int hashCode() {
    final int prime = 31;
    int hash = 1;

    hash *= prime + ((nome == null) ? 0 : nome.hashCode());
    hash *= prime + ((animalAquatico == null) ? 0 : animalAquatico.hashCode());
    hash *= prime + ((habitatAquatico == null) ? 0 : habitatAquatico.hashCode());
    hash *= prime + ((idade == 0) ? 0 : idade);
    hash *= prime + ((profundidadeMaxima == 0) ? 0 :Float.floatToIntBits(profundidadeMaxima));
   hash *= prime + ((comidaIngerida == 0) ? 0 : Float.floatToIntBits(comidaIngerida));
 hash *= prime + ((distanciaPercorrida == 0) ? 0 : Float.floatToIntBits(distanciaPercorrida));
    hash *= prime + ((horasDormidas == 0) ? 0 : horasDormidas);

    if (hash < 0) hash = -hash;

    return hash;

}
```

```java
@Override
    public String toString() {
        return "{" +
                "\"animalAquatico\":\"" + animalAquatico + "\"," +
                "\"habitatAquatico\":\"" + habitatAquatico + "\"," +
                "\"nome\":\"" + nome + "\"," +
                "\"idade\":" + idade + "," +
                "\"profundidadeMaxima\":" + profundidadeMaxima + "," +
                "\"comidaIngerida\":" + comidaIngerida + "," +
                "\"distanciaPercorrida\":" + distanciaPercorrida + "," +
                "\"horasDormidas\":" + horasDormidas +
                "}";
    }
}
```

```java
package classes_abstratas;

import enums.habitats.HabitatsAereos;
import enums.tipoAnimal.AnimalAereo;

import java.util.Objects;

public abstract class AnimalVoadorAB implements AnimalIF {

   protected HabitatsAereos habitatAereo;
   protected AnimalAereo animalAereo;
   protected String nome;
   protected int idade;
   protected float velocidadeMaxima;
   protected float envergadura;
   protected float altitudeMaxima;
   protected float comidaIngerida;
   protected float distanciaPercorrida;
   protected int horasDormidas;

   public AnimalVoadorAB() {
   }

   public AnimalVoadorAB(HabitatsAereos habitatAereo,
                         String nome,
                         int idade,
                         float velocidadeMaxima,
                         float envergadura,
                         float altitudeMaxima,
                         float comidaIngerida,
                         float distanciaPercorrida,
                         int horasDormidas) {
       this.habitatAereo = habitatAereo;
       this.nome = nome;
       this.idade = idade;
       this.velocidadeMaxima = velocidadeMaxima;
       this.envergadura = envergadura;
       this.altitudeMaxima = altitudeMaxima;
       this.comidaIngerida = comidaIngerida;
       this.distanciaPercorrida = distanciaPercorrida;
       this.horasDormidas = horasDormidas;
   }
```

```java
public AnimalVoadorAB(HabitatsAereos habitatAereo,
                      float comidaIngerida,
                      float distanciaPercorrida,
                      int horasDormidas) {
    this.habitatAereo = habitatAereo;
    this.comidaIngerida = comidaIngerida;
    this.distanciaPercorrida = distanciaPercorrida;
    this.horasDormidas = horasDormidas;
}

@Override
public void comer(float quantidade) {
    this.comidaIngerida += quantidade;
}

@Override
public void moverse(float distancia)
    this.distanciaPercorrida += distancia;
}

@Override
public void dormir(int horas) {
    this.horasDormidas += horas;
}

public void setIdade(int idade) {
    this.idade = idade;
}

public void setVelocidadeMaxima(int velocidadeMaxima) {
    this.velocidadeMaxima = velocidadeMaxima;
}

public void setNome(String nome) {
    this.nome = nome;
}

public void setEnvergadura(int envergadura) {
    this.envergadura = envergadura;
}

public void setAltitudeMaxima(int altitudeMaxima) {
    this.altitudeMaxima = altitudeMaxima;
}

public float getVelocidadeMaxima() {
    return velocidadeMaxima;
}
```

```java
public HabitatsAereos getHabitatAereo() {
    return habitatAereo;
}

public AnimalAereo getAnimalAereo() {
    return animalAereo;
}

public int getIdade() {
    return idade;
}

public String getNome() {
    return nome;
}

public float getEnvergadura() {
    return envergadura;
}

public float getComidaIngerida() {
    return comidaIngerida;
}

public float getAltitudeMaxima() {
    return altitudeMaxima;
}

public float getDistanciaPercorrida() {
    return distanciaPercorrida;
}

public int getHorasDormidas() {
    return horasDormidas;
}
```

```java
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null) return false;
        if (!(o instanceof AnimalVoadorAB that)) return false;

        return Objects.equals(this.animalAereo, that.animalAereo) &&
                Objects.equals(this.habitatAereo, that.habitatAereo) &&
                String.valueOf(this.nome).equals(that.nome) &&
                this.idade == that.idade &&
                Float.compare(this.velocidadeMaxima, that.velocidadeMaxima) == 0 &&
                Float.compare(this.envergadura, that.envergadura) == 0 &&
                Float.compare(this.altitudeMaxima, that.altitudeMaxima) == 0 &&
                Float.compare(this.comidaIngerida, that.comidaIngerida) == 0 &&
               Float.compare(this.distanciaPercorrida, that.distanciaPercorrida) == 0 &&
                this.horasDormidas == that.horasDormidas;
    }

    @Override
    public int hashCode() {
        final int prime = 31;
        int hash = 1;

        hash *= prime + ((nome == null) ? 0 : nome.hashCode());
        hash *= prime + ((animalAereo == null) ? 0 : animalAereo.hashCode());
        hash *= prime + ((habitatAereo == null) ? 0 : habitatAereo.hashCode());
        hash *= prime + ((idade == 0) ? 0 : idade);
hash *= prime + ((velocidadeMaxima == 0) ? 0 : Float.floatToIntBits(velocidadeMaxima));
       hash *= prime + ((envergadura == 0) ? 0 : Float.floatToIntBits(envergadura));
       hash *= prime + ((altitudeMaxima == 0) ? 0 : Float.floatToIntBits(altitudeMaxima));
       hash *= prime + ((comidaIngerida == 0) ? 0 : Float.floatToIntBits(comidaIngerida));
hash *= prime + ((distanciaPercorrida == 0) ? 0 : Float.floatToIntBits(distanciaPercorrida));
        hash *= prime + ((horasDormidas == 0) ? 0 : horasDormidas);

        if (hash < 0) hash = -hash;

        return hash;
    }
```

```java
@Override
public String toString() {
	return "{\n" +
		"  \"nome\": \"" + nome + "\",\n" +
		"  \"animalAereo\": \"" + animalAereo + "\",\n" +
		"  \"habitatAereo\": \"" + habitatAereo + "\",\n" +
		"  \"idade\": " + idade + ",\n" +
		"  \"velocidadeMaxima\": " + velocidadeMaxima + ",\n" +
		"  \"envergadura\": " + envergadura + ",\n" +
		"  \"altitudeMaxima\": " + altitudeMaxima + ",\n" +
		"  \"comidaIngerida\": " + comidaIngerida + ",\n" +
		"  \"distanciaPercorrida\": " + distanciaPercorrida + ",\n" +
		"  \"horasDormidas\": " + horasDormidas + "\n" +
		"}";
    }
}
```

```java
package animais_aquaticos;

import classes_abstratas.AnimalMarinhoAB;
import enums.habitats.HabitatAquatico;

import static enums.tipoAnimal.AnimalAquatico.PEIXE_PALHACO;

public class Peixe extends AnimalMarinhoAB {

   public Peixe(HabitatAquatico habitatAquatico,
                String nome,
                int idade,
                float profundidadeMaxima,
                float comidaIngerida,
                float distanciaPercorrida,
                int horasDormidas) {

       super(habitatAquatico,
               nome,
               idade,
               profundidadeMaxima,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       animalAquatico = PEIXE_PALHACO;
   }

   public Peixe(HabitatAquatico habitatAquatico,
                float comidaIngerida,
                float distanciaPercorrida,
                int horasDormidas) {

       super(habitatAquatico,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       animalAquatico = PEIXE_PALHACO;
   }

   @Override
   public void comer(float quantidade) {
       super.comer(quantidade);
       System.out.println("Peixe-palhaço comendo ... ");
   }

   @Override
   public void moverse(float distancia) {
       super.moverse(distancia);
       System.out.println("Peixe-palhaço se movendo ... ");
   }
```

```java
    @Overrid
    public void dormir(int horas) {
        super.dormir(horas);
        System.out.println("Peixe-palhaço dormindo...");
    }

    @Override
    public String toString() {
        return "{\n" +
                "  \"animalAquatico\": \"" + animalAquatico + "\",\n" +
                "  \"habitatAquatico\": \"" + habitatAquatico + "\",\n" +
                "  \"nome\": \"" + nome + "\",\n" +
                "  \"idade\": " + idade + ",\n" +
                "  \"profundidadeMaxima\": " + profundidadeMaxima + ",\n" +
                "  \"comidaIngerida\": " + comidaIngerida + ",\n" +
                "  \"distanciaPercorrida\": " + distanciaPercorrida + ",\n" +
                "  \"horasDormidas\": " + horasDormidas + "\n" +
                "}";
    }
}
```

```java
package animais_aereos;

import classes_abstratas.AnimalVoadorAB;
import enums.habitats.HabitatsAereos;

import static enums.tipoAnimal.AnimalAereo.ARARA_AZUL;

public class Pombo extends AnimalVoadorAB {

   public Pombo(HabitatsAereos habitatsAereos,
                String nome,
                int idade,
                float velocidadeMaxima,
                float envergadura,
                float altitudeMaxima,
                float comidaIngerida,
                float distanciaPercorrida,
                int horasDormidas) {

       super(habitatsAereos,
               nome,
               idade,
               velocidadeMaxima,
               envergadura,
               altitudeMaxima,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       animalAereo = ARARA_AZUL;
   }

   public Pombo(HabitatsAereos habitatsAereos,
                int comidaIngerida,
                float distanciaPercorrida,
                int horasDormidas) {

       super(habitatsAereos,
               comidaIngerida,
               distanciaPercorrida,
               horasDormidas);

       animalAereo = ARARA_AZUL;
   }

   @Override
   public void comer(float quantidade) {
       super.comer(quantidade);
       System.out.println("Arara Azul comendo ... ");
   }
```

```java
    @Override
    public void moverse(float distancia) {
        super.moverse(distancia);
        System.out.println("Arara Azul voando ... ");
    }

    @Override
    public void dormir(int horas) {
        super.dormir(horas);
        System.out.println("Arara Azul dormindo ... ");
    }

    @Override
    public String toString() {
        return "{\n" +
                "  \"animalAereo\": \"" + animalAereo + "\",\n" +
                "  \"habitatsAereos\": \"" + habitatAereo + "\",\n" +
                "  \"nome\": \"" + nome + "\",\n" +
                "  \"idade\": " + idade + ",\n" +
                "  \"velocidadeMaxima\": " + velocidadeMaxima + ",\n" +
                "  \"envergadura\": " + envergadura + ",\n" +
                "  \"altitudeMaxima\": " + altitudeMaxima + ",\n" +
                "  \"comidaIngerida\": " + comidaIngerida + ",\n" +
                "  \"distanciaPercorrida\": " + distanciaPercorrida + ",\n" +
                "  \"horasDormidas\": " + horasDormidas + "\n" +
                "}";
    }
}
```

5. Para todas as classes derivem os atributos de cada objeto a partir da representação real de cada animal, ou seja, seus atributos serão nome, tipo animal, idade, habitat, quantidade de patas, quantidadeAssas, envergaduraAssa, altura, peso. Descubram em quais classes abstratas criadas acima esses atributos se encaixam e os criem.