

Trabalho Final de Fundamentos

Turma 128



Sua dupla foi contratadas para desenvolver um sistema que ajude no gerenciamento de um parque temático, no qual dinossauros vivos serão exibidos, o **Jurassic Zoo**.

A classe Dinossauro foi criada da seguinte forma:

```
public class Dinossauro {
    private int idDinossauro; //corresponde ao ID do dinossauro. Cada dinossauro possui um ID único.
    private String nomeRaca; //corresponde ao nome da raça do dinossauro
    private int tipo; //corresponde ao tipo do dinossauro. 1 para carnívoros, 2 para herbívoros.
    private int categoria; //corresponde a categoria do dinossauro. 1 para Pequeno Porte, 2 para Médio
    Porte, 3 para grande porte.
    private double peso; //corresponde ao peso do dinossauro

    public Dinossauro(int idDinossauro, String nomeRaca, int tipo, int categoria,
        double peso) {
        this.idDinossauro = idDinossauro;
        this.nomeRaca = nomeRaca;
        this.tipo = tipo;
        this.categoria = categoria;
        this.peso = peso;
    }
    public int getIdDinossauro() {
        return idDinossauro;
    }
    public void setIdDinossauro(int idDinossauro) {
        this.idDinossauro = idDinossauro;
    }
    public String getNomeRaca() {
        return nomeRaca;
    }
    public void setNomeRaca(String nomeRaca) {
        this.nomeRaca = nomeRaca;
    }
    public int getCategoria() {
        return categoria;
    }
    public void setCategoria(int categoria) {
        this.categoria = categoria;
    }
    public double getPeso() {
```

```

        return peso;
    }
    public int getTipo() {
        return tipo;
    }
    public void setTipo(int tipo) {
        this.tipo = tipo;
    }
    public void setPeso(double peso) {
        this.peso = peso;
    }
}

```

E a classe CadastroDinossauro conforme a seguir:

```

public class CadastroDinossauro {

    Dinossauro [] cadastroDinossauros;
    private int proximaPosicao;

    public CadastroDinossauro(){
        cadastroDinossauros = new Dinossauro[50];
        proximaPosicao = -1; //evitar "out of range" nos metodos
    }

    public boolean adicionarDinossauro(Dinossauro dino) {
        if(proximaPosicao + 1 < cadastroDinossauros.length){ //pq comeca com -1
            cadastroDinossauros[proximaPosicao] = dino;
            proximaPosicao++;
            return true;
        } else { return false; }
    }

    public Dinossauro pesquisarDinossauro(int id) {
        for(int i=0; i<proximaPosicao; i++){
            if(cadastroDinossauros[i].getIdDinossauro() == id){
                return cadastroDinossauros[i];
            }
        }
        return null;
    }

    public boolean removerDinossauro(int id) {
        for(int i=0; i<proximaPosicao; i++){
            if(cadastroDinossauros[i].getIdDinossauro() == id){
                cadastroDinossauros[i] = null;
                for(int j=i; j<proximaPosicao; j++){
                    cadastroDinossauros[j] = cadastroDinossauros[j+1];
                }
                cadastroDinossauros[proximaPosicao]=null;
            }
        }
    }
}

```

```

        proximaPosicao--;
        return true;
    }
}
return false;
}
}

```

O problema é que a classe CadastroDinossauo ainda está incompleta. São necessários alguns relatórios, e é para isso que você foi contratado.

Relatórios encomendados para você (obrigatório serem métodos, cada um deles e serem chamados na aplicação:

1. Relatório *Quantidade de animais de cada tipo e categoria* (1.0)

O método deve retornar uma String na qual devem aparecer as informações referentes a quantidade de animais para cada tipo e categoria. Exemplo:

Carnívoros: PP: 52, MP: 20, GP: 12. Herbívoros: PP: 68, MP: 38, GP: 30.

2. Relatório *Peso pesado* (1.0)

Escreva um método que receba por parâmetro o tipo e categoria de um dinossauo e retorne o dinossauo (objeto) mais pesado do tipo e a categoria.

3. Relatório *Quantidade de carne para carnívoros* (1.5)

Cada carnívoro come um valor relacionado ao seu peso e de acordo com sua categoria.

Raças de pequeno porte comem cerca de 10% de seu peso por dia.

Raças de médio porte comem cerca de 15% de seu peso por dia.

Raças de grande porte comem cerca de 20% de seu peso por dia.

O método deve retornar a quantidade (em Kg) de carne que deve ser comprado no mês (considere o mês por 30 dias).

4. Relatório *Dá tempo de fugir?* (1.5)

Esse relatório, devido aos últimos acidentes, é considerado de extrema importância.

Considera-se que, para corridas curtas em situação de extremo desespero, uma pessoa consiga correr em média a 20km/h. Sabemos que alguns dinossauros são corredores natos. Assim, as cercas para observação são construídas a determinada distância do dinossauo e bunkers de proteção são instalados próximo as cercas para pessoas se abrigarem em caso de fuga do animal.

Crie um método que peça por parâmetro:

Distância entre o dinossauo e o bunker (em km).

Distância entre a pessoa e o bunker (em km).

Velocidade média do dinossauo (em km/h).

Seu método deverá retornar verdadeiro quando o tempo que a pessoa leva para correr até o bunker é menor que o tempo que o dinossauo leva para correr a mesma distância. Caso contrário, deve retornar falso.

A fórmula para o cálculo do tempo é $tempo = distancia/velocidade$.

5. Relatório *Zonas perigosas do parque* (1.5)

Esse método recebe por parâmetro uma matriz de int. Essa matriz corresponde ao terreno do parque (150 x 150) e em cada posição há o ID de cada um dos animais. O método deverá retornar qual zona do

parque (norte ou sul) possui mais carnívoros, portanto é a zona mais perigosa. Use o método de pesquisa da classe.

6. **Relatório com nome dos dinossauros que possuem mais vogais (1.5)**

Esse método recebe por parâmetro os nomes dos bichos cadastrados e escreve, em ordem alfabética crescente a lista daqueles que tem maior número de vogais

Detalhe para implementação:

- Criar um menu de opções que ative cada um dos métodos solicitados. Este menu seja possível o usuário retornar quantas vezes quiser para optar por um das alternativas sem sair do programa e que tenha a opção de sair quando desejar; **(1.0)**
- Que imprima todos os dinossauros cadastrados **(0.5)**
- Que informe quantas posições do vetor de dinossauros estão disponíveis **(0.5)**

**Prazo para entregar deste trabalho da DUPLA
(pacote zipado de implementação feita no BLUEJ):**

28 de novembro

sala Moodle Intitulada Dinossauros

envio aceito até 14 horas

não serão aceitos trabalhos atrasados

Cuidar para colocar código TODO comentado. Cada método criado e colocar a identificação dos autores no início do código: nome e número de matrícula e turma.

Atenção: *todos códigos ou trechos que forem iguais e não apresentarem o relatório de aprendizagem explicando qual foi a dificuldade da dupla e fontes consultadas (colegas ou sites) e, o que aprenderam com esta interação, serão zerados.*

Cuidado, ao pedir ajuda para colega ou consultar fonte a não observar deste requisito pode prejudicar os colegas que lhe auxiliaram.

Você tem a reponsabilidade de citar e escrever sua dificuldade e o eu aprendeu com ajuda desta fonte.

O trabalho é uma forma de fixar e construir conhecimento.

Como fazer isto?

Direto no código !!!!

Cada aula a partir do dia 31/10 você receber uma dica para lhe auxiliar a fazer o TF.

Importante comparecer na aula e fazer os exercícios de fixação solicitados.

Bom trabalho!