

Atividade Acadêmica de Computação II

Aline A. Silva, Jéssica Cristina O. Ruel

Departamento de Ciência da Computação
Instituto Multidisciplinar – Universidade Federal Rural do Rio de Janeiro (UFRRJ)
Nova Iguaçu – RJ – Brasil

1. Introdução

Este relatório tem como objetivo abordar todos os passos seguidos para a construção da AA de Computação II 2019.1.

2. Descrição

O programa é composto pela interface Animal, que possui os métodos: alimentar, locomover, getSexo, setSexo, getPorte e serPorte, que serão sobrescritos pelas classes/interfaces que a implementarem. Possui as Enums: Alimento, para listar alimentos que os animais comem; Sexo e Porte, para definir um sexo e um porte para cada animal; e Localizacao, para atribuir cada uma à uma Instalação. Temos as interfaces Terrestre, que estende a interface Animal e possui um método default void andar, que printa “estou andando!” quando um animal terrestre anda e sobrescreve o método locomover como default, invocando o método andar dentro dele; a interface Aquatico, que também estende a interface Animal e possui um método default void nadar, que printa “estou nadando!” quando um animal aquático nada e sobrescreve o método locomover como default, invocando o método nadar dentro dele; e a interface Anfíbio, que estende ambas as interfaces anteriores e sobrescreve o método locomover como default, invocando os métodos andar e nadar dentro dele. Temos 10 classes para cada animal: Elefante, Leao, Macaco e Zebra implementam a interface Terrestre; Foca, Pinguim e Tartaruga implementam a interface Anfíbio; e PeixeBoi, Tubarao e Baleia implementam a interface Aquatico. Todas as classes de animais possuem os atributos porte, sexo, alimento para os animais que comem apenas um tipo de alimento e um vetor de alimentos para os animais que comem mais de um tipo de alimento, e sobrescrevem os métodos getSexo, setSexo, getPorte, setPorte e alimentar, este último recebe como parâmetro um alimento e contém um for para percorrer o vetor de alimentos, caso o animal coma mais de um tipo, e um if para comparar o alimento recebido com o tipo que o animal come, retornando true se forem iguais e false caso contrário.

Há também a classe abstrata Instalacao, que possui os atributos capacidadeMaxima, localizacao, temperatura e um vetor de animaisInstalados, seus respectivos get e set, um construtor recebendo como parâmetro todos os atributos declarados e o método instalaAnimal, que recebe um animal. A classe abstrata ElementoInstalacao estende a classe abstrata anterior e possui construtor com os atributos da super. A classe Jaula possui os atributos altura, largura, comprimento, os métodos get, set e area, que faz o cálculo da área da jaula e retorna o resultado. A classe Tanque possui o atributo volumeEmLitros e os métodos

get e set. Ambas as classes estendem a classe abstrata ElementoInstalacao, possuem um construtor com os atributos da super e seus próprios atributos.

A classe FabricaInstalacao é composta por métodos para criar uma instalação para cada tipo de animal. Cada método recebe como parâmetro uma localização e um vetor de animais. Todos os métodos criam uma instalação e setam uma capacidade máxima, a localização, uma temperatura, uma altura, uma largura e um comprimento para as jaulas; e um volume em litros para os tanques. Os métodos apresentam uma mensagem contendo a área da jaula e verificam se os animais foram instalados através do método instalaAnimal. Todos os métodos fazem a contagem do vetor de animais e comparam com a capacidade máxima da instalação. Se o limite é atingido, é exibida uma mensagem para o usuário o informando. E se a quantidade de animais é menor do que a capacidade máxima determinada, é exibida uma mensagem dizendo que ainda é possível adicionar animais àquela instalação, juntamente com a quantidade.

A classe Zoologico possui um arraylist de Localizacao, um método para adicionar todos os locais ao arraylist, um método para criar a instalação de acordo com a localização e um construtor que invoca o método para adicionar locais dentro dele. O método para criar a instalação recebe como parâmetro uma localização e um vetor de animais, e verifica se já existe uma instalação na localização recebida. Se não existe, o método remove esta localização do arraylist e invoca o método para criar uma instalação de acordo com o animal desejado, na localização que foi recebida. Se já existe, é exibida uma mensagem dizendo que a localização está indisponível e para tentar com uma nova localização.

A classe Main contém o método main e é responsável por criar todos os animais, atribuir a eles um porte, um sexo, criar vetores com os animais criados anteriormente, criar um zoológico e atribuir as instalações à este, passando uma localização e o vetor de animais.