INSTITUTO POLITÉCNICO DO CÁVADO E DO AVE
ESCOLA SUPERIOR DE TECNOLOGIA

Relatório

Alexandre Vilaça – 26590

Unidade Curricular

Programação Orientada a Objetos

21 de Dezembro de 2024, Barcelos

2024/2025

Resumo

O projeto proposto para a Unidade Curricular de Programação Orienta a Objetos consiste em desenvolver soluções de software que abordam problemas relacionados com a criação de classes e suas instâncias, onde se deve tratar o encapsulamento, polimorfismo, herança, etc…

Visa este relatório ajudar a entender o propósito e resolução do projeto desenvolvido

Índice

Índice de Figuras

1. Introdução

Enquadramento

Este relatório aborda o progresso e as conclusões de um projeto prático proposto como componente da avaliação individual na Unidade Curricular Programação Orientada a objetos (POO). O objetivo principal deste projeto é aplicar os conceitos e técnicas aprendidas ao longo do semestre para resolver desafios relacionados à programação orientada a objetos. A programação orientada a objetos é um paradigma de desenvolvimento de software que promove o uso de objetos para representar entidades do mundo real. Neste contexto, o objetivo do projeto foi criar um sistema de software que simule a gestão de um zoológico, onde aborda aspetos como a administração de animais, funcionários, espetáculos, habitats e outras funcionalidades importantes para o funcionamento de um ambiente desse tipo.

Enquadramento Teórico e Prático

Requisitos

1. O programa deve permitir o registo e gestão de Animais, Espetáculos, Bilhetes, Habitats, Alimentações, etc..
2. As classes devem ser implementadas com recurso a interfaces, classes abstratas, herança, encapsulamento;
3. Todo o tipo de informação dos objetos do zoológico deve ser guardado em ficheiros txt.
4. É necessário que o programa seja capaz de manipular listas para poderem mais tarde ser adicionadas na “base de dados”.
5. Implementação de logs;
6. Deve ter a implementação de exceptions, principalmente nos métodos onde não se possa ignorar o erro;
7. Aplicação de testes unitários;
8. Criação e uso de DLLs.
9. Implementação de uma interface.

2. Trabalho Desenvolvido

Classes

Na realização do projeto, as classes foram implementadas com o uso de heranças, encapsulamento, interfaces, abstração e uso de enumeradores.

Classe Animal

- Permite criar um objeto animal e armazenar informações como o id, nome, tipo, data de nascimento e a alimentação de cada animal através de uma lista de alimentações.
- Contém método que possibilita a visualização de informações dos animais do zoológico e cálculo da idade.

Classe Pessoa

- É uma superclasse abstrata que define propriedades e métodos que são comuns para os funcionários e os clientes.
- Possui métodos abstratos para obter o número de telefone e email.

Classe Funcionário

- É uma subclasse que herda propriedades da superclasse Pessoa e representa os funcionários do zoológico, incluindo cuidadores, veterinários, voluntários, etc..
- Para além dos dados da superclasse, ela armazena o id, o telefone e o tipo de funcionário.
- Contém método que possibilita a visualização de informações dos funcionários do zoológico, cálculo da idade e os métodos da classe mãe.

Classe Cliente

- Também é uma subclasse que herda atributos da superclasse Pessoa e representa os clientes do zoológico, clientes que entram no zoo para assistir espetáculos e ver animais.
- Para além dos dados da superclasse ela armazena o tipo de cliente e a data de registo.
- Inclui método que possibilita a visualização de informações dos clientes do zoológico e métodos da classe mãe.

Classe Espetáculo

- Garante a gestão e criação de espetáculos realizados no zoo.
- Armazena detalhes como data, duração, tipo de espetáculo e seus participantes (animais, clientes e funcionários).
- Possui métodos que permitem a listagem e visualização de informações dos espetáculos do zoológico.

Classe Bilhete

- Implementa a venda e emissão de bilhetes para os espetáculos, armazenando dados como ID, preço, data de emissão, cliente e assento.
- Método de visualização de informações de todos os bilhetes vendidos.

Classe Habitat

- Responsável por definir o ambiente de cada animal e tem propriedades como o nome do habitat, funcionários responsáveis, animais presentes última manutenção, etc..
- Contém método de contagem de animais no habitat, listagem e visualização de informação dos habitats.

Classe Limpeza Habitat

- É responsável por representar o processo de limpeza de um habitat.
- Armazena informações como o id, a data da última limpeza, funcionários responsáveis pela limpeza.
- Contém método que permite mostrar as informações sobre as limpezas dos habitats, e listagem.

Classe Assistência Veterinária

- Permite o registo e gestão das assistências veterinárias.
- Armazena informações sobre as consultas ou procedimentos médicos realizados nos animais.
- Possui método de exibição de informações sobre as assistências e listagem.

Classe Informações

- Utilizada para armazenar e apresentar informações gerais sobre o zoológico.

Classe Alimentação

- Define a dieta dos animais, o tipo de alimento, a quantidade diária e o número de refeições.
- É utilizada como propriedade na classe Animal, o que permite a gestão da alimentação de cada animal.
- Tem método de exibição de informações das Alimentações e adição de dados em ficheiro.

Classe Genérica

- Usada para gerenciar e tratar listas de diferentes tipos.
- Utiliza uma lista interna (List<T>) para armazenar os objetos.
- Contém métodos, adição, remoção, edição de listas e métodos de adição e leitura de ficheiros.
- Grande parte das classes são tratadas e manipuladas aqui.

Relacionamento Classes

Espetáculos- Clientes, Funcionários, Animais

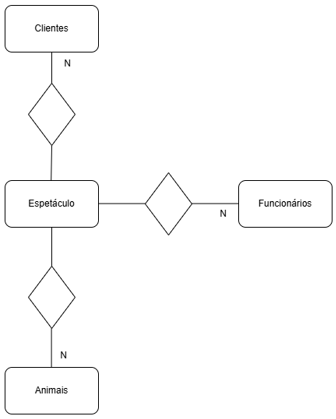

Figura 1 Relacionamentos classe Espetáculo

Cada instância da classe Espetáculo vai ter associado uma lista de animais, clientes e funcionários.

Habitat- Funcionários, Animais

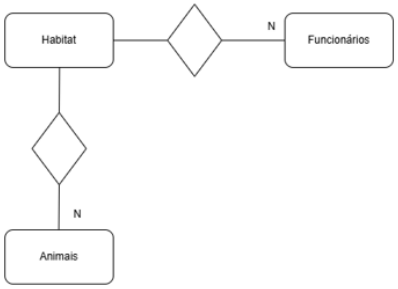

Figura 2 Relacionamentos classe Habitat

Cada instância da classe Habitat vai ter associada uma lista de funcionários e animais.

Limpeza Habitat- Funcionários

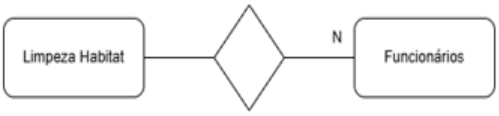

Figura 3 Relacionamento classe Limpeza Habitat

Cada instância da classe Limpeza Habitat vai ter associada uma lista de funcionários.

Assistência Veterinária- Funcionários, Animais

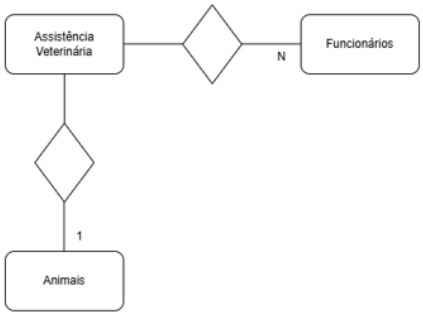

Figura 4 Relacionamentos classe Assistência Veterinária

Cada instância da classe Assistência Veterinária vai ter associada uma lista de funcionários e 1 animal.

Animal- Alimentação

Figura 5 Relacionamento classe Animal

Cada instância da classe Animal vai ter associada uma lista de alimentações.

Exceptions

O uso das exceptions foi bastante importante pois permitiu manter a integridade e segurança do programa.

Quando ocorresse um erro, seja ele argumentos nulos, valores repetidos, dados errados, eles são tratados através de uma mensagem na consola, e em ficheiro, usando os logs. Os erros contêm informação detalhada, de forma que o utilizador possa entender o que se passa e o possa tratar, de forma a que o erro não seja ignorado.

Foi usada a superclasse “Exception” que possui todos os tipos de erros, todas as “Exceptions” herdam dela. Alguns erros eram gerados com uma mensagem personalizada pois alguns métodos não implementavam o tipo de erro desejado.

Logs

O programa desenvolvido da gestão de um zoológico também usa logs que registam informações, dados e erros.

Todas as ações detetadas, quando se manipula listas e ficheiros, são registadas num ficheiro txt, de forma cronológica permitindo solucionar e identificar problemas rapidamente e visualizar a integridade do software.

Testes

Os testes desenvolvidos no programa, são testes implementados na classe Animal e Funcionário, onde é verificado se o tipo de formatação da string corresponde ao esperado, caso contrário o teste falha.

Caso o método ToString() for alterado, ou houver mudanças na estruturação da classe, rapidamente é detetado as alterações no código, serve como documentação viva de como o objeto deve ser representado em texto.

Para a classe Animal e Funcionário também é implementado um teste para verificar se o método de cálculo de idades está a fazer o cálculo de forma correta ou não.

Interface

O programa apenas possui uma interface, sendo ela o IContador. Ela possui um método que permite contar o número de elementos presentes numa lista, a interface apenas está a ser aplicada na classe Habitat mas também pode ser aplicada em muitas outras.

3. Conclusão

Concluindo, o desenvolvimento deste projeto proporcionou-me aplicar os conhecimentos teóricos adquiridos na disciplina de Programação Orientada a Objetos na prática.

A realização deste projeto ajudou-me imenso a aprimorar o meu conhecimento sobre o que é a programação orientada a objetos, como deve ser usada, como deve ser relacionada e boas práticas. Para além do principal, também me ajudou a perceber quando deve ser aplicado e o porquê do uso de exceções, logs e testes unitários.

Através da reflexão sobre as decisões tomadas e dos resultados obtidos, consegui ter um entendimento mais profundo sobre o uso de classes e objetos, e a sua aplicação em problemas computacionais do mundo real.