Curso: CTeSP em Tecnologisa e Desenvolvimento de Sistemas de Informação

Curso: CTeSP em Desenvolvimento Web

Ano Letivo: 2017 / 2018 2º Semestre

Trabalho Prático da Unidade Curricular: Programação Orientada a Objetos Número de Páginas:5

ZooPark

Notas prévias:

- O enunciado é propositadamente vago, genérico e incompleto em alguns pontos. O que se pretende é que os alunos avaliem as opções existentes e escolham a que considerarem mais apropriada para cada uma das situações com que se depararem. Todas as escolhas devem ser referidas e justificadas no relatório.
- O programa deve ser implementado em C#.
- O programa entregue deve ter uma interface simples e amigável, indicando o que pode ser feito em cada situação. Não são valorizados programas com interfaces gráficos.

Introdução

O jardim zoológico ZooPark pediu-lhe para desenvolver um programa em linguagem C# que permita gerir a informação dos seus animais. Em concreto, deverá ser possível conhecer as características básicas dos animais, a sua localização e as suas relações familiares. Ao longo do enunciado são descritas as principais funcionalidades a implementar.

O zoo está dividido num conjunto de espaços, que podem ser uma jaula ou um espaço vedado. Cada uma destas zonas contém alguns animais. A configuração do zoo garante que um dado espaço tem, no máximo, 3 outros locais adjacentes.

Todos os animais existentes no zoo pertencem a uma determinada área, não podendo existir animais sem localização atribuída. Nem todos os animais necessitam do mesmo habitat. Os animais devem ser colocados em áreas que recriem os habitats de que necessitam.

Armazenamento de dados

O programa deve armazenar a informação seguinte:

- Áreas do Zoo
- Animais
- Espécies

Características de cada área:

- ID: identificador alfanumérico único (por exemplo, o nome da área). Uma só palavra
- Habitat: identificador do tipo de habitat recriado nesta área. Uma só palavra
- Capacidade: peso total dos animais que pode comportar.
- Fronteiras: indicação dos espaços adjacentes.

Cofinanciado por:

CENTRO 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

Cada espécie é definida pelo seguinte conjunto de características:

- IDEspecie: Identificador único, comum a todos os animais da mesma espécie. Deve ser uma ou mais palavras;
- Habitat: Lista de habitats indicados para esta espécie

Cada animal é definido pelo seguinte conjunto de características:

- IDEspécie: Identificador da espécie;
- IDAnimal: Identificador único do animal dentro da espécie a que pertence. Deve ser numérico;
- Nome;
- Peso;
- Localização: cada animal está localizado numa das áreas existentes no zoo;
- Família: cada animal tem indicação de quem são os seus pais (caso se encontrem no zoo) e possui também uma lista dos seus filhos que se encontram no zoo.

A informação deve ser lida de um ficheiro para a memória no início do dia, quando o programa arranca, e deve passar de novo para ficheiro no fim do dia quando o programa termina. Durante a execução o programa deve manipular a informação guardada em memória.

Programa a implementar

Gestão das áreas do Zoo

A informação completa sobre as várias áreas do zoo é armazenada num ficheiro de texto. No início da execução, toda a informação sobre as áreas deve ser transferida para memória. Se este ficheiro não existir, o programa deve terminar imediatamente.

Na figura seguinte ilustra-se um ficheiro para um zoo com 4 áreas:

```
AreaA Savana 500 2 AreaB AreaC
AreaB SelvaNebulosa 200 1 AreaA
AreaC Pantano 100 2 AreaA AreaD
AreaD Savana 1000 1 AreaC
```

Neste exemplo, a primeira área tem identificador *AreaA*, que recria o habitat de savana, com 500 kg de capacidade e 2 áreas adjacentes (*AreaB* e *AreaC*).

Cada área apenas pode recriar um habitat, mas mais de uma área pode recriar o mesmo habitat.

Durante a execução apenas deve ser usada/manipulada a informação que se encontra em memória. No final da execução, a informação do vetor deve ser usada para atualizar o ficheiro de texto.

Há duas operações que podem mudar a distribuição das áreas durante a execução:

- Criar uma nova área: o utilizador indica as características da nova área, incluindo as áreas já existentes que vão ficar adjacentes. O programa deve garantir que nenhuma área fique com mais do que 3 adjacências. No processo de criação, a nova área não tem animais,

Cofinanciado por:
CENTRO 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

- Eliminar uma área existente: esta operação só é válida se nenhum animal se encontrar nesta área nessa altura.

Gestão das espécies

Quando o programa não está a ser executado, a informação completa sobre as espécies encontra-se armazenada num ficheiro. No início da execução, esta informação deve ser transferida para memória e mantida numa estrutura dinâmica.
Pode existir informação sobre espécies que não se encontrem presentemente no zoo. De igual forma nem todos os habitats têm de existir representados no zoo.

O programa deve permitir efetuar as seguintes operações sobre as espécies:

1. Listagem da informação:
    a. Listar todas as espécies existentes no ficheiro (apresentando para cada espécie os habitats apropriados para essa espécie)
    b. Listar todas as espécies existentes no zoo, e para cada uma delas os habitats onde se encontram
2. Adicionar:
    a. Adicional espécie e respetiva lista de habitats
    b. Adicionar habitat a uma espécie já existente
3. Remover:
    a. Remover espécie (só podem ser removidas espécies que não estejam representadas no zoo)
    b. Remover habitat de uma dada espécie (não deve ser permitido remover um habitat se no zoo existir algum animal dessa espécie nesse tipo de habitat)

As espécies são referidas pelo seu identificador.

Gestão dos animais do Zoo

Quando o programa não está a ser executado, a informação completa sobre os animais encontra-se armazenada num ficheiro. No início da execução, esta informação deve ser transferida para memória e mantida numa estrutura dinâmica. A implementação pode recorrer a uma ou várias listas, simples ou com outra organização. No final da execução, a informação no ficheiro deve ser atualizada. Caso o ficheiro não exista no início da execução, o programa deve assumir que o zoo ainda não tem animais.

A leitura do ficheiro de animais pode originar vários problemas de incompatibilidade de informação entre animais e locais que devem ser resolvidos de forma simples pelo programa. Exemplos de potenciais problemas são:

- O animal tem indicação de pertencer a uma zona que não existe no zoo: neste caso, o animal pode ser simplesmente ignorado;
- Os animais especificados para pertencer a uma determinada área ultrapassam a sua capacidade. O programa pode ignorar os animais que provocam o ultrapassar do limite.
- O animal tem indicação de pertencer apenas a habitats que não existem no zoo: neste caso, o animal pode ser simplesmente ignorado;

O programa deve implementar soluções igualmente simples para outros problemas que eventualmente possam surgir.

O programa deve permitir efetuar as seguintes operações sobre os animais:

1. Listagem da informação:
    a. Listagem completa de todos os animais do zoo;
    b. Listagem dos animais que se encontram numa determinada localização.
    c. Listagem dos animais que se encontram num determinado habitat.
    d. Listagem completa da informação relativa a um animal, incluindo a identificação dos seus ascendentes e descendentes diretos (caso existam)
2. Eliminar Animais: um animal pode ser retirado do zoo, devendo desaparecer toda a informação relativa a ele (incluindo as relações familiares com outros animais que se mantenham no zoo).
3. Transferir animal: um animal pode ser transferido para uma outra localização dentro do zoo, desde que as áreas envolvidas sejam adjacentes, o habitat adequado e a capacidade do novo local não seja ultrapassada.
4. Nascimento de um animal: um descendente pode ser criado a partir de 1 ou 2 animais que já existam no zoo. Caso estejam 2 progenitores envolvidos no nascimento, deve garantir-se que são da mesma espécie e que se encontram no mesmo local. O nome da cria é indicado pelo utilizador e o programa deve gerar automaticamente um novo identificador único dentro da espécie a que pertence. No processo de nascimento, 20% do peso de cada progenitor é transferido para o descendente.
5. Adicionar animais: O sistema deve verificar se os dados introduzidos são válidos.

Normas para a realização do trabalho prático

O trabalho deve ser **realizado individualmente**. O trabalho só pode ser entregue uma vez e numa única data. A nota obtida é válida em todas as épocas de avaliação do ano letivo 2017/2018.

Data limite de entrega do trabalho prático: **13:00** do dia **24 de Junho de 2018**. Pode ser entregue até às **19:00** desse dia**,** mas terá uma penalização de 50%.

Material a entregar:

- **Até às 13.00 do dia 24/06/2018:** entregar via *moodle* um ficheiro compactado em formato ZIP, contendo o relatório, o código fonte comentado e os ficheiros de dados necessários para o funcionamento do programa. Deverão incluir no ZIP todos os ficheiros do respetivo projeto.
  O nome do ficheiro ZIP deve obrigatoriamente ter o seguinte formato: ***POO_X_NumAluno.zip***, em que X é o CTeSP que frequenta.
- **No início da defesa:** entregar, ao professor responsável pela defesa, uma cópia impressa do relatório. O conteúdo do documento deve ser igual ao submetido via *moodle*.

Defesa:

Os trabalhos serão sujeitos a **defesa obrigatória** em data a anunciar. As defesas poderão incluir:

i) Demonstração do funcionamento do programa;

ii) Explicação detalhada do código;
iii) Implementação de alterações / novas funcionalidades.

As defesas serão feitas nos computadores dos laboratórios. Antes da submissão deverão certificar-se que o trabalho submetido está pronto para correr.

Relatório

Deve ser entregue um relatório contemplando os seguintes pontos:

- Estrutura geral do programa, incluindo a apresentação de algoritmos de alto nível que clarifiquem as suas principais funcionalidades;
- Apresentação das principais estruturas de dados definidas (classes, enums, structs, listas,…), justificando as escolhas feitas;
- Justificação para as opções tomadas em termos de implementação;
- Descrição dos ficheiros utilizados;
- Pequeno manual de utilização.

Avaliação

A cotação do trabalho é de **6 valores**.

Esta componente da avaliação não tem nota mínima.

A deteção de plágio parcial ou total implica a anulação imediata de todos os trabalhos envolvidos.

Critérios de Avaliação:

- Definição das estruturas de dados;
- Funcionalidades implementadas;
- Correção do código implementado, com especial ênfase na manipulação dos dados e dos ficheiros;
- Qualidades das soluções algorítmicas;
- Simplicidade/funcionalidade do interface com o utilizador;
- Estruturação e documentação do código fonte;
- Relatório;
- Defesa.

FIM

Cofinanciado por:
CENTRO 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu