

Programação Orientada a Objetos 2018/2019

Exame da Época Normal – Parte Prática

DEIS – LEI / LEI-PL / LEI-CE

Pretende-se representar sistemas estelares em C++. Um sistema estelar envolve vários tipos de corpos celestes e alienígenas. As características e propriedades destas entidades são descritas nas próximas alíneas. Leia o enunciado todo antes de começar a responder.

Notas

- Podem ser necessárias mais funções do que aquelas que são explicitamente pedidas.
- Todas as classes que fizer (estas ou outras adicionais que entenda necessárias) terão que ser, naturalmente, completas e robustas e coerentes para as operações habituais de C++.
- **Importante:** A resposta tem obrigatoriamente que ser organizada separadamente pelas alíneas pedidas.
- **Importante:** Não separe o código em .cpp e .h. Não se preocupe com #includes e namespaces.

a) Pretende-se representar em C++ o conceito de extra-terrestre que habita um determinado corpo celeste do sistema estelar. Construa uma classe Alien com as seguintes características:

- ↓ • Têm um id numérico único em todo o sistema que lhes é atribuído automaticamente quando nascem (i.e., quando são criados). Cada novo objeto alien que seja criado terá automaticamente o id “seguinte”, tendo o primeiro o id 100.
- ↓ • Em cada momento da sua existência, habitam num determinado corpo celeste. Os aliens conhecem o corpo celeste em que se encontram (i.e., dever poder aceder a ele).
- ↓ • Cada alien possui uma lista de nutrientes essenciais para a sua sobrevivência. Cada nutriente é representado pelo seu nome (*string*). Estes seres só conseguem sobreviver em corpos celestes onde existam, pelo menos, metade dos seus nutrientes essenciais.
- ↓ • Cada alien tem um nível de energia (valor inteiro).
- ↓ • Cada alien tem uma localização especificada como um único valor inteiro superior ou igual a 0. Nesta representação simplificada, pode assumir que o valor representa a distância a um determinado ponto qualquer no corpo celeste em que se encontra.

A classe deve possuir alguns métodos, definindo operações básicas que são essenciais para a sua sobrevivência:

- ↓ • O metabolismo de um alien assegura a gestão da sua energia e funciona da seguinte forma: periodicamente recebe um alimento distribuído pelo corpo celeste em que encontra. Se este alimento fizer parte da sua lista de nutrientes, a energia aumenta 10 unidades; caso contrário diminui 10.
- ↓ • Ao longo da vida, um alien pode desenvolver uma intolerância a um nutriente. Neste caso, o nutriente deve ser retirado da sua lista. A instrução que permite retirar um elemento da lista de nutrientes do alien é sempre especificada da seguinte forma: `objetoET -= “Agua”`; (`objetoET` é um objeto do tipo Alien e “Agua” é apenas um exemplo).
- Um alien pode deslocar-se. Move-se sempre uma unidade de cada vez, podendo aumentar ou diminuir o valor da sua localização. Os operadores unários de pré-incremento e pré-decremento devem ser usados para concretizar o movimento. Por exemplo, se o alien `objetoET` se encontrar na localização 76, a sequência de operações `++ET; --ET; --ET;` vai colocá-lo na posição 75.

A criação de um novo alien obriga a especificar as seguintes características: lista de nutrientes, energia inicial, localização inicial. Os aliens nascem sempre no espaço, e só mais tarde é que são adicionados a um corpo celeste.

b) Construa uma classe `CorpoCeleste` que represente o conceito de Corpo Celeste. Deve ter as seguintes características:

- Na implementação dos atributos desta classe não pode usar contentores da STL. Nas restantes operações (por exemplo, na passagem de informação para funções), ou noutras classes, esta restrição já não se aplica.
- Tem nome, massa e composição (conjunto de materiais que o constituem) Cada elemento material da sua composição é representado pelo seu nome (string). Possui, além disso, um conjunto de habitantes (ou seja, aliens). O corpo celeste é dono de todos os aliens que nele se encontram.
- A escolha do nutriente a usar na alimentação dos aliens e respetiva distribuição a estes é feita pelo corpo celeste. Este processo de escolha e distribuição do nutriente depende do tipo de corpo celeste e não pode ser determinado nesta altura.
- A criação de um novo corpo celeste obriga a especificar todas as suas características, e não conterá nenhum habitante.

Esta classe deve ter os seguintes métodos:

- ↓ • Adicionar um habitante recebido por parâmetro. O corpo celeste toma posse dele. Pode assumir que os habitantes foram criados dinamicamente.
- ↓ • Retirar um habitante do corpo celeste, dado o seu id único. O habitante é devolvido como resultado.
- ↓ • Escolher e distribuir alimento pelos habitantes.
- ↓ • Eliminar habitantes que tenham morrido (energia ≤ 0).

c) Construa uma classe `Planeta` que represente o conceito de um dos possíveis corpos celestes do sistema estelar. Deve ter as seguintes características:

- Tem um período de rotação e tem uma atmosfera. A atmosfera é composta pelo conjunto de elementos que a constituem (nomes dos elementos). Esta informação é sempre especificada no processo de criação de um objeto `Planeta`.
- Num planeta, a seleção do nutriente a usar para alimentar os nativos é feita aleatoriamente de entre todos os elementos da sua composição e da atmosfera.

d) Construa uma classe `Sistema Estelar` com as seguintes características:

- É constituída por um conjunto de corpos celestes. Cada objeto do tipo `Sistema Estelar` toma posse dos corpos celestes que o constituem.

Esta classe deve permitir as seguintes operações:

- Periodicamente deve verificar se existem nos seus corpos celestes habitantes que tenham uma lista de nutrientes essenciais desadequada para o local em que se encontram (menos de metade dos seus nutrientes estão presentes no corpo celeste que habitam). Caso isso aconteça, estes aliens devem ser transferidos para um corpo celeste em que sejam viáveis (pode ser um qualquer, por exemplo, o primeiro que surgir). Se não existir nenhum, o alien deverá ser eliminado. Os métodos que eventualmente tenham que ser adicionados às classes das alíneas anteriores, devem ser referidos e implementados explicitamente nesta alínea.
- O operador de atribuição não deve poder ser usado entre objetos do tipo `Sistema Estelar`.