NOVO - Exercício Extra de Revisão para P1

Faça um programa orientado a objeto (OO), respeitando todas as orientações iniciais, que possibilite o registro e a consulta de todas as doações recebidas por um asilo. O cadastro deste asilo deverá ser feito por meio de seu nome, em uma **string mutável** e nunca vazia (nula), sendo seu armazenamento totalmente em maiúsculo e sem possuir espaços em brancos excessivos (antes ou depois do nome cadastrado).

Realize o cadastro de todas as doações recebidas por este asilo, armazenando seus respectivos dados cadastrais:

- Nome do item doado (pode ter espaço separando os possíveis nomes da doação, por exemplo, pão docê);
- Código inteiro e único (que nunca se repete) que registra esta doação, devendo este código ser sempre informado pelo usuário.

Este código só será válido se não for repetido e maior que o valor definido na constante MAIOR que é 10 (dez). O funcionário deste asilo poderá registrar quantas doações forem desejadas, mas ele nunca saberá quantas doações acontecerão no início do uso do programa (não se sabe quantas doações serão recebidas por dia). Essas doações ainda precisam ser identificadas como Financeira ou Material, pois as financeiras deverão cadastrar o valor (número real) em reais (R\$) e nunca poderá ser menor que R\$1,00. Já para as doações do tipo material será necessário informar a quantidade (número inteiro) doada de tal item, não podendo ser menor que 1.

Após o registro do **nome do asilo**, sua solução deverá mostrar o menu a seguir, garantindo o registro de ao menos uma doação, sem que o programa seja encerrado (sua implementação deverá sempre exigir um registro de doação para poder ser encerrado, ou seja, as opções 2, 3 e 0 (zero) do menu só podem ser executadas se ao menos um cadastro já tiver sido realizado). Reflitam em como implementar este controle que seu programa precisa fazer, pois não será possível **Mostrar**, **Consultar** e nem **Terminar** o programa se o usuário final não tiver feito um cadastro de doação pelo menos).

- 1 Nova doação (registra nova doação para o asilo em questão);
- 2 Mostrar todas as doações (lista todas as doações registradas para este asilo de maneira tabelar em forma de tabela com todos os dados cadastrados);
- **3 Consultar** por nome do conteúdo doado (observe que este nome pode ter nomes repetidos cadastrados e todos deverão ser mostrados);
 - 0 Terminar (mostra a soma das quantidades de todos os materiais doados e a soma de

todos os valores financeiros em uma janela de diálogo do tipo de informação com somente um botão OK disponível). O título dessa janela será Doações Totais e mostrará em duas linhas na mesma JOptionPane a expressão Materiais = seguida da soma das quantidades, enquanto que na linha de baixo será apresentada a expressão Financeiro = R\$ com a soma em seguida. Somente quando o usuário pressionar o botão OK dessa janela o programa será encerrado.

As opções **Mostrar** e **Consultar** (2 e 3) do menu deverão apresentar os seus possíveis resultados de forma tabelar, sempre existindo uma única linha de cabeçalho nesta tabela (Código, Nome, Qtde./Valor(R\$)). Este relatório tabelar começará mostrando, depois de um salto de 5 linhas, o nome do asilo em uma nova linha e depois a tabela com os dados organizados e contendo mais uma coluna inicial à esquerda, que não tem nome no cabeçalho (ou simplesmente #), mas que mostrará sempre o valor correspondente ao item doado, iniciando sempre com o valor um (valor sequencial a partir de **01** - escrito sempre com duas casas como está no exemplo). Todos os registros das doações deverão ser mostrados em uma única linha respeitando a organização deste relatório tabelar (respeitar o cabeçalho por coluna).

A opção de **Mostrar** listará todos os cadastros feitos no programa, enquanto que a opção **Consultar** apresentará somente as doações que satisfazerem o nome do conteúdo doado que será fornecido pelo usuário para pesquisa (filtro por nome), sendo esta satisfação estabelecida por qualquer parte deste nome válido a ser pesquisado (consultar como se pesquisa por parte de uma String e implementar usando a API coerente, como já fizemos em outro exercício já corrigido).

Assim, serão listados todos os registros localizados, tendo também a coluna extra com um valor sequencial iniciado em um (01). As opções **Nova, Mostrar** e **Consultar** serão realizadas quantas vezes o usuário quiser, sendo **encerrado o programa somente se o usuário final escolher a última opção** do menu (**Terminar**), em que a soma das quantidades e dos valores todos de doações deverão ser apresentados em uma janela de diálogo e em duas linhas, como esclarecido anteriormente.

Note que as duas saídas de dados (**Mostrar** e **Consultar**) estarão apresentando o mesmo estilo de relatório com os dados. Assim, reflita sobre sua implementação reaproveitar os recursos da saída de dados (relatório), no qual somente os dados recuperados para apresentação para o usuário final poderão ser diferentes.

É importante destacar que uma doação não poderá ser instanciada (criada na memória) semargumentos em seu método construtor (restrição definida para implementação mais coerente em Orientação a Objetos - objetos são criados pelo construtor). Assim, analise o problema proposto e elabore uma solução Orientada a Objetos, implementando serviços e recursos coerentes com uma solução adequada a este paradigma de programação, além de verificar sempre a entrada de dados para evitar o registro incorreto dos dados necessários aos registros

deste programa, com informações incoerentes, causando interrupção imediata e incorreta em sua execução por meio de exceções. Todos os dados numéricos fornecidos pelo usuário não poderão ser negativos e as strings nunca serão nulas (vazias). Sua solução deverá possuir os pacotes mais coerentes a implementação desejada que veem sendo estudadas a algumas aulas, além do respeito as todas as propriedades de OO que já foram estudantes até o momento dessa atividade.

Respeite o padrão de entrega no MOODLE, incluindo a expressão inicial **aulaRevisaoP1**, onde seu projeto JAVA deverá chamar **aulaRevisaoP1** seguida de seu primeiro nome, como pode ser observado no exemplo:

aula1RevisaoP1_IveteSangalo_13-45678901.zip

Nome do projeto JAVA seria:

RevisaoP1 Ivete

ATENÇÃO!

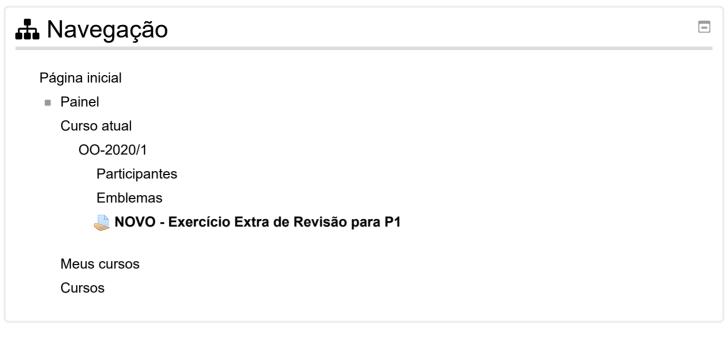
A atividade só será considerada entregue se o estudante encaminhar nesta atividade somente o projeto JAVA exigido na atividade. Caso seja enviado qualquer outra coisa, como o Workspace do ambiente do estudante, esta atividade não será considerada entregue a contento.

Status de envio

Status de envio	Nenhuma tentativa
Status da avaliação	Não há notas
Data de entrega	segunda, 1 jun 2020, 23:30
Tempo restante	5 dias 10 horas
Última modificação	-
Comentários sobre o envio	Comentários (0)

Adicionar tarefa

Fazer alterações em seu envio





Copyright © UnB|CEAD|Centro de Educação a Distância

Campus Universitário Darcy Ribeiro - Brasília - Contato: apoioaprender@ead.unb.br

Todos os direitos reservados