

🏠 Home (/) » SCC0122 (/offerings/view/2108) » Exercício 03 - Catálogo de jogos

Exercício 03 - Catálogo de jogos

Disciplina: SCC0122 - Estruturas de Dados

Prazo de Entrega: 06/11/2022 23:55:55 Fechado

Exercício 03 - Catálogo de jogos eletrônicos

Objetivo

O objetivo deste primeiro exercício prático é reforçar tópicos que já foram vistos em disciplinas anteriores e que serão essenciais para os próximos trabalhos e atividades. Além de reforçar o bom uso de TADs simples, que também serão essenciais para a continuidade da disciplina.

Queremos que os alunos se acostumem com o uso de múltiplos arquivos `.c` e `.h` em um projeto, bem como a construção de um arquivo **Makefile**, responsável por gerenciar a execução do programa.

Descrição

Um site de jogos eletrônicos perdeu o seu catálogo e quer refazê-lo. Você foi contratado para resolver esse problema e deve criar um programa que gerencie os jogos desse site. Isto é, permita o armazenamento e acesso às informações dos produtos e seus dados conforme o pedido do usuário.

Espera-se que seu programa seja capaz atender aos seguintes requisitos:

1. Armazenar as informações de um jogo: Nome , Empresa produtora , Ano de lançamento
2. Buscar pelos jogos lançados em um determinado ano.
3. Buscar pelos jogos criados por uma determinada produtora.

Entrada

Seu algoritmo deve receber os atributos associados aos jogos **um por linha**, na ordem:

- Nome
- Produtora
- Ano

Até ler a letra "F" (Maiúscula).

Após a primeira letra "F" , seu código deve esperar por um próximo carácter

- "A" correspondendo à opção de busca por **ano**
- "E" - correspondendo à opção de busca por **empresa produtora**

Deve continuar realizando as operações requisitadas até ler novamente a letra "F" , marcando o fim do algoritmo.

Exemplo de entrada:

```
Horizon Zero Dawn
Guerrilla Games
2017
Killzone
Guerrilla Games
2004
Crash Bandicoot
Naughty Dog
1999
F
A
2004
A
2000
E
Guerrilla Games
F
```

Saída

Como saída, seu programa deve retornar a listagem dos nomes dos jogos pesquisados de acordo como o parâmetro determinado. Caso não haja jogos registrados que satisfaçam o critério, seu programa deve responder com "Nada encontrado".

Saída da caso acima:

```
Killzone
Nada encontrado
Horizon Zero Dawn
Killzone
```

Observações da implementação

Como é descrito na sessão objetivos desse documento, não queremos apenas que os alunos resolvam o problema, mas que utilizem métodos que serão comuns no decorrer da disciplina.

Devido a esse objetivo, será exigido que vocês desenvolvam seu projeto representando o objeto `jogo` e o objeto `catálogo` de forma abstrata, isto é, com uma `struct` para cada. A memória deve ser alocada **dinamicamente** e ser devidamente liberada ao fim da execução.

Utilizar múltiplos arquivos `.c` e `.h` para separar a implementação e a responsabilidade dos métodos de cada objeto (`jogo.c` e `catalogo.c`).

Construir funções para realizar operações repetitivas, ou seja, modularizar adequadamente seu código.

Escrever um arquivo **Makefile** que será responsável por gerenciar a execução do projeto dentro da plataforma *run.codes*.

Observações da avaliação

A avaliação do seu programa será feita além do resultado da plataforma *run.codes*. Portanto, ter um bom resultado com os casos de teste, não será suficiente para garantir a **nota máxima** e nem a **aprovação do exercício**.

Caso seu projeto não satisfaça os pontos exigidos nos **objetivos** e explicitados nas **observações de implementação**, sua nota poderá ser reduzida ou ser desconsiderada.

Cópias de código entre alunos, acusadas pela plataforma, resultarão imediatamente em **zero** aos dois ou mais alunos envolvidos.

[Esconder Descrição](#)

Este exercício aceita os seguintes tipos de arquivos:

[ZIP Makefile](#)

Atenção: Para ser corretamente corrigido, seu código, se entregue em um único arquivo Zip/Makefile:

- deve obrigatoriamente ter um arquivo Makefile na raiz do arquivo zip;
- deve conter **apenas** o comando de **compilação** na diretiva "all" (ou seja, será executado o comando "make all" para compilar o seu código);
- deve conter **apenas** o comando de **execução** na diretiva "run" (ou seja, será executado o comando "make run" para execução do seu código);

Caso seu arquivo Makefile não siga as instruções acima, seu código não será corrigido. Se você tem alguma dúvida, entre em contato com support@run.codes.

[📄 Baixar Casos de Teste \(/Exercises/downloadCases/27218\)](/Exercises/downloadCases/27218)

Novo Envio

[📅 \(/Exercises/exportExerciseToGoogleCalendar/27218\)](/Exercises/exportExerciseToGoogleCalendar/27218)

O exercício está fechado

06/11/2022 23:55:55

[🔒 Fechado](#)

Meu Último Envio

[📄 Download \(/Commits/download/1908393\)](/Commits/download/1908393)

status

Finalizado

compilado

Sim

casos corretos

8/8

pontuação

10.00

Caso	Status	Tempo de CPU	Tam. de Memória Utilizado	Mensagem
------	--------	--------------	---------------------------	----------

Caso	Status	Tempo de CPU	Tam. de Memória Utilizado	Mensagem
Caso 1	Correto	0.0027 s	-1 Kb	Resposta Correta
Caso 2	Correto	0.0029 s	-1 Kb	Resposta Correta
Caso 3	Correto	0.0025 s	-1 Kb	Resposta Correta
Caso 4	Correto	0.0026 s	-1 Kb	Resposta Correta
Caso 5	Correto	0.0025 s	-1 Kb	Resposta Correta
Caso 6	Correto	0.0027 s	-1 Kb	Resposta Correta
Caso 7	Correto	0.0031 s	-1 Kb	Resposta Correta
Caso 8	Correto	0.0027 s	-1 Kb	Resposta Correta

Detalhes dos Casos de Teste

Selecione um caso de teste...



Histórico de Entregas

Data	Status	Corretos	Notas	Ações
16/10/2022 17:03:32	Finalizado	8/8	10.00	Download (/Commits/download/1908393) Detalhes (/commits/details/1908393)
16/10/2022 16:52:19	Finalizado	8/8	10.00	Download (/Commits/download/1908367) Detalhes (/commits/details/1908367)
16/10/2022 16:49:21	Incompleto	0/8	0.00	Download (/Commits/download/1908359) Detalhes (/commits/details/1908359)
16/10/2022 16:45:19	Erro	0/8	0.00	Download (/Commits/download/1908350) Detalhes (/commits/details/1908350)