Trabalho Prático I - TEP - Biblioteca

Prof. Vinicíus Mota DI/UFES

Data de entrega: 18/08/2021 Locais de entrega:

- submetido pelo Repl.it;
- E também pelo AVA.

1. Objetivos

Este primeiro trabalho visa o desenvolvimento das habilidades de criação de tipos abstratos de dados, bibliotecas, organização e documentação correta de código, gerenciamento de memória e compilação automática de código. Para isto, o objetivo deste trabalho é o desenvolvimento de um sistema de gerenciamento de uma locadora de livros. Considere um sistema de biblioteca simples no terminal, capaz de listar, adicionar, pesquisar clientes, livros e locações efetuadas. As funcionalidades e especificações do sistema (quanto a formato de arquivos, como compilar, etc.) seguem nesse arquivo, mas estão sujeitas a mudanças:

2. Descrição do sistema

O objetivo deste trabalho é simular um serviço de locadora de livros. Para isto, o usuário do programa deve cadastrar clientes, livros, autores e locações efetuadas. Será fornecido 3 arquivos .csv onde cada um contém informações de clientes, livros e locações. O programa deve ler esses arquivos e armazenar as informações de cada um.

O usuário deve ser capaz de visualizar a lista de livros disponíveis, as informações sobre os livros (metadados), pesquisar por um livro e escolher livros para locacao. Deve ser feito o tratamento da entrada do usuário, para caso o usuário digite alguma opção inválida o programa saiba lidar com isso.

O usuário do sistema também poderá gerar relatórios acerca de clientes, livros e locações. Os relatórios devem ser gerados em forma de arquivos de saída.

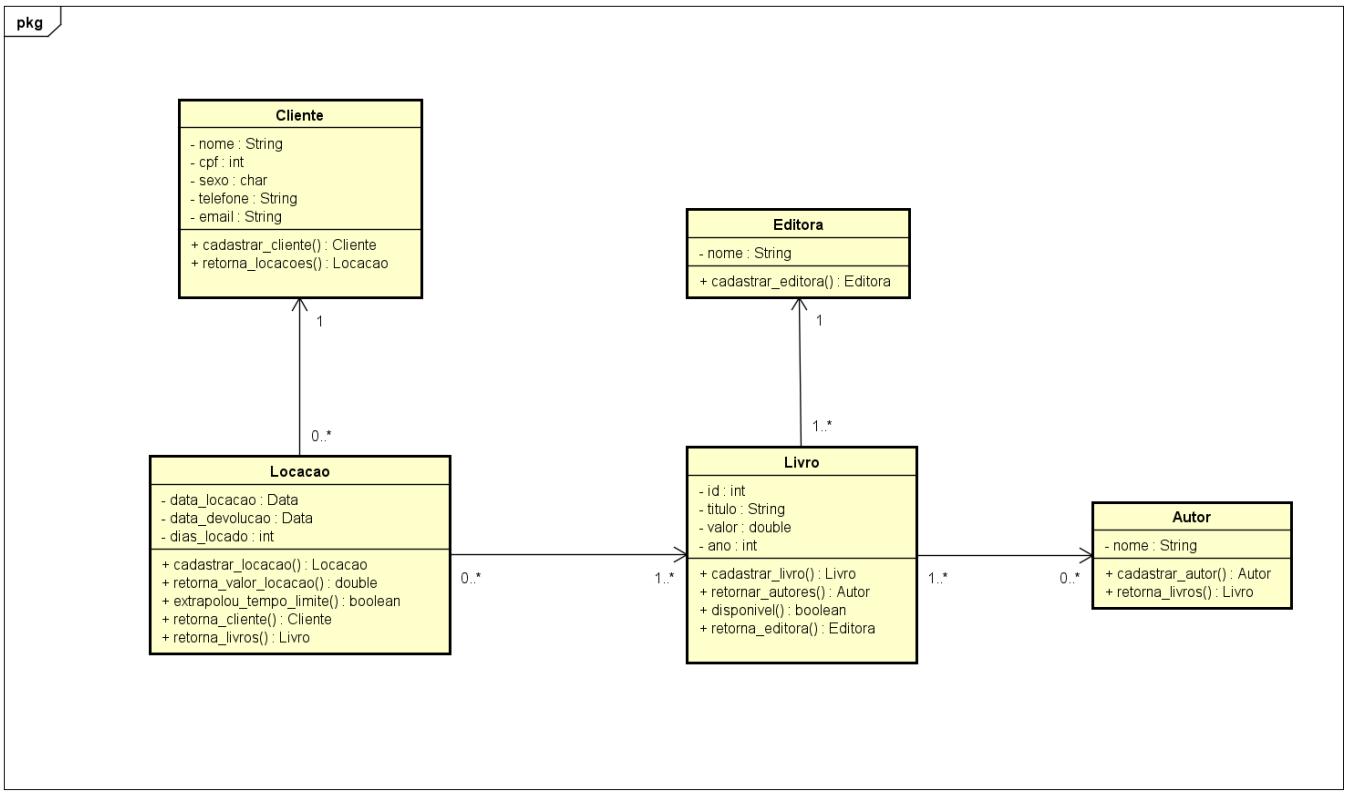
IMPORTANTE:

- O código deve ser devidamente separado em bibliotecas com os devidos cabeçalhos e implementações, sempre levando em consideração as boas práticas.
- É responsabilidade do desenvolvedor garantir que o programa faz uso eficiente da memória.
- É seu dever como desenvolvedor filtrar erros de entrada dos usuários. Mesmo que não estejam listados aqui. Sempre comente essas verificações.
- A compilação do executável deve ser por meio de um Makefile.
- Código deve estar bem comentado.
- É opcional "limpar o terminal" entre as passagens do menu de opções.

2.1 Funcionalidades

O programa funcionará com um menu do sistema. Isso deve ser feito no terminal a partir da leitura de números que representam decisões do usuário. Considere que ao iniciar o programa o usuário pode indicar se ele quer que o menus sejam impressos ou não.

A Figura abaixo apresenta de forma sumarizada algumas as funcionalidade que o sistema deve conter:



Ao executar o programa, deve ser carregado na memória a lista de filmes disponíveis e os usuários cadastrados. O Formato dos arquivos é discutido na próxima seção.

2.1.1 Menu

A primeira "tela" que o usuário deve ver é uma tela de menu com as opções:

```
1- Cadastrar
2- Pesquisar
3- Locação
4- Relatórios
5- Devolução
6- Sair
```

Opção 1 - Cadastrar

Um novo menu aparece possibilitando o usuário do sistema o cadastro Livros e Clientes. Use a opção 3 para voltar ao menu anterior ou 4 para terminar o programa.

```
1- Cadastrar Cliente
2- Cadastrar Livro
3- Voltar
4- Sair
```

Opção 1.1 - Cadastrar Cliente

Permite cadastrar clientes no sistema. É necessário inserir informações como nome(string), cpf(inteiro), sexo(char),telefone(string) e e-mail(string). Como no exemplo:

```
Nome: <entrada><Str>
CPF: <entrada><Int>
Sexo: <entrada><Char>
Telefone: <entrada><Str>
E-mail: <entrada><Str>
```

Se o cliente já existe, deve ser informado a mensagem:

```
Cliente já cadastrado.

1- Voltar
2- Sair
```

A verificação de existência pode ser feita a partir do CPF do cliente.

Opção 1.2 - Cadastrar Livro

Permite cadastrar livros no sistema. É necessário inserir informações como título(string), Id único(inteiro), autores sera dado separado por '/', ou seja, quantidade de autores indefinida, editora(string), valor da diária(double,float), ano(inteiro). Como no exemplo:

```
Título: <entrada><Str>
ID: <entrada><Int>
Autores: <entrada><Str> / <entrada><Str> / <entrada><Str>...
Nota: <double>
Editora: <entrada><Str>
Valor: <entrada><Double>
Ano: <entrada><Int>
```

Se o livro já existir, será adicionado um novo exemplar do livro no catálogo. Apesar de existirem vários exemplares de livros com mesmo nome, o ID de cada livro é único, não pode existir dois livros com o mesmo ID. Fica a cargo do aluno como implementar o número de exemplares, podendo ser uma flag (int) de quantidade em uma estrutura contendo Livros.

Caso algum autor não exista no sistema, deverá ser criado um novo autor e fazer o link do livro com o autor e/ou do autor com o livro. Caso já exista não será necessário criar um novo, apenas fazer o link. O mesmo vale para editora.

Opção 2 - Pesquisar

Um novo menu aparece possibilitando o usuário do sistema pesquisar por Livros, Clientes e Autores. Use a opção 4 para voltar ao menu anterior.

```
1- Pesquisar Cliente
2- Pesquisar Livro
3- Pesquisar Autor
4- Voltar
5- Sair
```

Opção 2.1 - Pesquisar Cliente

Permite pesquisar um cliente a partir de seu CPF. O sistema deverá mostrar no terminal as informações do cliente e os livros que ele já locou. Como no exemplo:

```
CPF do Cliente: <entrada><Int>
```

Informações a serem mostradas:

```
Nome: <saida>
CPF: <saida>
Sexo: <saida>
Telefone: <saida>
E-mail: <saida>

Livros locados: <saida>, <saida>, ..., <saida>

1- Voltar
2- Sair
```

Caso o cliente não exista deverá ser exibida a seguinte mensagem:

```
Cliente inexistente.

1- Voltar
2- Sair
```

Opção 2.1 - Pesquisar Livro

Permite pesquisar um livro a partir de seu título. O sistema deverá mostrar no terminal as informações do livro e a quantidade de vezes que foi locado. O número de exemplares deve ser considerado neste cálculo de quantidade. Como no exemplo:

```
Título do Livro: <entrada><Str>
```

Informações a serem mostradas:

```
Título: <saida>
Autores: <saida>, <saida>, ..., <saida>
Editora: <saida>
Nota: <saida>
Valor: <saida>
Ano: <saida>

Quantidade de exemplares: <saida>
Número de vezes que foi locado: <saida>

1- Voltar
2- Sair
```

Caso o livro não exista deverá ser exibida a seguinte mensagem:

```
Livro inexistente.

1- Voltar
2- Sair
```

Opção 2.1 - Pesquisar Autor

Permite pesquisar um autor a partir de seu nome. O sistema deverá mostrar no terminal as informações do autor e os livros que ele participou. O usuário pode selecionar um livro para exibir informações do mesmo. Para selecionar um livro basta digitar um número de 3 até a quantidade de livros que o autor possui, o digito 1 e 2 nesses casos será sempre as teclas de Voltar e Sair para facilitar a programação. Como no exemplo:

```
Nome do Autor: <entrada><Str>
```

Informações a serem mostradas:

```
Nome: <saida>

Livros Publicados: <saida>, <saida>, ..., <saida>

1- Voltar
2- Sair
```

Opção 3 - Locação

Permite o usuário do sistema fazer uma locação de livro para um cliente. É necessário o CPF do cliente, que caso não existe deverá ser criado um novo cliente, e o ID do livro. Um livro só poderá ser locado caso o mesmo esteja disponível (pode usar a flag de quantidade de exemplares por exemplo, ou criar um boolean indicado disponibilidade). Após digitar o CPF e ID, será mostrado no terminal o nome do cliente, o título do livro e o valor da locação do livro. Também será pedido para digitar a quantidade de dias que o cliente quer ficar com o livro, após isso será mostrada as informações novamente juntamente com a data de locação e devolução do livro e o valor total da locação (valor da diária do livro x dias de locação) e o usuário deverá confirmar a locação exibindo uma mensagem de confirmação. Se o livro não existir ou não estiver disponível o usuário deverá indicar outro livro. Como no exemplo:

```
CPF do Cliente: <entrada><Int>
ID do Livro: <entrada><Int>
```

Informações a serem mostradas:

```
Nome do Cliente: <saida>
Título do Livro: <saida>
Valor: <saida>

Digite a quantidade de dias para locação: <entrada><Int>
```

Informações a serem mostradas:

```
Nome do cliente: <saida>
Título do Livro: <saida>
Valor: <saida>
Data de locação: <saida> (dia/mês/ano)
Data de devolução: <saida> (dia/mês/ano)
Valor total: <saida>

Confirmar Locação(S/N): <entrada><Char>
```

Informações a serem mostradas:

```
Locação Confirmada!

1- Voltar
2- Sair
```

Informações a serem mostradas caso o cliente não exista:

```
Cliente inexistente! Criando um novo cliente.

Nome: <entrada><Str>
CPF: <entrada><Int>
Sexo: <entrada><Char>
Telefone: <entrada><Str>
E-mail: <entrada><Str>
```

Opção 4 - Relatórios

Permite ao usuário do sistema gerar relatórios acerca de clientes, livros e locações. Os relatórios devem ser gerados em arquivo de saida com os seguintes nomes: relatorio_clientes.txt, relatorio_livros.txt e relatorio_locacoes.txt. O seguinte menu será aprensentado:

```
1- Relatório de Clientes
2- Relatório de Livros
3- Relatório de Locações
4- Voltar
5- Sair
```

Opção 4.1 - Relatório de Clientes

Gera um arquivo de saída chamado relatorio_clientes.txt que deverá listar todos os clientes cadastrados juntamente com suas locações feitas, seguindo o padrão:

```
Nome: <saida>
CPF: <saida>
Locações:
    Titulo do Livro: <saida>
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>

    Titulo do Livro: <saida>
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>

    .
    .
    .

    Titulo do Livro: <saida>
```

```
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>

Nome: <saida>
CPF: <saida>
Locações:
    Titulo do Livro: <saida>
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>

    Titulo do Livro: <saida>
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>

    .
    .
    .

    Titulo do Livro: <saida>
    Data de Locação: <saida>
    Data de Devolução: <saida>
    Valor Total: <saida>
```

Opção 4.2 - Relatório de Livros

Gera um arquivo de saída chamado relatorio_livros.txt que deverá listar todos os livros cadastrados ordenados por quantidade de vezes que foi locado, seguindo o padrão:

```
Título: <saida>
Autores: <saida>, <saida>, ..., <saida>
Valor: <saida>
Locado por <saida> vezes

Título: <saida>
Autores: <saida>, <saida>, ..., <saida>
Valor: <saida>
Locado por <saida> vezes

.
.
.

Título: <saida>
Autores: <saida>, <saida>, ..., <saida>
Valor: <saida>
Locado por <saida> vezes
```

Opção 4.3 - Relatório de Locações

Gera um arquivo de saída chamado relatorio_locacoes.txt que deverá listar todas as locações ativas no momento, indicando se a devolução está atrasada ou não, seguindo o padrão:

```
Nome do cliente: <saida>
Título do Livro: <saida>
ID do Livro: <saida>
Data de locação: <saida> (dia/mês/ano)
Data de devolução: <saida> (dia/mês/ano)
Valor total: <saida>
Atrasado: <saida>

Nome do cliente: <saida>
Título do Livro: <saida>
ID do Livro: <saida>
Data de locação: <saida> (dia/mês/ano)
Data de devolução: <saida> (dia/mês/ano)
Valor total: <saida>
Atrasado: <saida>

.
.
.

Nome do cliente: <saida>
Título do Livro: <saida>
ID do Livro: <saida>
Data de locação: <saida> (dia/mês/ano)
Data de devolução: <saida> (dia/mês/ano)
Valor total: <saida>
Atrasado: <saida>
```

Opção 5 - Devolução

Permite ao usuário do sistema devolver um livro locado por um cliente. Será necessário informar apenas o ID do livro e será mostrado as informações da locação. Também deverá ser mostrado se a devolução está atrasada ou não, e caso esteja o cliente deverá pagar uma multa de 10% por dia de atraso em cima do valor do livro. O usuário deverá confirmar a devolução. Como no exemplo:

```
ID do Livro: <entrada>
```

Informações a serem mostradas:

```
Nome do cliente: <saida>
Título do Livro: <saida>
Data de locação: <saida> (dia/mês/ano)
Data de devolução: <saida> (dia/mês/ano)
Valor total: <saida>
Atrasado: <saida>

Confirmar devolução(S/N): <entrada>
```

Informações a serem mostradas:

```
Devolução confirmada.

1- Voltar
2- Sair
```

Opção 6 - Sair

Sempre que o usuário sair do programa pelo menu, deve-se garantir que quaisquer dados que precisam ser persistidos já foram salvos em arquivos, ou seja, se for cadastrado um novo livro, autor ou cliente, deve ser registrado ao final do arquivo que foi dado como entrada. Por exemplo, adicionou um novo cliente no sistema, então o arquivo de entrada inicial cliente.txt deverá ter esse novo cliente ao final do arquivo após sair do programa.

Além disso, é dever do desenvolvedor garantir que toda a memória foi liberada. Vazamentos de memória serão penalizados.

3. Arquivos de entrada

O sistema realiza a leitura de três arquivos `.csv`: o arquivo de clientes, livros e locações. Suas informações são separadas por vírgula simples.

Para o arquivo de clientes, são dispostas as informações

```
nome, cpf, sexo, telefone, email
```

onde nome, telefone e email são strings de, no máximo, 128 caracteres, sexo é um caracter e cpf é um inteiro.

Para o arquivo de livros, são dispostas as informações

```
id, titulo, autor1/autor2/...autorn, nota, valor, data de publicação, editora, qtd
de exemplares
```

em que titulo, autores e editora são strings, qtd. de autores e ano são inteiros e o valor é um float/double.

3.1 Dados disponíveis

4. Compilação e execução

O arquivo `Makefile` compila e gera o executável `tp1`. Na compilação ele procura arquivos cabeçalhos na pasta `include`, códigos fonte em `src` e o arquivo que contém a função main em `client`. Ao executar `make`, modifique a variável `CLI` para mudar o arquivo cliente. Por exemplo,

```
make CLI=client/main2.c
```

utilizaria a função main do arquivo `client/main2.c`. Por padrão, o makefile usa o arquivo `client/main.c`.

Para rodar o trabalho, execute o comando

```
./tp1
```

5. Regras Gerais

O trabalho deverá ser feito **em dupla** e pelos próprios alunos, isto é, os alunos deverão necessariamente conhecer e dominar todos os trechos de código criados.

Cada dupla deverá trabalhar independente das outras, não sendo permitido a cópia ou compartilhamento de código. O professor irá fazer verificação automática de plágio. Trabalhos identificados como iguais, em termos de programação, serão penalizados com a nota zero. Isso também inclui a dupla que forneceu o trabalho, sendo, portanto, de sua obrigação a proteção de seu trabalho contra cópias ilícitas.

6. Entrega do Trabalho

O trabalho deverá ser submetido pelo Repl.it e também pelo AVA até as 23:59 do dia 11/04/2021.

O arquivo enviado pelo AVA deve seguir o padrão: TEP-2021_MATRICULA1_MATRICULA2.zip, substituindo a matrícula pelos respectivos números de matrícula da dupla.
O arquivo zip deve conter: Todos os arquivos necessários para compilação (cabeçalhos e códigos fontes) `Makefile` que compile o programa corretamente em um ambiente Linux. `README.md` contendo nome da dupla e com considerações de projeto que julgue necessário, incluindo as decisões de projeto tomadas. Por exemplo, explicação breve das bibliotecas, etc.

O código fonte deverá estar bem comentado, isto é, todas os structs e funções devem ter descrições claras do que ela faz, o que representa cada parâmetro e o que ela retorna (se houver). Pode-se adicionar também comentários que ajude a entender a lógica da questão.

7. Avaliação

O trabalho será corrigido em dois turnos: correção do professor e entrevista. A correção do professor (CP) gerará uma nota entre 0 e 10 e as entrevistas (E) serão realizadas posteriormente à data de entrega do

trabalho e será atribuída uma nota entre 0 e 1. Alunos que fizeram o trabalho em dupla farão a entrevista juntos porém terão seus desempenhos na entrevista avaliados de forma individual. Pequenas e pontuais correções de código serão permitidas no momento da entrevista, o que poderá acarretar em uma CP maior.

A nota final (NF) do trabalho será dada pela seguinte fórmula: NF = E * CP

O trabalho será pontuado de acordo com sua funcionalidade e outros aspectos de implementação, conforme as listas abaixo.

Aspectos funcionais

1. Inicialização. Capacidade de carregar os arquivos usando estruturas dinâmicas. (5%)
2. Cadastrar e persistir clientes, livros, locações. Isto é, ao cadastrar um cliente e sair do programa, este usuário é salvo no mesmo arquivo de entrada. O mesmo vale para livros e locações. (10%)
3. Pesquisar clientes e livros(por nome e por autor). ((5+5+5)=15%)
4. Locação de livros para clientes. (10%)
5. Devolução de livros. (10%)
6. Gerar relatórios de clientes, livros e locações ((5+5+5)=15%)

Aspectos de desenvolvimento:

1. Uso correto de TADs para abstração dos elementos envolvidos no trabalho. Menos *main* mais TADs! (20%)
2. Executar o programa com o valgrind, sair corretamente e não ter vazamento de memória. (20%)