EXERCÍCIO - HERANÇA

PROFESSOR: ADELARDO ADELINO DANTAS DE MEDEIROS

O objetivo é desenvolver uma versão bastante simplificada de um armazenador do estoque de uma loja que demonstre o conceito de herança entre classes em C++.

1. Crie uma classe base, `Produto`, que armazene as informações básicas sobre um item do estoque de uma loja:
   - nome
   - preço (inteiro: valor em centavos)
2. Crie, usando herança, uma hierarquia de classes para uma loja que venda livros, CDs e DVDs, com as seguintes informações:
   - Para livros: nome, preço e autor;
   - Para CDs: nome, preço e número de faixas;
   - Para DVDs: nome, preço e duração.

   Essas classes herdam da classe `Produto`.
3. Em seguida, crie a classe `Loja` e um programa principal que permita adicionar produtos à loja e imprimir a lista de produtos, usando os métodos da classe `Loja`.
4. Devem ser previstas funcionalidades para ler e salvar o estoque em arquivo.

CLASSES

Devem ser criadas as seguintes classes:

- `Produto`
- `Livro`
- `CD`
- `DVD`
- `ListaLivro`
- `ListaCD`
- `ListaDVD`
- `Loja`

`Produto`

Armazena o nome (string C++) e preço (inteiro) de qualquer item do estoque. Deve prever ao menos as seguintes funcionalidades:

- `ler`: lê as informações de um parâmetro `istream`. Retorna false em caso de erro.
- `salvar`: escreve as informações em uma `ostream` passada como parâmetro.
- `digitar`: permite que o usuário digite as informações da classe.
- `imprimir`: exibe em tela as informações. Definido usando o método `salvar`.

`Livro, CD, DVD`

Herda da classe `Produto` por herança pública. Armazena o dado específico para cada tipo de item. Por exemplo, o nome do autor (string C++) para livros. Deve prever ao menos as seguintes funcionalidades:

- `ler`: chama o método `ler` da classe `Produto` e em seguida lê as informações específicas da classe de uma `istream`.
- `salvar`: chama o método `salvar` da classe `Produto` e em seguida escreve as informações específicas da classe em uma `ostream`.
- `digitar`: chama o método `digitar` da classe `Produto` e em seguida permite que o usuário digite as informações específicas da classe
- `imprimir`: exibe em tela as informações. Definido usando o método `salvar`.
- `operator>>`: utiliza o método `digitar`.
- `operator<<`: utiliza o método `salvar`.

`ListaLivro, ListaCD, ListaDVD`

Array dinâmico de [`Livro, CD, DVD`] que armazena uma quantidade arbitrária de itens do tipo específico. Guarda o número de itens (tamanho do array) e o ponteiro para a área que contém os dados. Deve prever ao menos as seguintes funcionalidades:

- `incluir`: acrescenta no array um [`Livro, CD, DVD`] (passado como parâmetro).
- `excluir`: exclui do array o item de índice i (passado como parâmetro), caso exista. Retorna false em caso de erro na exclusão.
- `ler`: lê de uma `istream` uma lista de [`Livro, CD, DVD`]. O arquivo deve conter na primeira linha uma palavra-chave e o tamanho da lista (número de itens) e, em seguida, as informações de cada item. As informações dos itens devem ser lidas pelo método `ler` da classe [`Livro, CD, DVD`]. Retorna false em caso de erro na leitura.

- `salvar`: escreve em uma `ostream` uma lista de [`Livro, CD, DVD`]. O arquivo deve conter na primeira linha uma palavra-chave e o tamanho da lista (número de itens) e, em seguida, as informações de cada item. As informações dos itens devem ser escritas pelo método `salvar` da classe [`Livro, CD, DVD`].
- `imprimir`: exibe em tela o índice [0 a N-1] e as informações para cada um dos itens da lista, chamando várias vezes o método `imprimir` da classe [`Livro, CD, DVD`]. O formato de cada linha deve ser igual ao formato descrito para salvamento em arquivo (ver na seção ARQUIVO), precedido do índice (0 a N-1) de cada livro. A lista deve ser precedida de um texto explicativo livre. Por exemplo:

```
>> LIVROS:
0) L: ”Memorial”;$32.34;”Pedro”
1) L: ”Minha Vida”;$2.86;”Adolf”
2) L: ”Poemas”;$13.14;”João Sá”
```

`Loja`

Contém (por composição) uma `ListaLivro`, uma `ListaCD` e uma `ListaDVD`. Deve prever ao menos as seguintes funcionalidades:

- métodos [`incluir, excluir`][`Livro, CD, DVD`] que chamam os métodos [`incluir, excluir`] das classes `Lista`[`Livro, CD, DVD`]. Por exemplo, o método `incluirCD` deve chamar o método `incluir` da classe `ListaCD`. Os métodos `excluir`[`Livro, CD, DVD`] retornam false em caso de erro na exclusão.
- `ler`: chama consecutivamente o `ler` das classes `Lista`[`Livro, CD, DVD`] (3 chamadas). O nome do arquivo é passado como um parâmetro `string`, que gera a abertura de uma `istream` a ser passada como parâmetro para o método `ler` das classes `Lista`[`Livro, CD, DVD`]. Retorna false em caso de erro na leitura.
- `salvar`: chama consecutivamente o `salvar` das classes `Lista`[`Livro, CD, DVD`] (3 chamadas). O nome do arquivo é passado como um parâmetro `string`, que gera a abertura de uma `ostream` a ser passada como parâmetro para o método `salvar` das classes `Lista`[`Livro, CD, DVD`]. Retorna false em caso de erro no salvamento.
- `imprimir`: chama consecutivamente o `imprimir` das classes `Lista`[`Livro, CD, DVD`] (3 chamadas).

ARQUIVO

Os itens devem ser armazenados em arquivo, com um item por linha. A forma de armazenamento deve ser tal que a leitura/escrita de uma classe derivada utilize a leitura/escrita da classe base, acrescentando mais informação.

`Produto`

"STRING_NOME";$FLOAT_PRECO
A informação é composta pela string do nome do produto, delimitada por aspas (“), seguida de um ponto-e-vírgula (;) e de um cifrão ($); após vem o preço do produto, escrito como um número em ponto flutuante com dois decimais.
Exemplo:
`”Memorial”;$32.34`

`Livro`

`L:` *<Produto>;"STRING_AUTOR"*
A informação é composta pelos caracteres L, dois pontos (:) e espaço, seguida pelo conteúdo do `Produto`; em seguida, há um ponto-e-vírgula (;) e a string do nome do autor, delimitada por aspas (“).
Exemplo:
`L: ”Memorial”;$32.34;”Pedro”`

`CD`

`C:` *<Produto>;INT_NUM_FAIXAS*
A informação é composta pelos caracteres C, dois pontos (:) e espaço, seguida pelo conteúdo do `Produto`; em seguida, há um ponto-e-vírgula (;) e o número de faixas (inteiro)
Exemplo:
`C: ”Memorial”;$32.34;15`

`DVD`

`D:` *<Produto>;INT_DURACAO*
A informação é composta pelos caracteres D, dois pontos (:) e espaço, seguida pelo conteúdo do `Produto`; em seguida, há um ponto-e-vírgula (;) e a duração em minutos (inteiro)
Exemplo:
`D: ”Memorial”;$32.34;131`

`ListaLivro`

Inicia com uma linha contendo a palavra `LISTALIVRO` e número N de livros, separados por espaço. Em seguida, em cada nova linha, há a informação de um livro.
Exemplo:

```
LISTALIVRO 3
L: ”Memorial”;$32.34;”Pedro”
L: ”Minha Vida”;$2.86;”Adolfo”
L: ”Poemas”;$13.14;”João de Sá”
```

`ListaCD`

Inicia com uma linha contendo a palavra `LISTACD` e número N de CDs, separados por espaço. Em seguida, em cada nova linha, há a informação de um CD.
Exemplo:

```
LISTACD 2
C: ”Solidão”;$27.86;12
C: ”Verso Azul”;$8.52;14
```

`ListaDVD`

Inicia com uma linha contendo a palavra `LISTADVD` e número N de DVDs, separados por espaço. Em seguida, em cada nova linha, há a informação de um DVD.
Exemplo:

```
LISTADVD 4
D: ”Combate VI”;$14.38;126
D: ”Barroco”;$8.52;72
D: ”Nunca Mais!”;$11.00;185
D: ”Eu, Tu, Eles”;$7.00;134
```

`Loja`

A informação é composta pela junção das informações da `ListaLivro`, `ListaCD` e `ListaDVD`, nessa ordem.
Exemplo:

```
LISTALIVRO 3
L: ”Memorial”;$32.34;”Pedro”
L: ”Minha Vida”;$2.86;”Adolfo”
L: ”Poemas”;$13.14;”João de Sá”
LISTACD 2
C: ”Solidão”;$27.86;12
C: ”Verso Azul”;$8.52;14
LISTADVD 4
D: ”Combate VI”;$14.38;126
D: ”Barroco”;$8.52;72
D: ”Nunca Mais!”;$11.00;185
D: ”Eu, Tu, Eles”;$7.00;134
```