ACH 2003 – Computação Orientada a Objetos
EACH – Primeiro Semestre de 2009
Primeiro Exercício Programa – Turma 02
Professor: Marcos L. Chaim

1 Descrição

Imagine que você queira organizar os arquivos MP3 armazenados no seu disco rígido ou no seu *MP3 player*. Tipicamente, as informações que irão ser classificadas nesta organização são o `nome da música` (35 caracteres, no máximo), `nome do artista` (35 caracteres, no máximo), `localização no disco rígido` (125 caracteres, no máximo) e `preço` (isto se você comprou o MP3... senão o preço é zero). Escreva uma classe `AcervoMP3` que possui elementos da classe `MP3`. Você deve implementar a os seguintes métodos:

1. `insereMP3(nome, artista, localizacao, preço)` – este método insere um novo MP3 que possui os valores passados por meio dos parâmetros `nome`, `artista`, `localizacao` e `preço`. Este método deve emitir uma mensagem "Registro inserido", se houve a inserção do registro, ou "Registro nao inserido", se o MP3 não foi inserido.

2. `eliminaMP3(nome, artista)` – este método elimina um MP3 que possui os valores passados por meio dos parâmetros `nome` e `artista`. Este método deve emitir uma mensagem "Registro eliminado", se houve a eliminação do registro, ou "Registro inexistente", se o MP3 não foi cadastrado anteriormente.

3. `listaEmOrdemDeNome()` – lista em ordem alfabética em função do nome da música o conteúdo do acervo. Todas as informações sobre o MP3 são listadas; o conteúdo de cada atributo é listado em uma linha. Se o acervo estiver vazio, o método deverá imprimir: "Acervo de MP3 vazio".

4. `listaEmOrdemDeArtista()` – lista em ordem alfabética em função do nome do artista o conteúdo do acervo. Todas as informações sobre o MP3 são listadas; o conteúdo de cada atributo é listado em uma linha. Se o acervo estiver vazio, o método deverá imprimir: "Acervo de MP3 vazio".

5. `listaMusicasDeArtista(artista)` – lista músicas de um determinado artista. Todas as informações sobre o MP3 são listadas; o conteúdo de cada atributo é listado em uma linha. Se não houver MP3 cadastrado para o artista, o método deverá imprimir: "Acervo de MP3 vazio".

6. `selecionaAleatorioMusica()` – seleciona aleatoriamente uma música do acervo e lista seus dados. Todas as informações sobre o MP3 são listadas, cada atributo listado em uma linha. Se não houver MP3 cadastrado, o método deverá imprimir: "Acervo de MP3 vazio".

7. `selecionaAleatorioMusica(artista)` – seleciona aleatoriamente uma música do acervo de `artista` e lista seus dados. Todas as informações sobre o MP3 são listadas; o conteúdo de cada atributo é listado em uma linha. Se não houver MP3 cadastrado para o artista, o método deverá imprimir: "Acervo de MP3 vazio".

8. `salvaAcervoMP3(nomearq)` – este método salva um objeto `AcervoMP3` em um arquivo cujo nome é `nomearq`. Se o arquivo for salvo com sucesso a saída deverá ser "Arquivo salvo"; caso contrário, "Problemas no salvamento do arquivo".

9. `carregaAcervoMP3(nomearq)` – este método carrega um objeto `AcervoMP3` a partir de um arquivo `nomearq`. Se o arquivo for salvo com sucesso a saída deverá ser "Arquivo carregado"; caso contrário, "Problemas no carregamento do arquivo".

10. `salvaMP3()` – este método salva os MP3 do acervo em um arquivo `MP3.acv`. Se o arquivo for salvo com sucesso a saída deverá ser "Arquivo salvo"; caso contrário, "Problemas no salvamento do arquivo".

11. `carregaMP3()` – este método carrega os MP3 salvos no arquivo `MP3.acv` em um objeto `AcervoMP3`. Se o arquivo for salvo com sucesso a saída deverá ser "Arquivo carregado"; caso contrário, "Problemas no carregamento do arquivo".

12. `alteraMP3(n,nome, artista, localizacao, preço)` – este método altera o `n`-ésimo MP3 salvo em `MP3.acv` com os valores passados por meio dos parâmetros `nome`, `artista`, `localizacao` e `preço`. Note-se que este método altera o arquivo mesmo, e não um estrutura de dados em memória. Se o arquivo for salvo com sucesso a saída deverá ser "Arquivo alterado"; caso contrário, "Problemas na alteracao do arquivo".

13. `toca(nome, artista, localizacao, preço)` – este método é opcional para aqueles que quiserem que o seu EP valha 12 pontos (ao invés de 10). Este método toca o MP3 definido pelos parâmetros.

2 Entrada e saída de dados

A entrada de dados é feita através da entrada padrão, usualmente o teclado (claro que você pode redirecionar um arquivo para a entrada padrão para facilitar o processo de teste do seu programa). Cada um dos métodos descritos acima está associado a um número. Por exemplo, o número que indica ao programa para realizar o método `insereMP3(nome, artista, localizacao, preço)` é o número 1, para realizar `eliminaMP3(nome, artista)` é o 2 e assim por diante conforme o número indicado na descrição do método.

Para passar tarefas ao seu programa você deverá colocar em uma linha o número do método que deverá ser executado e nas linhas seguintes, um em cada linha, os parâmetros para o método. Veja o exemplo de entrada de dados abaixo:

```
1
Nessum dorma
Luciano Pavarotti
c:\meus documentos\minhas músicas\dorma.mp3
1.5
1
Mama Africa
Chico César
c:\meus documentos\minhas músicas\dorma.mp3
0.0
3
5
Luciano Pavao
8
meuarquivo.dat
```

Neste exemplo, note-se que são inseridos dois registros de MP3; depois eles são listados por ordem alfabética pelo nome da música e, em seguida, é solicitada a listagem dos MP3 do artista "Luciano Pavao", e nada é encontrado. Finalmente, os dados são salvos no arquivo "meuarquivo.dat". A saída obtida é mostrada abaixo.

```
Registro inserido
Registro inserido
Mama Africa
Chico César
c:\meus documentos\minhas músicas\dorma.mp3
0.0
Nessum dorma
Luciano Pavarotti
c:\meus documentos\minhas músicas\dorma.mp3
1.5
Acervo de MP3 vazio
Arquivo salvo
```

3 Informações importantes

- O EP é individual. A data máxima de entrega é 2 de junho de 2009 às 24h e a forma de entrega é utilizando o CoL. Deverão ser entregues dois arquivos: mp3-<número USP do aluno>.zip com os arquivos fontes Java; e o arquivo mp3-<número USP do aluno>.jar para execução. Os alunos que

incluírem a funcionalidade de tocar MP3 deverão adicionar um arquivo "readme.txt" informando como utilizá-la.

- O arquivo jar será executado e testado automaticamente, por isso, siga estritamente as entradas e saídas especificadas. Para facilitar a execução automatica, **não utilize acentos na entrada nem na saída de dados**. Além do funcionamento do seu código, será verificada a indentação, a clareza do código, a utilização correta das estruturas de dados e a existência de comentários suficientes para o entendimento do programa.

- Arquivos que não compilam ou que não executam receberão nota ZERO. Trabalhos copiados serão punidos com nota ZERO, tanto para o copiador quanto para o copiado. Este programa é semelhante ao da turma de ACH 2003 do ano passado. Os programas submetidos pelos alunos deste ano serão comparados com programas dos alunos de 2008. Se for detectado que o programa submetido foi copiado de alunos do ano passado, será considerado plágio e o aluno receberá nota ZERO.

- Você deverá tratar as possíveis exceções que poderão vir a ocorrer e utilizar arquivos e coleções na implementação do EP.

4 Dicas

- A leitura de dados pode ser facilmente feita com `Scanner`. Utilize `nextInt()` para ler números e `next()` para ler cadeias de caracteres.

- Para testar seu programa, crie um arquivo de entrada `teste.in` com os dados de teste (você pode inventar dados de teste à vontade) e então redirecione a entrada padrão:

  ```
  java -jar  mp3-<número USP do aluno>.jar < teste.in > teste.out
  ```

  Verifique se o conteúdo do arquivo `teste.out` é o esperado para a entrada dada. Dessa forma não é necessário ficar digitando os dados toda vez que se deseja testar o programa.

- Invente alguns testes espertos para seu programa. Isso tornará mais fácil encontrar possíveis erros. Se seu programa funciona para o exemplo que está neste texto não significa que ele funciona para todos os exemplos possíveis!

- Siga estritamente as regras da entrada e saída de dados! Não inclua linhas com avisos, pedidos para digitação, menus, etc. Assuma que os dados são lidos de forma direta, sem interrupção (como se fosse de um arquivo). Na saída utilize de forma precisa o formato indicado.