

Primeira Prova

Considere um sistema que faz o gerenciamento de músicas através de uma coleção de músicas caracterizadas por um identificador, título, duração (em minutos e segundos), autores (uma única *string*), data e gênero musical (*string*). Uma música pode ser uma canção ou uma música instrumental. Uma canção, além dos atributos da música, contém uma letra, armazenada em um arquivo de tipo texto que deve ser carregado para memória e armazenada como uma *string*. Uma música instrumental contém como atributo adicional: o nome de um arquivo contendo a imagem da partitura. O sistema deve permitir adicionar e remover músicas da coleção de músicas, além de permitir buscar uma música por um título. O sistema deve ser flexível, de forma que diferentes implementações da coleção de músicas possam ser implementadas no futuro, por exemplo, coleções implementadas por *array*, por listas encadeadas ou através de uma coleção do Java.

Implemente (sem escrever os métodos de acesso):

a) As classes *Musica*, *Cancao* e *MusicaInstrumental*. Assuma que as classes *Data* e *Duracao* sejam dadas. **Use apropriadamente os conceitos de herança na sua implementação.** (3.0 pontos).

b) A classe *ColecaoMusicasArray* que herda da classe abstrata *ColecaoMusicas*, através de uma implementação baseada em arrays. (5.0 pontos).

```
public abstract class ColecaoMusicas {  
    private int numMusicas;  
    public ColecaoMusicas(){ numMusicas = 0;}  
    public int getNumMusicas() { return numMusicas;}  
    public abstract void adicionar(Musica musica);  
    public abstract Musica buscar(String titulo);  
    public abstract Musica remover(String titulo);  
    public abstract void imprimir();  
}
```

c) A classe *Letra* de forma que, durante a construção dos seus objetos, os dados do atributo texto sejam obtidos do arquivo *nomeArqLetra* passado ao construtor (2.0).

Observações: no item (b) pode ser usado o tipo genérico `ArrayList<E>`

Retorno	Método
boolean	<code>add(E e)</code> adiciona o elemento específico ao final da lista
void	<code>add(int index, E element)</code> insere o elemento especificado na posição especificada na lista
E	<code>get(int index)</code> retorna o elemento na posição index
E	<code>remove(int index)</code> remove o elemento da posição index
boolean	<code>remove(Object o)</code> remove a primeira ocorrência do objeto caso exista
E	<code>set(int index, E element)</code> substitui o elemento na posição index por element
int	<code>size()</code> retorna o tamanho da lista

Fonte: obtido e adaptado de <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>