

TÉCNICAS DE CONSTRUÇÃO DE PROGRAMAS
INF01120

TRABALHO PRÁTICO - FASE 3

Eduardo Spitzer Fischer	290399
Gabriel Lando	291399
Rodrigo Paranhos Bastos	261162

Neste relatório é definido o projeto de um programa gerador de música a partir de texto. São apresentados os requisitos funcionais bem como não funcionais que se têm para o programa, além de sua interface gráfica.

Requisitos do Sistema

Funcionais:

Campo para entrada do texto gerador do som.

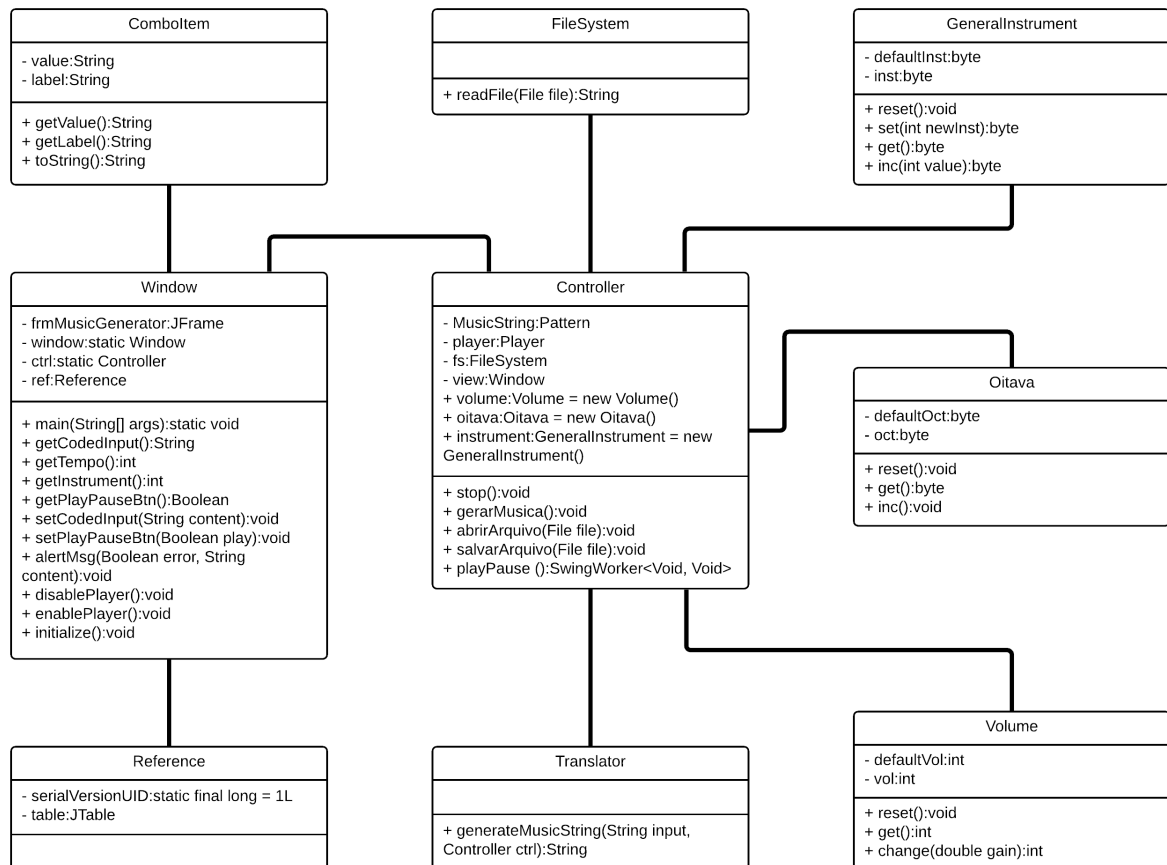
- Campo para upload de arquivo com o texto gerador do som.
 - Ao fazer upload de um arquivo, o seu conteúdo é exibido no campo para entrada de texto, onde pode ser editado.
- Botão “Gerar música”.
 - Ao clicar, o algoritmo de geração é executado.
 - O programa verifica se existem erros.
 - Caso não existam problemas, o áudio é gerado e fica disponível para reprodução.
- Possibilidade de salvar o áudio gerado no formato MIDI para ser exportado e utilizado em outros softwares.
- Botões de play/pause e stop para controle da reprodução.
 - Play: continua a reprodução a partir do último estado;
 - Pause: pausa a reprodução e mantém o cursor no ponto onde parou;
 - Stop: interrompe a reprodução e volta o cursor para o começo do texto;
 - O botão de play/pause é alterado de acordo com o estado atual da música.
- Portabilidade.
 - O programa funcionará em Windows, Mac e Linux.
- Botão “Referência”.
 - Ao ser clicado, abre um popup com uma tabela contendo cada caractere e o efeito correspondente no som gerado (como descrito na especificação do trabalho);

Não-Funcionais:

- Linguagem de programação: Java;
- Será utilizada a biblioteca JFugue (<http://www.jfugue.org/index.html>) para a sintetização dos sons.
- Será utilizada a biblioteca Swing para criação da interface gráfica (GUI).
- Como o programa é simples e provavelmente tem seu valor na esfera recreativa, deve haver um foco na usabilidade.

Projeto de Classes

Diagrama de Relações:



Definições das Classes:

O programa foi implementado com nove classes, descritas a seguir.

Window: A classe Window representa a janela pela qual o usuário interage com o programa. Ela instância todos os elementos da interface gráfica e a classe controladora.

Seus métodos são:

- `getCodedInput()`: retorna o texto digitado pelo usuário na caixa de texto da janela principal.
- `setCodedInput()`: recebe um texto que é colocado na caixa de texto da janela principal. Usado para carregar para edição o texto de arquivos de música abertos.
- `getPlayPauseBtn()`: retorna TRUE se o botão Play/Pause está mostrando o símbolo de *play*, FALSE se mostra *pause*.
- `getInstrument()`: retorna o código do instrumento escolhido pelo usuário na *combo box* da janela principal.
- `getTempo()`: retorna um inteiro representativo do valor de BPM escolhido pelo usuário no *spinner* da janela principal.
- `setPlayPauseBtn(Boolean play)`: muda o símbolo que o botão play/pause mostra.
- `alertMsg()`: exibe mensagem de erro(por exemplo, quando é notificado erro de sintaxe).
- `disablePlayer()`: desabilita o botão de play/pause(a interface do player).
- `enablePlayer()`: habilita o botão de play/pause.
- `Initialize()`: inicializa os elementos gráficos, desenhando a janela.

Controller: realiza a interface entre as demais classes, por exemplo, pedindo mudanças a *Window* com base nos resultados de carregamento de arquivos(tarefa da *fileSystem*) ou do processo de geração da música(executado pela *Translator*).

Possui como atributos:

- **Pattern MusicString:** tipo especial para uma *MusicString* do *JFugue*. Guarda uma sequência de símbolos que representam uma música, em um formato compreendido pelas funções do *JFugue*.
- **Player player:** classe do *JFugue* capaz de reproduzir sons a partir de uma *MusicString*.
- **MainWindow view:** representa a janela principal.
- **Volume volume:** guarda o volume em que a música deve ser reproduzida. Controla os limites mínimo e máximo do mesmo(revertendo seu valor para o mínimo no caso de uma tentativa de configuração menor que esta e para o máximo no caso de tentativa de configuração de valor maior que o mesmo) bem como aplica os ganhos codificados quando tal é solicitado por meio de seus métodos.
- **Oitava oitava:** de maneira semelhante a volume, guarda a Oitava presente. Também impede que sejam configurados valores não permitidos.
- **GeneralInstrument instrument:** corresponde ao instrumento com o qual a música deve ser reproduzida. Realiza os controles para evitar instrumentos inválidos.

E como métodos:

- **stop()** : para a execução da música e volta para seu início.
- **gerarMusica()** : aciona a *Translator* e a *Window* para que uma *MusicString* seja gerada a partir do que foi digitado na caixa de texto.
- **PlayPause()** : pausa ou continua a reprodução da música e pede para que a *Window* faça as alterações correspondentes de *UI*.
- **abrirArquivo():** recebe um arquivo para ser aberto, interage com a classe da janela principal passando para esta o texto do arquivo para que ela o exiba.
- **salvarArquivo():** salva o MIDI correspondente a música codificada. Retorna se houve sucesso ou não.

Translator: converte a *string* fornecida pelo usuário em um formato compatível com as funções de sintetização musical do Jfugue.

Não possui atributos.

Seus métodos são:

- `generateMusicString()` : retorna uma `MusicString` gerada a partir de uma `string` fornecida.

FileSystem: carrega arquivos contendo as músicas geradas e suas codificações.

Não possui atributos. Seu único método é:

- `readFile()` : lê texto de arquivo para ser usado como código de música.

Reference: representa uma janela com uma tabela contendo a referência da codificação usada para as músicas.

Possui apenas um atributo:

- `Jtable table`: a tabela de referência.

ComboItem: representa um item da *combo box* da janela principal, que guarda dois valores, um rótulo que é exibido ao usuário e um código para uso interno que representa seu valor.

Seus atributos são:

- `String value`: código interno.
- `String label`: rótulo exibido ao usuário.

Seus métodos são os getters e setters relacionados:

- `getValue()`: retorna seu valor.
- `getLabel()`: retorna o rótulo.
- `toString()`: a *combo box* da biblioteca Swing utilizada automaticamente chama o método `toString` do que é inserido na mesma, por isso esse método foi criado na `ComboItem` e retorna também seu rótulo para exibição.

Oitava: utilizada para guardar a oitava em que deve ser reproduzida a música.

Atributos:

- byte defaultOct: valor de oitava padrão.
- byte oct: Oitava atual.

Métodos:

- reset(): reseta o valor da oitava para o padrão.
- get(): retorna o valor de oitava guardado.
- inc(): incrementa o valor da oitava. Satura o resultado em valores mínimos e máximos estabelecidos em caso de tentativa de violá-los.

Volume: guarda o volume de reprodução da música.

Atributos:

- int defaultVol: valor de volume padrão.
- int vol: volume utilizado.

Métodos:

- reset(): retorna o volume ao valor padrão.
- get(): retorna o valor atual.
- change(): multiplica o volume por um valor fornecido. Satura o resultado em valores mínimos e máximos estabelecidos em caso de tentativa de violá-los.

GeneralInstrument:

Atributos:

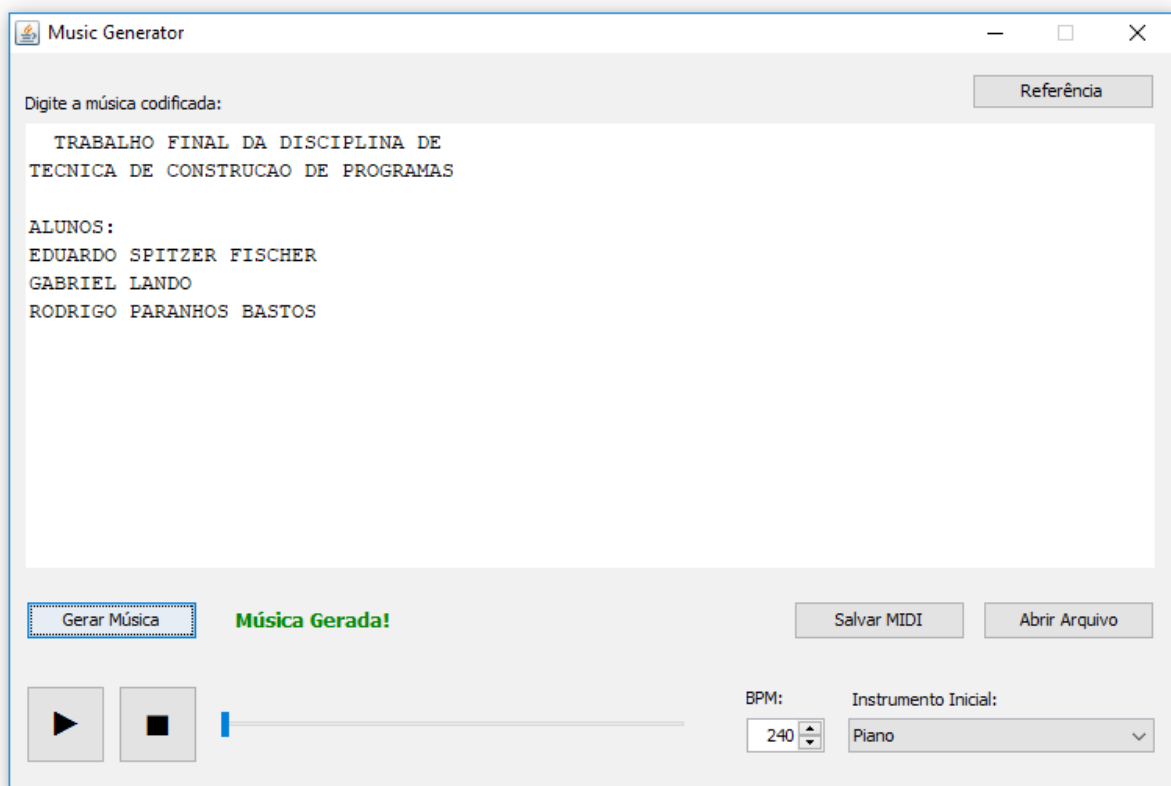
- byte defaultInst: guarda o instrumento padrão.
- byte inst: guarda o instrumento a ser utilizado.

Métodos:

- get(): retorna o instrumento sendo utilizado.
- reset(): configura o instrumento para seu valor padrão.
- set(): configura o instrumento para um fornecido.
- inc(): troca o instrumento pelo próximo na lista de instrumentos.

Interface Gráfica

O programa rodará em uma janela simples, contendo no centro um campo de texto para a entrada da música a ser gerada na forma de código e ao seu redor botões e sliders para sintetizá-la, play/pause, abrir arquivos, avançar/retroceder na música, acessar um guia de referência e selecionar instrumentos e temporização(BPM).



Referência	
Texto	Ação
A	Lá
B	Sí
C	Dó
D	Ré
E	Mi
F	Fá
G	Sol
a,b,c,d,e,f,g	Se caractere anterior era NOTA (A a G), repete nota; Caso contrário, silêncio ou pausa.
Espaço	Aumenta volume para o DOBRO do volume.
"!"	Trocar instrumento para o instrumento General MIDI #7 (Harpsichord).
i, o, u, I, O, U	Aumenta volume em 10%
Outras cons.	Se caractere anterior era NOTA (A a G), repete nota; Caso contrário, silêncio ou pausa.
Dígitos	Trocar instrumento para o instrumento General MIDI (instrumento ATUAL + valor do dígito).
"?" e "."	Aumenta uma oitava. Se não puder, volta a oitava default.
Nova linha	Trocar instrumento para o instrumento General MIDI #15 (Tubular Bells).
","	Trocar instrumento para o instrumento General MIDI #76 (Pan Flute).
","	Trocar instrumento para o instrumento General MIDI #20 (Church Organ).
Outros	Se caractere anterior era NOTA (A a G), repete nota; Caso contrário, silêncio ou pausa.