Compose

Play

Music Library

Rate

Listen

Acknowledgement

We would like to express our special thanks of gratitude to our teacher Ms. Padmavathy as well as our principal Ms. Judith Singh who gave us the golden opportunity to do this wonderful project on the topic (Write the topic name), which also helped us in doing a lot of Research and we came to know about so many new things.

We would also like to thank all the teaching and non-teaching staff of Computer Science department who helped directly or indirectly in the completion of this project.

We would like to thank our music teachers who instilled the interest of music in us and this interest helped us choose this project on music.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

Gopika and Janani

Contents

Introduction

This program converts the keyboard of a computer like keys of a piano. A composer can compose songs and save the song in a song library. A listener shall be able to select a song based on the name, composer or ratings and play the song. After listening to the song, the listener can rate the song.

The program uses an open source library called jfugue to help create music.

Requirements

This section captures the requirements of this project.

1. A composer shall be able to use keyboard of the computer to compose a song.
2. A composer shall be able to choose a tone among Piano, Violin or Flute for the song.
3. A composer shall be able to save the song after composing it.
4. A user shall be able to select a song from a list of songs (library) and play it.
5. A user shall be able to see the list of songs classified according to the Song name, Composer name or Ratings.
6. After listening to the song, the user shall be able to rate the song.

Design

The program has four main classes

1. Menu
2. Composer
3. Song Library
4. Song

All the classes of this project are maintained in a Music package.

The program also uses an open source library jfugue. This library has to be imported into BlueJ environment so that

Menu class

The Menu class accepts the input from the user whether they want to compose a song or they want to listen to the songs.

Composer class

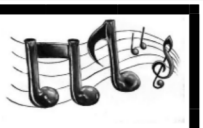
The composer class helps the user to compose a song using the plays of the keyboard (acts as the piano keys) of the computer. The user can also save the song he composed.

Song Library class

This class helps in listing the song which the user saved. The user can choose the song from the list and listen to it.

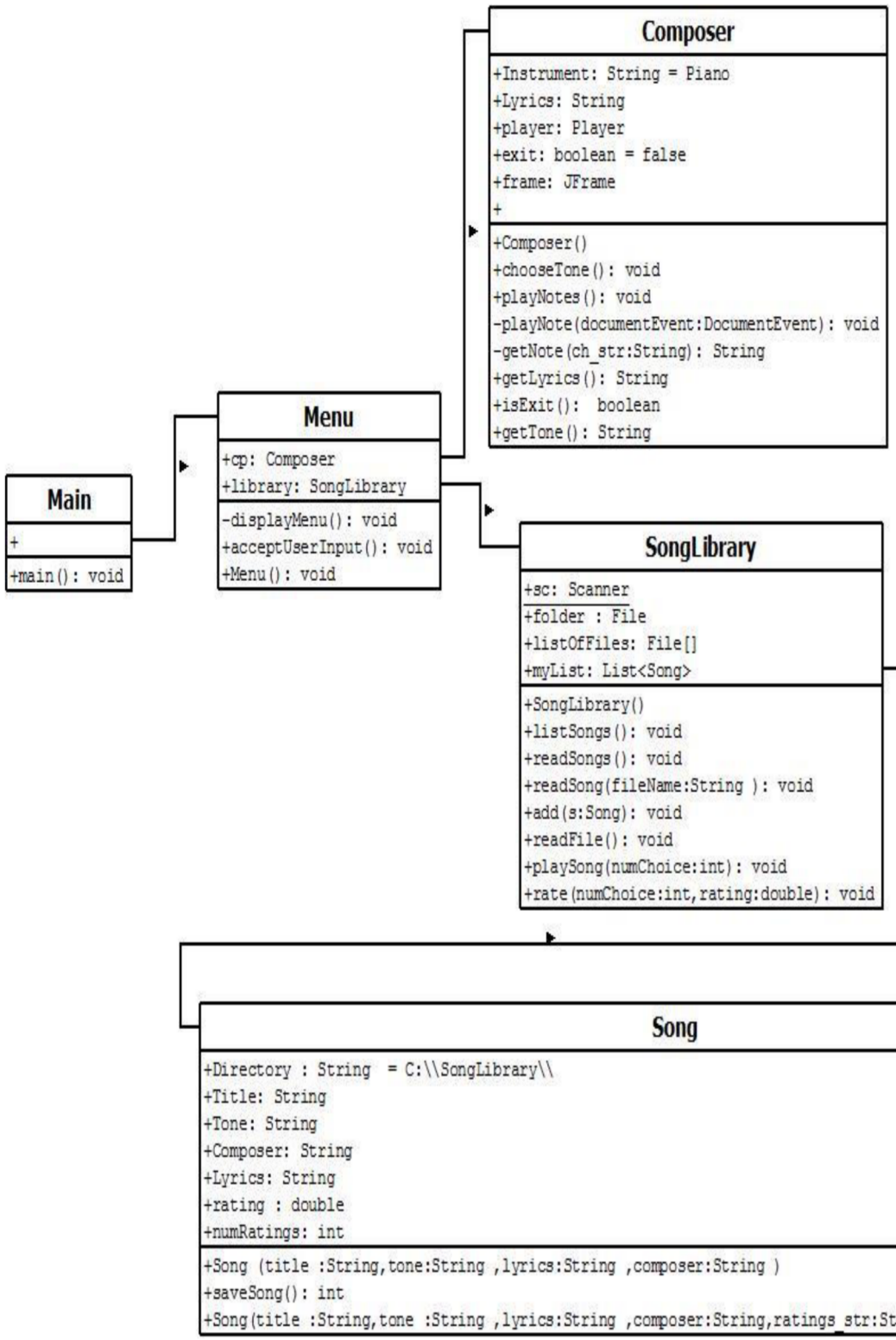
Song class

The function helps in making the object of the song. The user can save the song in to the file.



Jfugue library

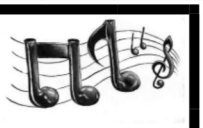
JFugue is an open source programming library that allows one to program music in the Java programming language without the complexities of MIDI. It was first released in 2002 by David Koelle. We are using the version 5.0.9 of jfugue. The jfugue library jar file has to be imported into BlueJ for developing this program.


Composer
+Instrument: String = Piano
+Lyrics: String
+player: Player
+exit: boolean = false
+frame: JFrame
+
+Composer()
+chooseTone(): void
+playNotes(): void
-playNote(documentEvent:DocumentEvent): void
-getNote(ch_str:String): String
+getLyrics(): String
+isExit(): boolean
+getTone(): String
Menu
+cp: Composer
+library: SongLibrary
-displayMenu(): void
+acceptUserInput(): void
+Menu(): void
Main
+
+main(): void
SongLibrary
+sc: Scanner
+folder : File
+listOfFiles: File[]
+myList: List<Song>
+SongLibrary()
+listSongs(): void
+readSongs(): void
+readSong(fileName:String): void
+add(s:Song): void
+readFile(): void
+playSong(numChoice:int): void
+rate(numChoice:int,rating:double): void
Song
+Directory : String = C:\\SongLibrary\\
+Title: String
+Tone: String
+Composer: String
+Lyrics: String
+rating : double
+numRatings: int
+Song (title :String,tone:String ,lyrics:String ,composer:String)
+saveSong(): int
+Song(title :String,tone :String ,lyrics:String ,composer:String,ratings_str:String,numRatings_str:String)

Source code

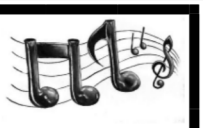
Main Class

```
package Music;
import java.io.*;
public class Main
{
	public static void main() throws IOException
	{
		Menu menu = new Menu();
		menu.acceptUserInput();
	}
}
```

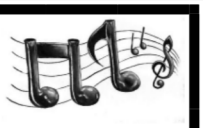



Song Class

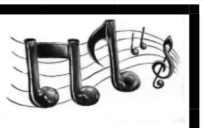
```
package Music;
import java.util .*;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

class Song
{
   String Title="";
   String Tone="";
   String Composer="";
   String Lyrics="";
   double rating;
   int numRatings;
   private String Directory = "/home/higopika/SongLibrary/";
   //private String Directory = "C:\\SongLibrary\\";

   public Song(String title, String tone, String lyrics, String composer)
   {
      Title = title;
      Tone = tone;
      Lyrics = lyrics;
      Composer = composer;
      File file = new File(Directory);
      if (!file.exists()) {
         if (file.mkdir()) {
            System.out.println("SongLibrary is created!");
```

```
            } else {
                System.out.println("SongLibrary Failed to create
directory!");
            }
        }
    }
    public Song(String title, String tone, String lyrics, String composer,
String ratings_str, String numRatings_str)
    {
        Title = title;
        Tone = tone;
        Lyrics = lyrics;
        Composer = composer;
        try{
            rating = Double.valueOf(ratings_str);
        } catch (NullPointerException nullpointerexception){
            rating = 0.0;
        }

        try{
        numRatings = Integer.valueOf(ratings_str);
    }catch (Exception e){
        numRatings = 0;
    }
        File file = new File(Directory);
        if (!file.exists()) {
            if (file.mkdir()) {
                System.out.println("SongLibrary is created!");
            } else {
```

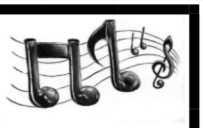



```
                System.out.println("SongLibrary Failed to create
directory!");
            }
        }
    }

    public int saveSong()
    {
        try {
            String filename = Directory + Title;
            File file = new File(filename);
            FileWriter fileWriter = new FileWriter(file);
           fileWriter.write("name:"+Title+"\n");
            fileWriter.write("composer:"+Composer+"\n");
            fileWriter.write("tone:"+Tone+"\n");
            fileWriter.write("lyrics:"+Lyrics+"\n");
            fileWriter.write("ratings:"+rating+"\n");
            fileWriter.write("numRatings:"+numRatings+"\n");
            fileWriter.flush();
            fileWriter.close();
        } catch (IOException e) {
            e.printStackTrace();
            return -1;
        }
        return 0;
    }
}
```

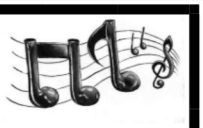



Menu Class

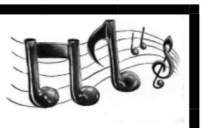
```
package Music;
import java .io.*;
import java .util.*;
public class Menu
{
   Composer cp;
   SongLibrary library;
   Scanner sc= new Scanner(System.in);
   public Menu() throws IOException
   {
      cp = new Composer();
      library =new SongLibrary();
   }

   private void displayMenu()
         System.out.println("Akipog Arts Music Library :-)");
      System.out.println("_____");
       System.out.println("* Press 1 to compose a song");
      System.out.println("* Press 2 to listen to the songs in the music
      library");
      System.out.println("* Press 3 to Exit");
      System.out.println("_____");

 public void acceptUserInput()throws IOException
   {
      while (true)
    {
```

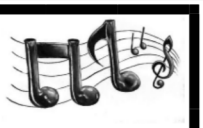
```
        displayMenu();
        BufferedReader br=new BufferedReader(new
InputStreamReader (System.in));
        int ch=Integer.parseInt(br.readLine());
        switch(ch)
        {
           case 1:
           {
       cp.chooseTone();
      cp.playNotes();
             while (!cp.isExit())
            {
                try{
                   java.lang.Thread.sleep(100);
                }catch (InterruptedException interruptedException){
                   System.out.println("Caught InterruptedException");
                }
            }
            String name="";
            String composer="";
            String fileName = "";
            System.out.println("Do you want to save you song??
enter 'y' for yes and 'n' for no");
            String c= sc.next();
            if(c.equalsIgnoreCase("y"))
            {
               System.out.println("Enter title of the song");
               name =br.readLine();
               System.out.println("Enter name of composer");
```

```
                composer = br.readLine();

                fileName =name+".txt";
                System.out.println(fileName);

                Song s=new Song(name, cp.getTone(),
cp.getLyrics(),composer);
                s.saveSong();
                library.add(s);
              }
              cp.clearLyrics();


              break;
            }
            case 2:
            {
              System.out.println("Do you want to sort it according to
1.composer 2.Title 3.rating");
              int type=sc.nextInt();
              if (type == 1)
                library.sort("Composer");
              else if (type == 2)
                library.sort("Title");
              else if (type == 3)
                library.sort("Rating");
              else
                library.sort("Title");
              library.listSongs();
```

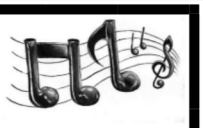
```
                System.out.println("Enter the number of song you want
to listen");
                int choice =sc.nextInt();
                library.playSong(choice);
                System.out.println("Do you want to rate the song??(y or
n)");
                String rate=sc.next();
                if(rate.equalsIgnoreCase("y"))
                {
                   System.out.println("Enter rating");
                   double rating= sc.nextDouble();
                   library.rate(choice, rating);
                }
                break;
            }
            case 3:
            {
                return;
            }

        }
        System.out.println("Came out of the loop");
    }
  }
}
```

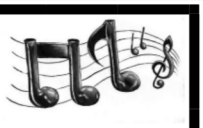
Composer

```
package Music;

import org.jfugue.player.Player;
import org.jfugue.pattern.Pattern;
import java.io.*;
import java.util.*;

import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.event.*;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.text.Document;
import javax.swing.text.BadLocationException;
import java.awt.event.*; //WindowAdapter should be within this...
import javax.swing.*;

public class Composer {
   String instrument = "Piano";
   String lyrics = "";
   Player player;
   boolean exit = false;
   JFrame frame;
   JTextField textField;

   public Composer(){
```


```
        System.out.println("Constructing Composer");
        player = new Player();
        frame = new JFrame("Default Example");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        textField = new JTextField();
        frame.add(textField, BorderLayout.NORTH);

        DocumentListener documentListener = new DocumentListener()
{
            public void changedUpdate(DocumentEvent
documentEvent)
            {
               playNote(documentEvent);

               //System.out.println("Changed");
            }

            public void insertUpdate(DocumentEvent documentEvent)
            {
               playNote(documentEvent);
               //System.out.println("Insert");
            }
            public void removeUpdate(DocumentEvent
documentEvent)
            {
               ///playNote(documentEvent);
               //sSystem.out.println("Remove");
            }
```

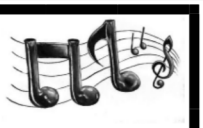



```
        };

textField.getDocument().addDocumentListener(documentListener);

        frame.setSize(250, 150);
    }

    public void chooseTone()throws IOException
    {
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
        System.out.println("Enter the tone 1 - grand piano 2 - flute 3 -
violin");
        int tone=Integer.parseInt(br.readLine());
System.out.println("Instructions_____"
);
        System.out.println("The keys a-C s-D d-E f-F g-G h-A j-B k-C");
        System.out.println("The keys w-C# e-D# t-F# y-G# u-A#");
        System.out.println("Press 'x' to exit");
        System.out.println("--------------------------------------");
        switch(tone)
        {
            case 1:
            instrument = "Piano";
            break;
            case 2:
            instrument = "Flute";
            break;
            case 3:
```

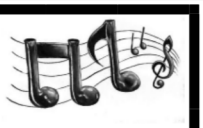



```
            instrument = "Violin";
            break;
        }
    }
    String getTone()
    {
        return instrument;
    }
    public void playNotes()  {
        exit = false;
        frame.setVisible(true);
    }
    private void playNote(DocumentEvent documentEvent)
    {
        DocumentEvent.EventType type = documentEvent.getType();
        Document source = documentEvent.getDocument();
        int length = source.getLength();
        String ch = "";
        try {
            ch = source.getText(length-1, 1);
        } catch (BadLocationException badLocationException) {
            System.out.println("Contents: Unknown");
            return;
        }
        String note = getNote(ch);
        if(exit == true)
        {
            frame.setVisible(false);
        }
```

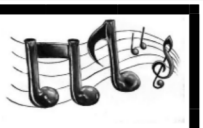

```
    else if (note != "")
    {
       player.play("I["+instrument + "] " +note);
    }
}
public String getNote(String ch_str)
{
    char ch = ch_str.charAt(0);
    switch(ch)
    {
       case 'a':
       {  lyrics += "c ";return("c"); }
       case 'w':
       {  lyrics += "c# ";return("c#"); }
       case 's':
       {  lyrics += "d ";return("d"); }
       case 'e':
       {  lyrics += "d# ";return("d#"); }
       case 'd':
       {  lyrics += "e ";return("e"); }
       case 'f':
       {  lyrics += "f ";return("f"); }
       case 't':
       {  lyrics += "f# ";return("f#"); }
       case 'g':
       {  lyrics += "g ";return("g"); }
       case 'y':
       {  lyrics += "g# ";return("g#"); }
       case 'h':
```

```
            {  lyrics += "a ";return("a"); }
            case 'u':
            {  lyrics += "a# ";return("a#"); }
            case 'j':
            {  lyrics += "b ";return("b"); }
            case 'k':
            {  lyrics += "c6 ";return("c6"); }
            case ' ':
            {  lyrics += "Rq ";return("Rq"); }
            case 'x':
            {  exit = true;
               return("exit"); }
         }
         return "";
      }
      public String getLyrics()
      {
         return lyrics;
      }

      public void clearLyrics()
      {
         lyrics = "";
         try{

textField.getDocument().remove(0,textField.getDocument().getLeng
th());
         } catch (Exception e) {
            System.out.println("Clearing text failed");
```

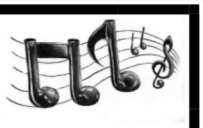



```
        }
    }
    public boolean isExit()
    {
        return exit;
    }
}
```

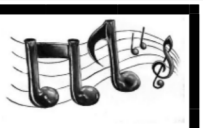



Song Library

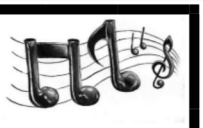
```
package Music;
import java.lang.*;
import java.util.*;
import java.io.*;
import java.io.RandomAccessFile;
import java.io.FileReader;
import java.io.File;
import java.util.Iterator;
import org.jfugue.pattern.Pattern;
import org.jfugue.player.Player;

class SongLibrary
{
   Scanner sc;
   File folder;
   String location = "/home/higopika/SongLibrary/";
   //String location = "C:\\SongLibrary\\";
   static File[] listOfFiles;
   String fname="";
   Song s;
   static List<Song> myList = new ArrayList<Song>();

   public SongLibrary() throws IOException
   {

      readSongs();
   }
```

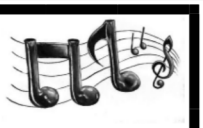
```
    public void listSongs() throws IOException
    {
        //Read the list of songs
        //Print on the screen with serial number, the song name,
composer and ratings
        //
        System.out.println("-----------------------------------------------------
--------------------");
        //System.out.println("List Size " +myList.size());
        System.out.println("Title\tComposer\t\t\tRating");
        System.out.println("-----------------------------------------------------
--------------------");
        for(int i=0;i<=myList.size()-1;i++)
        {
            Song s = myList.get(i);
            System.out.print((i+1)+")");
            System.out.print(s.Title);
            System.out.print("\t"+s.Composer);
            System.out.println("\t \t"+s.rating);
        }

        System.out.println("-----------------------------------------------------
--------------------");
    }

    public void readSongs() throws IOException
    {
        File folder = new File(location);
```

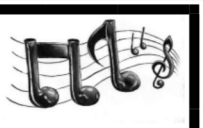
```
        File[] listOfFiles = folder.listFiles();

        for (int i = 0; i < listOfFiles.length; i++) {
           if (listOfFiles[i].isFile()) {
              readSong(listOfFiles[i].getAbsolutePath());
           }
        }
     }
     public void readSong(String fileName) throws IOException
     {
        String thisLine = null;
        FileReader fr = null;
        String title = null, tone = null, lyrics = null, composer = null,
ratings_str = null, numRatings_str = null;

        System.out.println(fileName);
        // open input stream test.txt for reading purpose.
        try{
           fr=new FileReader(fileName);
        } catch (IOException ioException){
           System.out.println("File not present");
        }
        BufferedReader br = new BufferedReader(fr);

        while ((thisLine = br.readLine()) != null) {
           try {
              String name = thisLine.substring(0,thisLine.indexOf(":"));
              if (name.equals("name"))
                 title = thisLine.substring(thisLine.indexOf(":")+1);
```

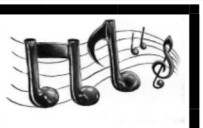
```
            else if (name.equals("composer"))
               composer = thisLine.substring(thisLine.indexOf(":")+1);
            else if (name.equals("tone"))
               tone = thisLine.substring(thisLine.indexOf(":")+1);
            else if (name.equals("lyrics"))
               lyrics = thisLine.substring(thisLine.indexOf(":")+1);
            else if (name.equals("ratings"))
               ratings_str = thisLine.substring(thisLine.indexOf(":")+1);
            else if (name.equals("numRatings"))
               numRatings_str =
thisLine.substring(thisLine.indexOf(":")+1);
         }
         catch(Exception e) {

            e.printStackTrace();
         }

      }
      Song s = new Song(title, tone, lyrics, composer, ratings_str,
numRatings_str);
      myList.add(s);
      System.out.println("********************");
   }

   public void add(Song s)
   {
      myList.add(s);
   }
```

```
    public void readFile()throws IOException
    {
       System.out.println("Enter the number of file you want to play");
       int n=sc.nextInt();

       fname = listOfFiles[n-1].getName();
       System.out.println(fname);
       BufferedReader br = new BufferedReader(new
FileReader(fname));
       String st;
       while((st=br.readLine()) != null){
          System.out.println(st);
       }
    }

    public void playSong(int numChoice)
    {
       Song s1 = myList.get(numChoice-1);
       Pattern p = new Pattern(s1.Lyrics).setInstrument(s1.Tone);
       Player player = new Player();
       player.play(p);
    }

    public void rate(int numChoice, double rating)
    {
       Song s1 = myList.get(numChoice-1);
       double old_rating = s1.rating;
       int num_ratings = s1.numRatings;
```

```
        double new_rating = ((old_rating * num_ratings) +
rating)/(++num_ratings);
        s1.numRatings = num_ratings;
        s1.rating = new_rating;
        s1.saveSong();
    }
    public static void sort(String type)throws IOException
    {
    for (int a=1; a<myList.size(); a++)
        {
         for(int b=0; b<myList.size()-a; b++) {
                boolean swap = false;

           if(type.equals("Composer"))
           {

if(myList.get(b).Composer.compareTo(myList.get(b+1).Composer) >
0)
                swap = true;
           }
           else if(type.equals("Title"))
           {
              if(myList.get(b).Title.compareTo(myList.get(b+1).Title) >
0)
                swap = true;
           }
           else if(type.equals("Rating"))
           {
              if(myList.get(b).rating < myList.get(b+1).rating)
```

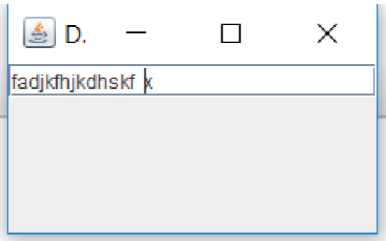
```
                swap = true;
            }

            if (swap){
               //swap movies[b] with movies[b+1]
               Song temp = myList.get(b);
               myList.set(b,myList.get(b+1));
               myList.set(b+1,temp);
            }
         }
      }
   }
}
```

Output

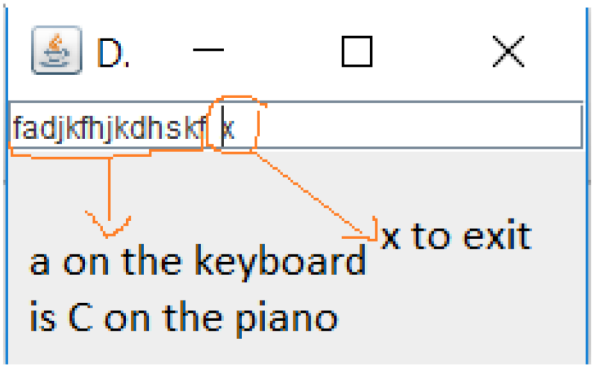
Composing Music

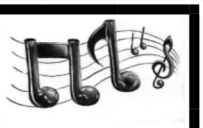
BlueJ: Terminal Window - GopikaProject

Options




```
Construcing Composer
********************
********************
********************
********************
********************
********************
Akipog Arts Music Library :-)
-----------------------------
* Press 1 to compose a song
* Press 2 to listen to the songs in the music library
* Press 3 to Exit
-----------------------------
1
Enter the tone 1 - grand piano 2 - flute 3 - violin
3
Instructions_______________________________
The keys a-C s-D d-E f-F g-G h-A j-B k-C
The keys w-C# e-D# t-F# y-G# u-A#
Press 'o' to save  and Press 'x' to exit
-----------------------------------------
Do you want to save you song?? enter 'y' for yes and 'n' for no
```





Listening to the music from Song Library

BlueJ: Terminal Window - songLibrary

Options

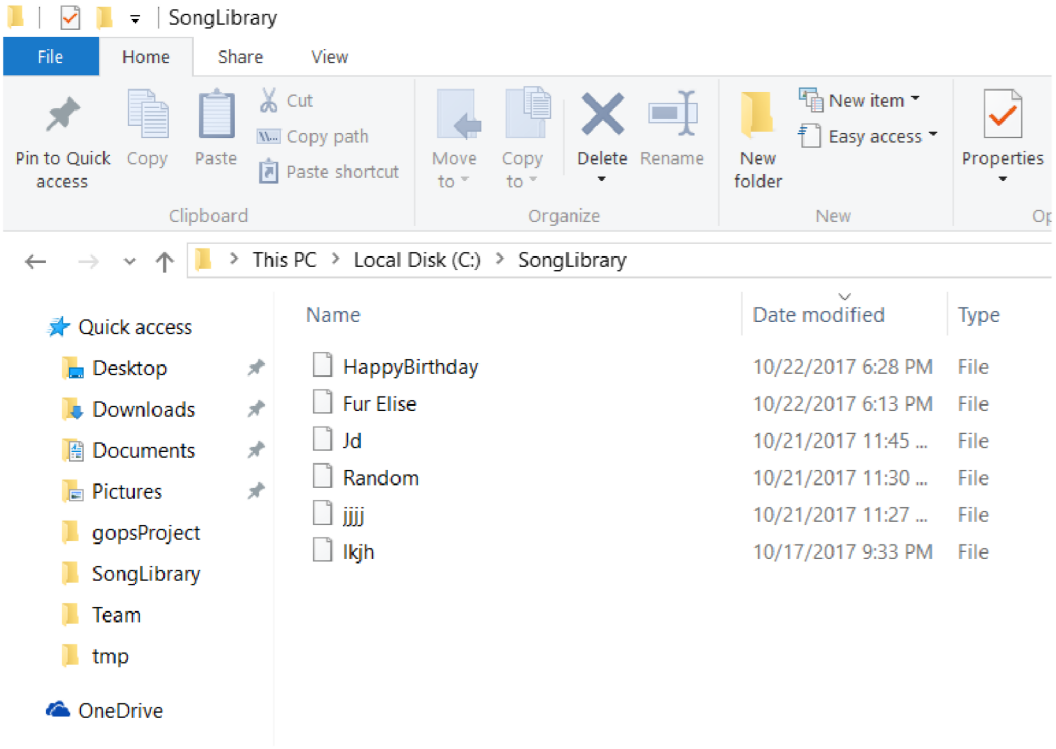
```
Akipog Arts Music Library :-)

_____________________________
* Press 1 to compose a song
* Press 2 to listen to the songs in the music library
* Press 3 to Exit

_____________________________
2
Do you want to sort it according to 1.composer 2.Title 3.rating
3
-------------------------------------------------------------------------
List Size 7
Title   Composer                          Rating
-------------------------------------------------------------------------
1)aa    aaa             5.0
2)FurElise      Ludwig van Beethoven            5.0
3)JanaGanaMana                  4.0
4)Titan jd              4.0
5)ItsASmallWorld                        3.0
6)Jd    JD              2.0
7)ShapeOfYou    anonymous               1.0
-------------------------------------------------------------------------
Enter the number of song you want to listen
6
Do you want to rate the song??(y or n)
y
Enter rating
5.0
```

Song File

The song files are persisted in the disk at the following location

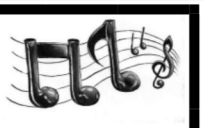
On Windows : C:/SongLibrary

On Linux /home/<user>/SongLibrary



Song File Format

```
name:HappyBirthday
composer:Gopika Jayadev
tone:Piano
lyrics:g g a g c b g g a g d c g g g e c b a f f e c d c
ratings:4.0
numRatings:1
```



Bibliography

https://stackoverflow.com/

https://docs.oracle.com/

http://www.jfugue.org/