MUSOPLAN PLANNING REPORT BY GLOBEX

Authored by Robert Blackshaw

1. Music Management Office & Globex minutes - MUSOPLAN

1. Greetings
2. Current understandings and established requirements of the software.
    a. Command line or GUI
        - ☑ ~~Cmd line~~
        - □ GUI

    b. Enter a musician into the program each with their own data.
        - ☑ ~~Yes~~
        - □ No

    c. Create troupes with the musicians
        - ☑ ~~Yes~~
        - □ No

3. What would you like the program to do?
    a. Inputs
        - ☑ ~~Enter a musician into the program~~
        - ☑ ~~Add a Troupe into the program~~
        - □ Introduce themselves?
        - ☑ ~~Describe a fact about themselves?~~
        - ☑ ~~Add musicians to troupes?~~
        - ☑ ~~Get a summary description?~~
        - ☑ ~~Get a detailed description? (i.e. adding musician descriptions in with the summary description)~~
        - ☑ ~~Calculate costs?~~
        - □ Present lists of troupes and musicians?
        - ☑ ~~Import a list of troupes from a text file~~
        - ☑ ~~Write descriptions to files? (line by line), text file with troupe line by line.~~

4. How the Program will work (i.e. loop)
    a. Explaining how the program will likely work. I.e. looping back to the menu. Putting numbers in and prompts.
        - ☑ ~~Okay~~
        - □ Not okay

    b. Inputs through the keyboard
    c. Input text files

5. Data Structures
    a. What points of information will likely be required for each musician?
        - ☑ ~~Name~~
        - ☑ ~~Instrument~~
        - ☑ ~~Experience~~
        - ☑ ~~Hourly rate~~

    b. What points of information will likely be required for each troupe?
        - ☑ ~~Name~~
        - ☑ ~~Minimum duration~~
        - ☑ ~~Genre~~
        - □ Summary description?
        - □ Detailed description?
        - □ Members?
        - □ Total rate?

    c. Min max characters? 3-30
    d. Hourly rate more than? $50

2. Class Description

a. What data will the class have?

In the MusoPlan program there will be a total of 6 classes. There will be two parent classes for musicians and troupes. The other 4 classes for each instrument, Guitarist, Flautist, Bassist and Percussionist, will be children of the musician class.

The Musician class will consist of 4 objects of data. It will have the musicians name, their instrument, experience and hourly rate.

The Troupe class will have 4 objects of data, 1 array and 4 methods. There will be troupe name, minimum duration, maximum duration and genre. It will have an array that stores musician instances.

The 4 child classes of Guitarist, Flautist, Bassist and Percussionist will use constructor and super functionality to store musicians name, their instrument, experience and hourly rate.

b. What behaviours will the class have

The Musician class will use getters and setters for all 4 of its objects. This allows the use of if statements for validation to be used. This will be required for the Name variable (3-30 characters), the experience variable (positive numbers) and the hourly rate (over 50).

The Troupe class will also use getters and setters for all 4 of its objects. If statements will be used for the same purpose. Validation for troupe name(3-30 characters), Minimum duration (0.5-3) and Maximum duration (0.5-3 and greater than Minimum duration).

The troupe class will have an array that stores musician instances and use **encapsulation** for a method that pushes musician instances into that array. The description method will return a string of information about the troupe. Another method will combine strings of information from each musician in the array using **iteration**. This method may also have some degree of **polymorphism** to attach strings and functions together. Finally there will be a method that returns the total cost of the Musicians in the array.

Each Child class will have identical methods that return a detailed description about each instance using **selection**. The troupe class will use this function, whilst keeping **abstraction** in mind to keep it simple.

c. What classes will the class inherit from

The Guitarist, Flautist, Bassist and Percussionist classes will **inherit** musicians name, their instrument, experience and hourly rate from the musician parents class.

3. Class Relationships

All Classes will have to be exported to the other files to be accessed.

The troupe class will use information in its methods from the other child instances stored in its array. This is called **aggregation**. It will need to be able to call the detailed description method from each child class.

The Musician, Guitarist, Flautist, Bassist and Percussionist classes will not need to be related to the Troupe class.

4. Program inputs

Due to the requirements of Music Management's budget. A graphical user interface is outside the scope of this program. Therefore, all inputs into the program will be keyboard inputs via the command line using prompts and the enter key.

When practical the user will be presented with a list and given options to enter numbers from 1-9 to interact with that list to increase useability.
This will not always be practical however, as the user will have to enter information such as band names and hourly rates.
In conclusion, the user will only be required to type letters and numbers into the program in order to get full use from the program.

For example, the user may select option 1 and be given a list of instruments to choose from, from 1 to 4. Then they will be prompted to enter a name, then asked for an hourly rate and finally experience. The code required to do this uses the concept of **sequence**.

5. Program outputs

Most of the data will be visually displayed to the user. However, the program will have the functionality to print a description of each troupe to a text file. The description will consist of the Troupe name and all information the user entered. As well as a description of each musician in the troupe.