HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY



**PROJECT REPORT**

Object Oriented Language and Theory

**Project Topic: Electronic Piano**

Lecturer: Nguyen Thi Thu Trang

Students: Nguyen Truong Giang – 20194751

Nguyen Ba Hai – 20194753

Vu Thanh Hai - 20194756

Class: 131101

Hà Nội, 6/2022

**Content**

[PREFACE](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.gjdgxs) 4

Assignment of members 4

[PROJECT DEMAND 5](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.30j0zll)

[1. Program Demands 5](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.3znysh7)

[2. Program’s Function 5](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.2et92p0)

[3. Program Design 5](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.tyjcwt)

[PROGRAM’S DESIGN ANALYSIS 6](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.3dy6vkm)

[I. Use case Diagram and Technical Specifications 6](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.1t3h5sf)

[1.Use case Diagram 6](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.4d34og8)

[2.Teachnical Specifications 6](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.2s8eyo1)

[2.1.  Play the Piano 6](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.17dp8vu)

[2.2.  Change volume](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.3rdcrjn) 6

[2.3.  Record and listen musical composition](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.26in1rg) 7

[2.4. Change the music style](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.lnxbz9) 7

[2.5. Help menu](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.1ksv4uv) 7

[2.6. Exit program](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.44sinio) 7

[II. Class Diagram](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.2jxsxqh) 8

[DETAILED DESIGNMENT AND TECHNICAL SOLUTION](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.z337ya) 9

[I. Screen Package 9](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.3j2qqm3)

1. MidiApp Class 9
2. MidiAppController Class 9
3. RecordItemScreen Class 12
4. RecordScreen Class 13

II. Keyboard Package 13

1. Note Class 13
2. LoadKeyboard Class 13
3. LoadPianoKeyboard, LoadViolinKeyboard, 13

LoadTrobomeKeyboard, LoadGuitarKeyboard

III. Record Package

1. Record Class 14
2. RecordList Class 14

IV. GUI Design 15

OOP PRICIPLES ANALYSIS 16

1. Encapsulation 16
2. Abstraction 17
3. Inheritance 17
4. Polymorphism 19

[SUMMARY 2](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg" \l "heading=h.32hioqz)2

[I. Advantage 2](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.1hmsyys)2

[II. Disadvantage](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.41mghml) 22

III. Demo video link 22

[REFERENCE](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.3fwokq0) 23

**PREFACE**

Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.Object Technology is one of the most important technics in information and communication technology. To understand how it works, we can make a simple program like an Electronic Piano program. In this project, we practice what we learned about how OOP knowledge and some other application like libraries, and GUI design.

**Members and Assignment**

Group 3:

Nguyen Truong Giang:

* Design Class Diagram, structure of application
* Design GUI: aboutUs.fxml ,instruction.fxml, piano.fxml
* Write package:

+ Keyboard: LoadGuitarKeyboard, LoadKeyboard, LoadPianoKeyboard, LoadTromboneKeyboard, LoadViolinKeyboard, Note, Playable(interface).

+ screen: MidiApp, MidiAppController.

* Write report, making slide.

Nguyen Ba Hai:

* Design Class Diagram, structure of application
* Write package:

+ record: Record, RecordList.

+ screen: RecordScreen, RecordItemScreen.

Vu Thanh Hai:

* Writing report.
* Making Demo video.
* Test the application.

**PROJECT DEMAND**

1. **Program Demands**

Design a program that provides GUI for the user to virtually play an electronic piano. The program allows users to play piano by interacting with GUI, record and play the musical product, and view more detail and instruction.

1. **Program’s Function**

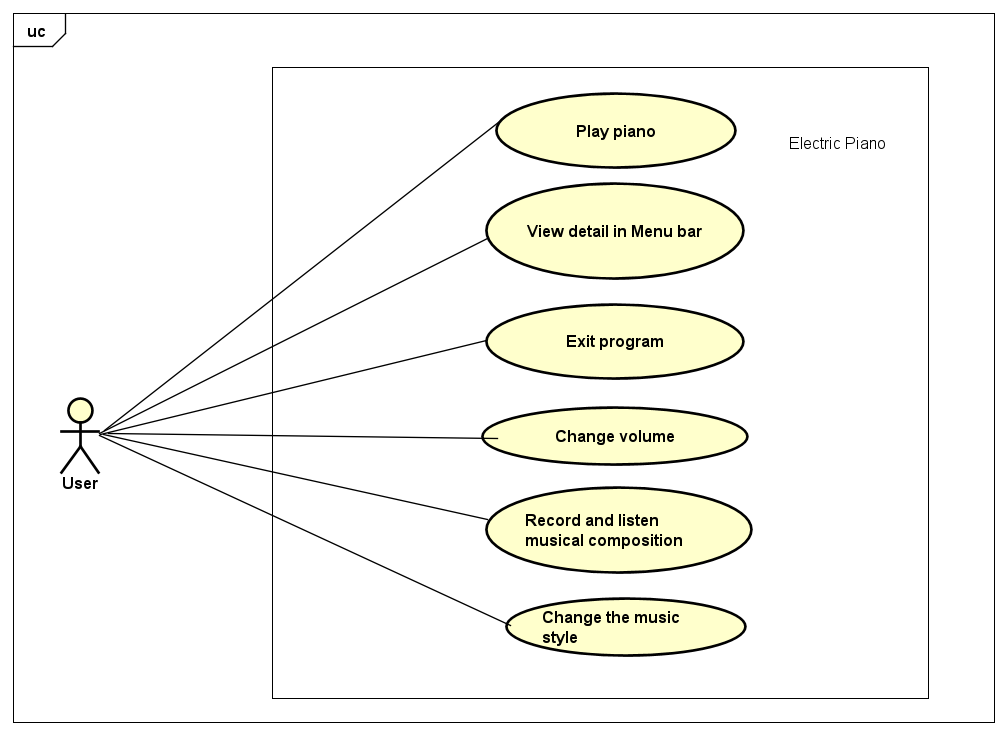
* Play piano on a virtual keyboard.
* Change some other instruments.
* Change the volume of the play.
* Record the plays.
* View instructions and some basic information about our team.

1. **Program Design**
   * Using Java language to build the main program.
   * Using jfugue-5.0.9 library to build and play the sound.
   * Using JavaFx to build GUI of program.
   * Using Java Swing to build Record screen.
   * The program will be organized into packages with different functions.

**PROGRAM’S DESIGN ANALYSIS**

**I. Use case Diagram and Technical Specifications**

**1. Use case Diagram**



**2.Technical Specifications**

**2.1.  Play the Piano**

* Use case name: Play piano
* Mark: The user plays piano by clicking on the button of the note of the virtual keyboard designed.
* Actor: User

**2.2.  Change volume**

* Use case name: Change volume
* Mark: Change the volume of the sound by changing the value of the volume slider.
* Actor: User

**2.3.  Record and listen to musical composition**

* Use case name: Record and listen to musical composition
* Mark: Record the user’s musical composition. The record can be played again to listen to it. The application allows making multiple records.
* Actor: User

**2.4. Change the music style**

* Use case name: Change the music style.
* Mark: Change the music style can be played by the program to other instruments, such as guitar, violin, and trombone.
* Actor: User

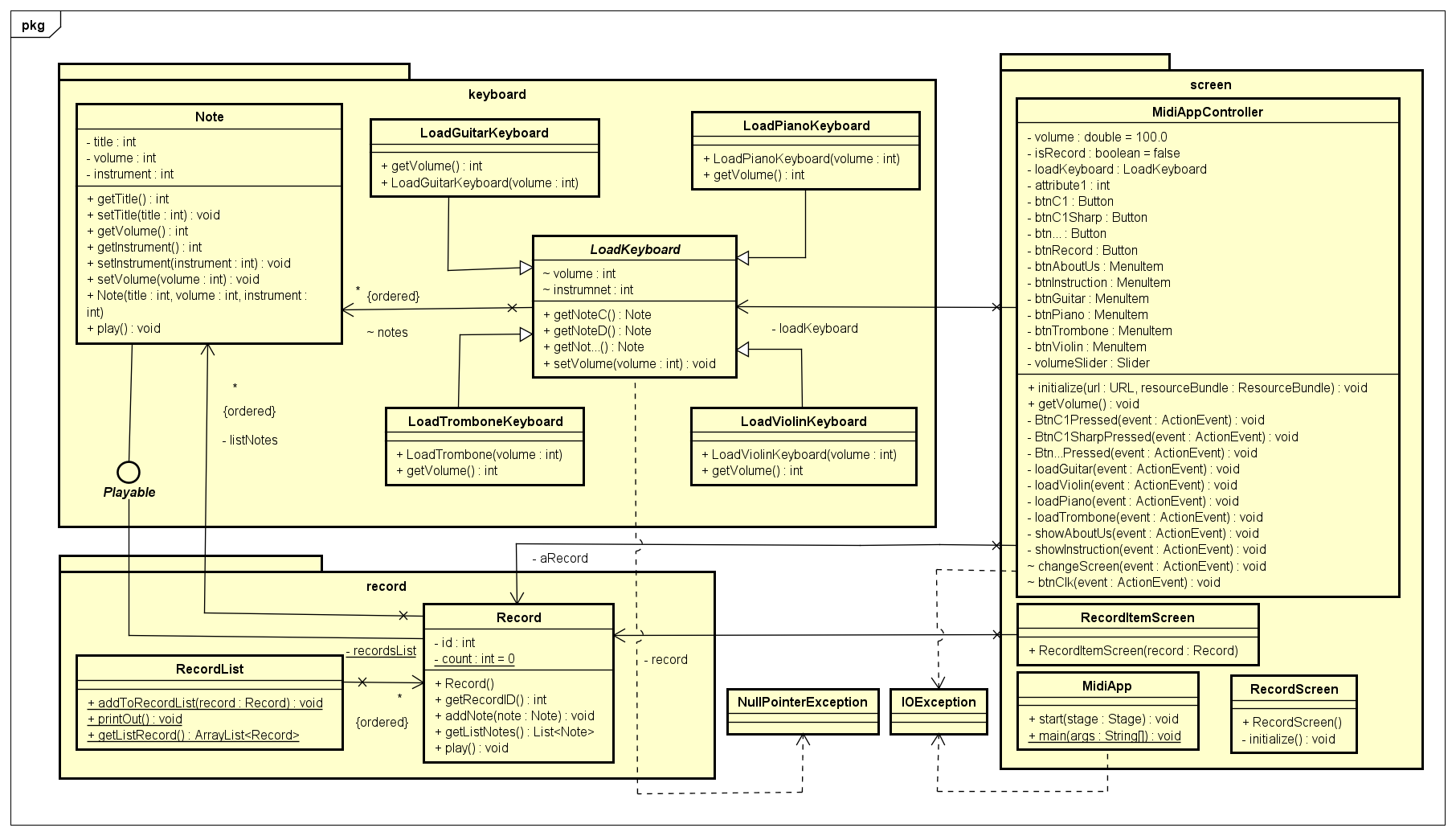
[**2.5. Help menu**](https://docs.google.com/document/d/0ByoYId-KG3HUY0FETUxQVlhodmc/edit?resourcekey=0-Utn6ifbS1oIg87ndRpUxQg#heading=h.1ksv4uv)

* Use case name: View detail in the Menu bar
* Mark: Provide a help menu that shows the basic usage and aim of the program (About us and Instruction) by choosing the function in the help menu.
* Actor: User

**2.6. Exit program**

* Use case name: Exit program
* Mark: Quit the program, with an exit’s confirmation
* Actor: User

**II. Class Diagram**



**DETAILED DESIGNING AND TECHNICAL SOLUTION**

1. **Screen Package**
2. MidiApp Class

Have the main code to launch the program and show GUI. In this class, we throw IOException because it may have an error when the path of the fxml file is not found. To make a confirmation, we add function stage.setOnCloseRequest() and push an alert to it. This Alert let the user confirm close again. If the user chooses “Ok”, the application will shut down.

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. MidiAppController Class
2. Attribute

* volume(double): the value to control the volume of application with the initial value is 100.0
* isRecord (boolean): check if the composition has been recorded, with the initial value being false.
* loadNote(LoadNote): the abstract variable, to load the type of instrument.
* aRecord(Record): If isRecord() is true, the note which players will be recorded.
* Buttons of function in GUI.

1. Operation

* initialize:

Set the value of the volume slider equal to 100% and the set type of instrument.

Ảnh có chứa văn bản

Mô tả được tạo tự động

The range of volume in jfugue 5.0 is 0 to 127, but the range of slide is 0-100% so we need a small convert to set up the value.

* getVolume:

receive the value that which user customed and set it to the application

Ảnh có chứa văn bản

Mô tả được tạo tự động

* Btn…Pressed: Event action when pressing the button on the virtual keyboard. This function, throws a NullPointerException to catch the error null value the function “getNote…()”. In case the note is not set in function loadGuitar, loadViolin, loadPiano, and loadTrombone, the application will send a message to use.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* loadPiano, loadViloin, etc: Load the music style that which user chose

Ảnh có chứa văn bản

Mô tả được tạo tự động

* showAboutUs: show information about our team in the Help menu, we design it in a new window using the fxml file so we throw an IOException like in MidiApp class.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* showInstruction: show information about program instructions. It has an IOException like the howAboutUs function.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* changeScreen: Change to record screen.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* btnClk: change the color of the button to let the user know the application is recording and make a record.

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. RecordScreen Class: make a screen of record using Java Swing.

Include initialize() method to initialize the content of the frame.

Ảnh có chứa văn bản

Mô tả được tạo tự động

1. RecordItemScreen Class: To show each record made.
2. **Keyboard Package**
3. Note Class

a) Attribute

* title(int): the value of exist note in Jfurgue library.
* volume(int): the value of the sound out.
* instrument(int): the value of exist instrument in Jfurgue library.

b) Operation

* getTitle(): return title of a note
* setTitle(int): set title to a note
* getVolume(): return volume of a note
* getInstrument(): return type of instrument of a note
* setInstrument(int): set type of instrument for a note
* setVolume(int): set volume to a note.
* Note(int, int, int): create a new note.
* play(): play the note.

1. LoadKeyboard Class

* Using this abstract class to easy set up the note of each instrument.
* Attribute volume, instrument is the value to change the characteristic of sound
* Using ArrayList<NewNote> notes = new ArrayList<Note> to store the notes of instrument.
* getNote…(): return a specific note if note is not found, the function will return NULL.
* setVolume(int): set volume for all note of the “notes” ArrayList.

1. LoadPianoKeyboard, LoadViolinKeyboard, LoadGuitarKeyboard, LoadTromboneKeyboard Class..
2. Operation
   * getVolume(): return volume of this class.
3. Constructor
   * Load…KeyBoard(int volume): set the sound of identified instrument for each button with the specific volume was set by user.
4. **Record Package**
5. Record Class
6. Attribute

* int id and static int count: to mark the id of a record.
* List<Note> listNotes: to store notes of record.

1. Operation

* Record(): generate record id.
* getRecordID(): return record id.
* addNote(NewNote): add a note to record.
* getListNotes(): return list of notes of a record.
* play(): play the record.

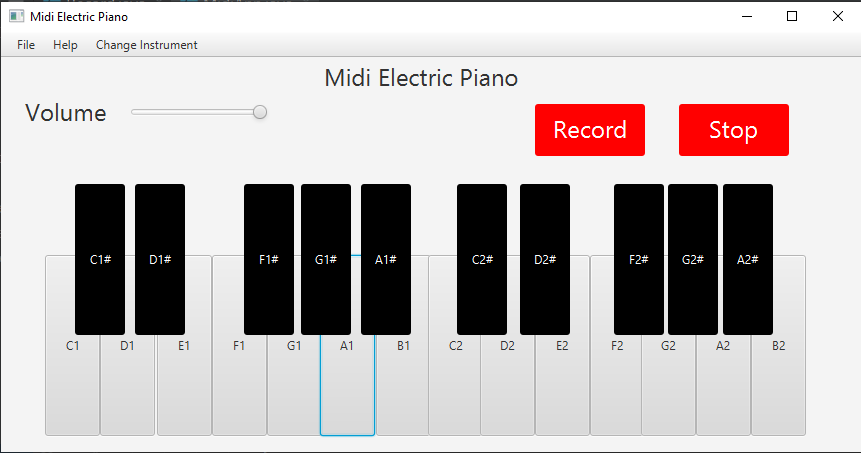
1. RecordList Class
2. Attribute

* ArrayList<Record> recordList: to store a record

1. Operation

* addToRecordList(Record): store a record in a list of records
* printOut(): show list of notes of a record
* getListRecord(): return a list of records

1. **GUI Design**

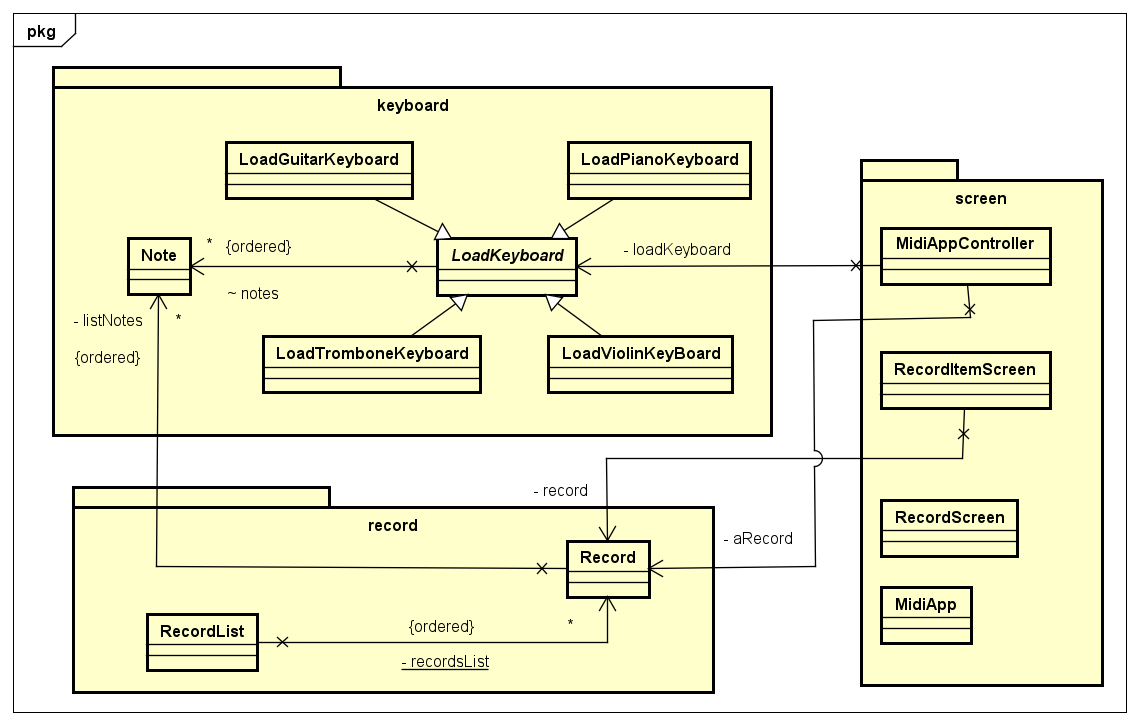


We design a basic short version of Piano with 2 octaves like in the button.

**OOP PRINCIPLES ANALYSIS**

1. **Encapsulation**

The encapsulation is represented in design the program. In this app, we divide the program into 3 main packages: keyboard, screen, and record. Each package has another role and when we update or fix something we will work with specific package.



1. **Abstraction**

In the program, LoadKeyboard is a abstraction class which the function of it is setting the sound for buttons. LoadKeyboard class just follow the list of notes setting for keyboard and get this sound when the button is clicked.

Abstract classes serve as templates for their subclasses(LoadGuitarKeyboard, LoadKeyboard, LoadPianoKeyboard, LoadTromboneKeyboard, LoadViolinKeyboard). Because each instruments have special characteristics so some of button in keyboard can’t not play. We can’t not identify it clearly when listening.

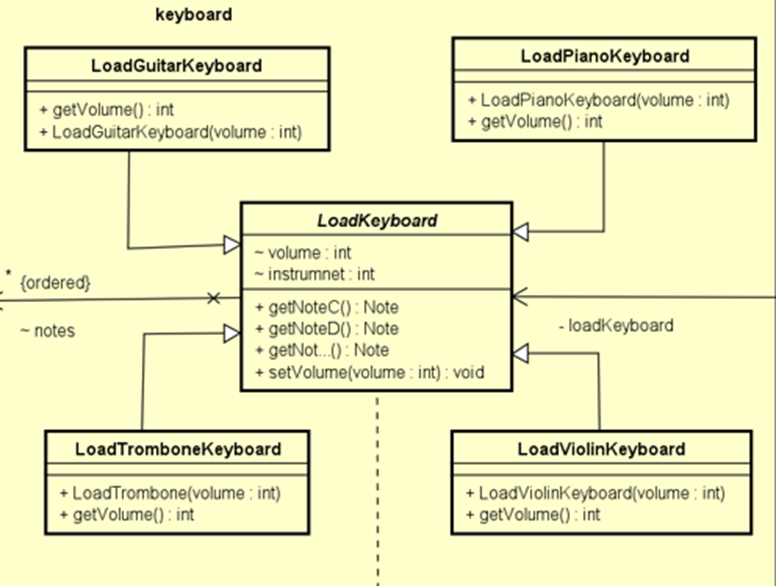
Ảnh có chứa văn bản

Mô tả được tạo tự động

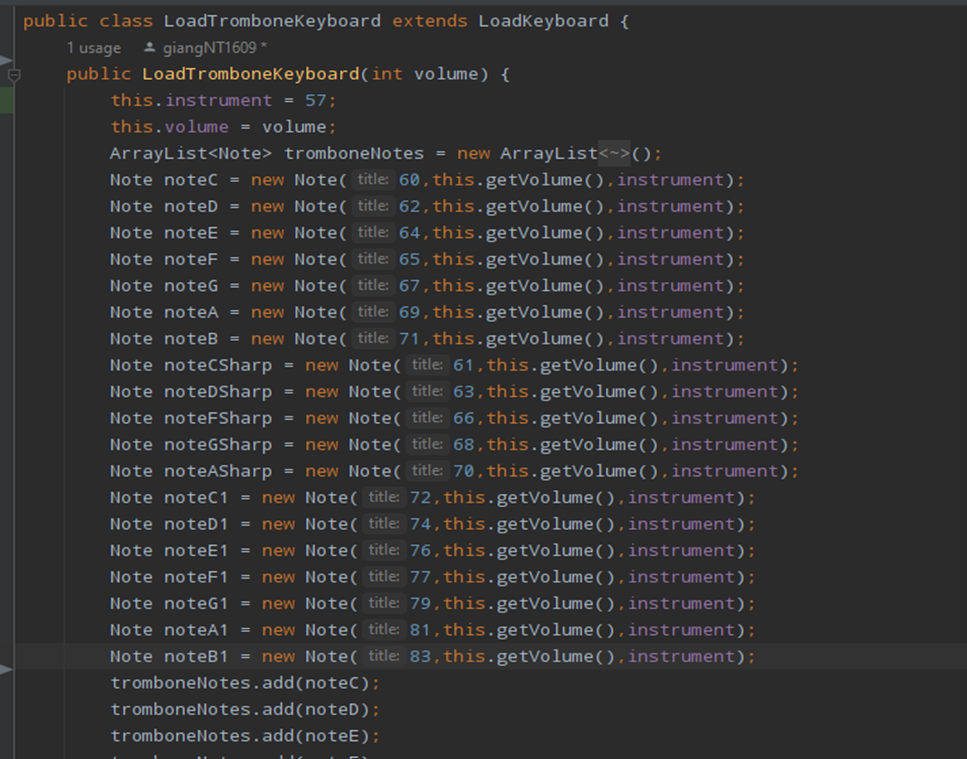
( A part of LoadKeyboard, another notes are similar with 2 function above)

1. **Inheritance**

As mentioned in the abstract principle, LoadKeyboard class have all the details that LoadPianoKeyboard and other similar class must have (the list of notes and volume of each note), so LoadPianoKeyboard, LoadGuitarKeyboard, LoadViolinKeyboard, and LoadTromboneKeyboard classes have inherited LoadKeyboard class.



(Design LoadKeyboard is parent class in application)



( Example of LoadGuitarKeyboard with constructor set up the sound for each button in GUI)

1. **Polymorphism**

The program having polymorphism is shown when we use the interface Playable. And Override the method play() for different object

Ảnh có chứa văn bản

Mô tả được tạo tự động

(Interface Playable)

Ảnh có chứa văn bản

Mô tả được tạo tự động

( Override play method of Note class)

Ảnh có chứa văn bản

Mô tả được tạo tự động

(Override the method play of Record class)

**SUMMARY**

I. Advantage

1. Improve programming skills with Java language.
2. Improve knowledge about Object-oriented Programming.
3. Learn more about JavaFx and Java library.

II. Disadvantage

1. A Music style can be changed not so much, so we just make some instruments for this simple version of this application. I have more specific requirements; we will update this application to get more instruments.
2. The tone of the music is fixed to easily program, so in this version, we cannot change the tone of the sound.
3. There is some delay when playing the first note because the hardware is out of date.

III. Demo link

<https://drive.google.com/drive/folders/1TFkOjVpH8tt_QFoTXQyaMbbIgBfutfoF>

**REFERENCE**

* + 1. **The complete guide to JFugue: Programming Music in Java ( Second Edition Updated for JFugue 5).**
    2. **JavaFx Tutorial For Beginners series on YouTube**

[**https://www.youtube.com/watch?v=9YrmON6nlEw&list=PLS1QulWo1RIaUGP446\_pWLgTZPiFizEMq**](https://www.youtube.com/watch?v=9YrmON6nlEw&list=PLS1QulWo1RIaUGP446_pWLgTZPiFizEMq)