



รายงาน

เรื่อง โครงการพัฒนาแอปพลิเคชัน Student Management

จัดทำโดย

- | | |
|------------------------------|----------|
| 1. นาย ภาณุวัฒน์ ภาวะโคตร | 61070159 |
| 2. นาย วรภัทร ภัทรเมธาวรรกุล | 61070199 |
| 3. นาย วิศรุต พันธุ์ไม้ศรี | 61070211 |
| 4. นาย ศรัณย์ ยันตะบุษย์ | 61070218 |
| 5. นาย ศิริวิทย์ ยอดจิตร | 61070221 |

นำเสนอ

ดร.ธราวิชษฐ์ ธิติจรรย์โรจน์

ดร. สุพัฒน์ดา โชติพันธ์

รายงานนี้เป็นส่วนหนึ่งของวิชา Object-Oriented Programming

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, คณะเทคโนโลยีสารสนเทศ

ปีการศึกษา 2562 ภาคเรียนที่ 1

คำนำ

ผู้จัดทำได้เล็งเห็นถึงความสำคัญในการจัดการกับนักเรียนในเรื่องต่างๆ ไม่ว่าจะเป็นข้อมูลเกรด เป็นต้น และเพื่อการช่วยอาจารย์ในการจัดการนักเรียนได้อย่างง่ายและสะดวกมากขึ้น จึงได้จัดทำโครงงานแอปพลิเคชันที่มีชื่อว่า Student Management เพื่อช่วยให้อาจารย์หรือผู้ใช่ง่ายต่อการจัดการกับข้อมูลนักเรียนต่างๆ ได้สะดวกมากขึ้น ด้วยเหตุนี้ผู้จัดทำจึงได้จัดทำโครงงานแอปพลิเคชัน Student Management เพื่อพัฒนาทักษะและความรู้ในการเขียนโปรแกรมเชิงวัตถุด้วยภาษา Java

ผู้จัดทำหวังว่า รายงานเล่มนี้จะเป็นประโยชน์กับผู้อ่าน หากมีข้อผิดพลาดประการใดผู้จัดทำขอน้อมรับไว้และขออภัยมา ณ ที่นี้ด้วย

ผู้จัดทำ

1. นาย ภาณุวัฒน์ ภาวะโคตร 61070159
2. นาย วรภัทร ภัทรเมธาวรรกุล 61070199
3. นาย วิศรุต พันธุ์ไม้ศรี 61070211
4. นาย ศรัณย์ ยันตะบุษย์ 61070218
5. นาย ศิริวิทย์ ยอดจิตร 61070221

สารบัญ

เรื่อง	หน้า
คำนำ	
บทคัดย่อ	1
วัตถุประสงค์	2
ประโยชน์ที่ได้รับ	3
คลาสไดอะแกรม	4
หน้าที่ของแต่ละคลาสและเมธอด	5
แพ็คเกจ main	5
แพ็คเกจ model	5
แพ็คเกจ controller	6
แพ็คเกจ view	11
แหล่งอ้างอิง	28
หน้าที่ของสมาชิกภายในกลุ่ม	29

บทคัดย่อ

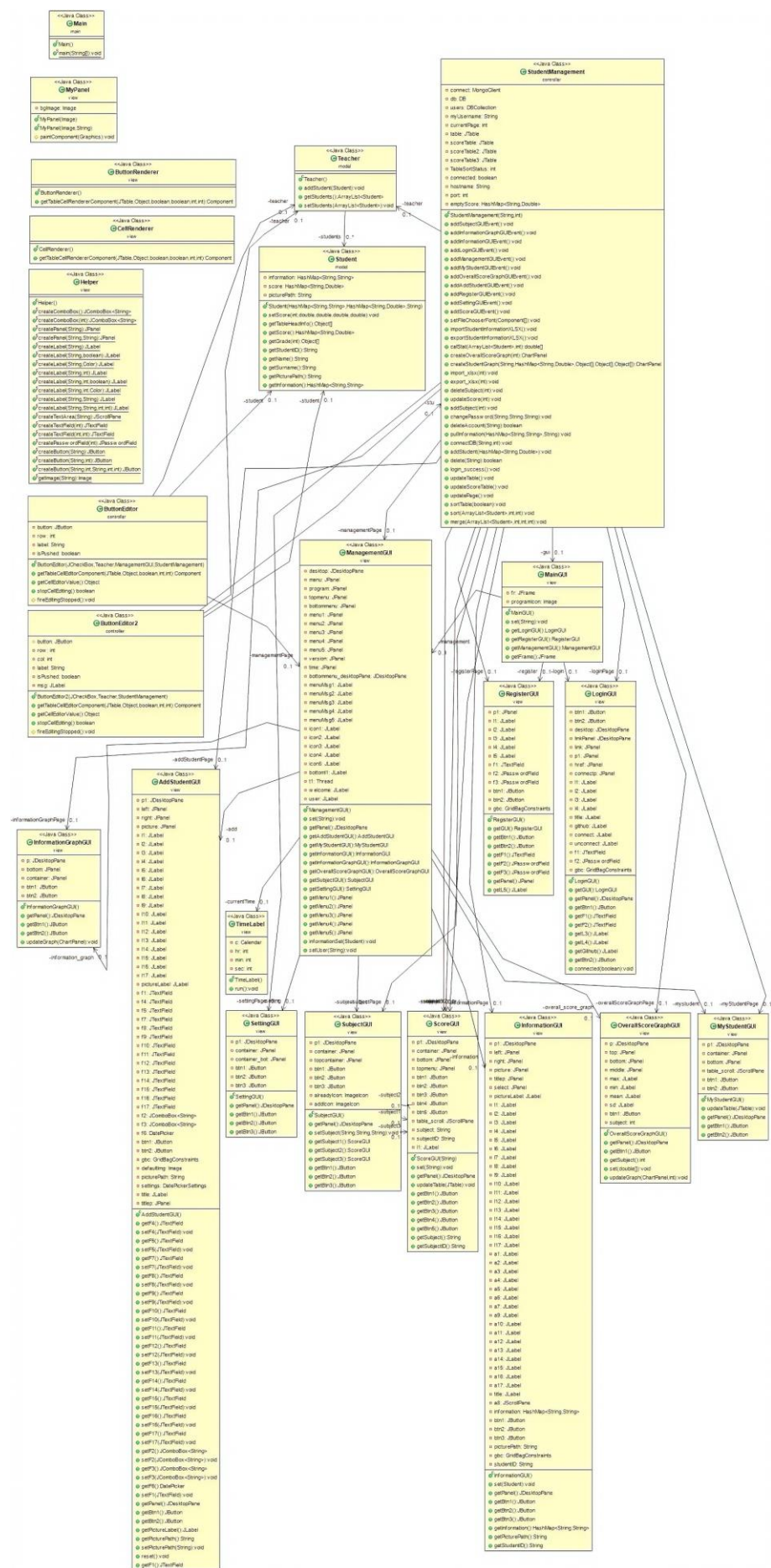
แอปพลิเคชัน Student Management จัดทำขึ้นเพื่อเป็นแอปพลิเคชันอำนวยความสะดวกให้แก่อาจารย์หรือคุณครูที่มีนักเรียนที่ต้องดูแล ซึ่งแอปพลิเคชันสามารถจัดการข้อมูลของนักเรียนหรือนักศึกษาได้ เช่น ชื่อ นามสกุล ที่อยู่ เบอร์ติดต่อ โรคประจำตัว เป็นต้น โดยจะแสดงผลออกมาเป็นตาราง โดยอาจารย์สามารถระบุวิชาที่ตัวเองสอนได้มากที่สุดสามวิชาและสามารถระบุคะแนนของนักเรียนแต่ละคนในแต่ละวิชาได้ โดยทางแอปพลิเคชันจะทำการคำนวณเกรดออกมาให้ตามเกณฑ์มาตรฐาน (ตัด F ที่ 50) และจะแสดงผลออกมาในรูปแบบของตารางและ สามารถดูกราฟคะแนนทั้ง 3 วิชาของนักเรียนแต่ละคนได้ รวมถึงยังสามารถดูกราฟที่จะแสดงว่าในแต่ละวิชามีนักเรียนได้เกรดต่างๆกี่คนและสามารถดูคะแนนรวมที่มากที่สุด คะแนนรวมที่น้อยที่สุด ค่าเฉลี่ย และส่วนเบี่ยงเบนมาตรฐานของวิชานั้นๆได้ และสามารถดาวน์โหลดข้อมูลนักเรียน คะแนน ออกมาเป็นไฟล์ excel (.xlsx) ได้ และสามารถที่จะอัปโหลดข้อมูลนักเรียนหรือคะแนนที่เป็นไฟล์ excel (.xlsx) ได้เช่นกัน โดยข้อมูลต่างๆจะถูกเก็บลงในฐานข้อมูล MongoDB และหากต้องการที่จะโหลดรูปภาพกราฟต่างๆก็สามารถทำได้เช่นกัน

วัตถุประสงค์

1. เพื่อการพัฒนาทักษะการคิดและการเขียนโปรแกรมเชิงวัตถุด้วยภาษา Java และทักษะการทำงานร่วมกันเป็นกลุ่ม
2. เพื่อสร้างแอปพลิเคชันเพื่อให้ความสะดวกสบายกับอาจารย์หรือคนที่มาใช้งานแอปพลิเคชัน Student Management ในการจัดการกับนักเรียนในเรื่องต่างๆ
3. เพื่อนำเทคโนโลยีที่ใช้ในปัจจุบันเข้ามาช่วยในการแก้ปัญหาหรือสร้างความสะดวกสบายในด้านใดด้านหนึ่งได้

ประโยชน์ที่ได้รับ

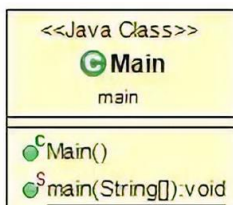
1. อาจารย์หรือผู้ใช้งานมีความสะดวกสบายในการจัดการกับข้อมูลนักเรียนในด้านต่างๆได้
ง่ายมากขึ้น
2. ได้นำความรู้ที่ได้จากการเรียนวิชา Object-Oriented Programming มาใช้งาน รวมถึงได้
พัฒนาทักษะด้านการเขียนโปรแกรมเชิงวัตถุด้วยภาษา Java และทักษะการออกแบบ
หน้าตาของแอปพลิเคชันและฟังก์ชันการทำงานต่างๆ (UI / UX)
3. ได้เรียนรู้การทำงานร่วมกันเป็นกลุ่ม และเรียนรู้การแก้ไขปัญหาหรือข้อผิดพลาดของ
แอปพลิเคชันในตอนที่กำลังพัฒนาแอปพลิเคชันอยู่



หน้าที่ของแต่ละคลาสและเมธอด

แพ็คเกจ main

1. คลาส Main

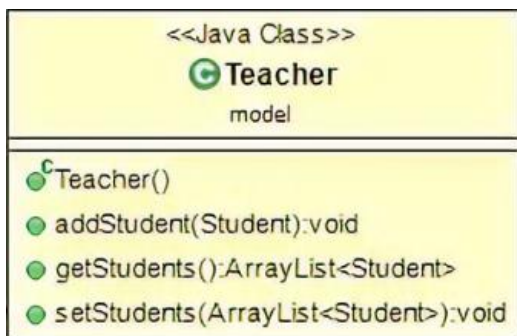


เป็นคลาสที่มีเมธอด main โดยจะทำการตั้งค่า Look and feel ของโปรแกรมรวมถึงโหลดฟอนต์และตั้งค่าฟอนต์ของ Component บางส่วนของโปรแกรม หลังจากนั้นจะทำการสร้างออบเจ็คของคลาส

StudentManagement

แพ็คเกจ model

1. คลาส Teacher

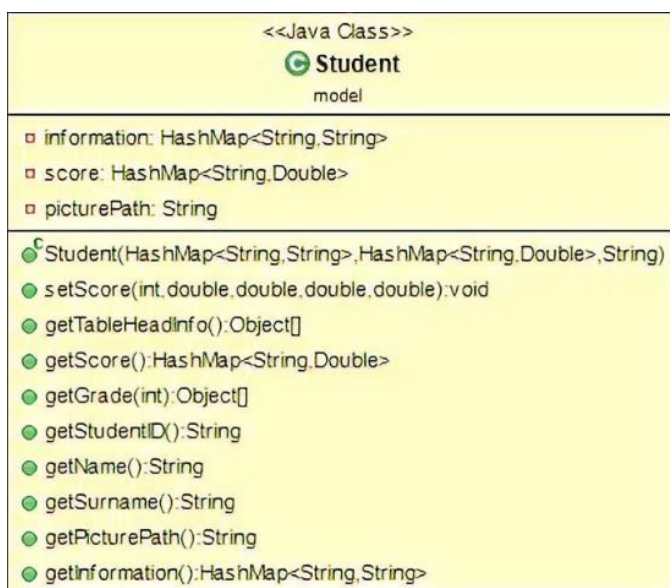


เป็นคลาสข้อมูลของอาจารย์หรือผู้ใช้งานโปรแกรม

เมธอด

1.addStudent(Student) ทำการเพิ่มนักเรียน

2. คลาส Student



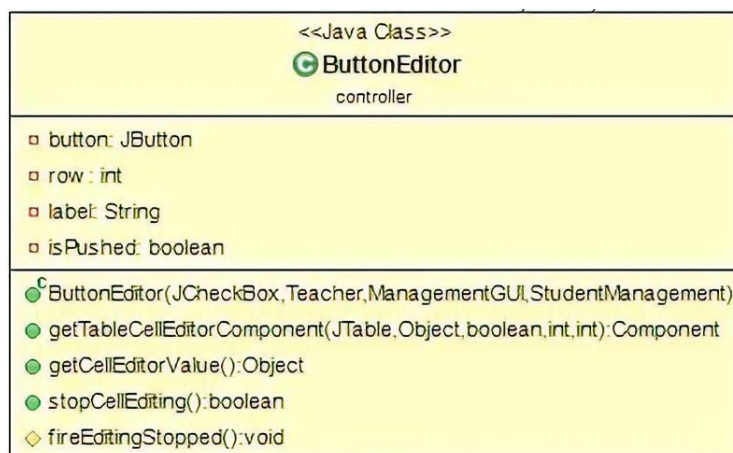
เป็นคลาสข้อมูลของนักเรียนต่างๆ รวมถึงรูปภาพและคะแนนในแต่ละวิชา

เมธอด

1.setScore(int, double, double, double, double) เป็นการกำหนดคะแนน 4 ส่วนในวิชาที่กำหนด

แพ็คเกจ controller

1. คลาส ButtonEditor

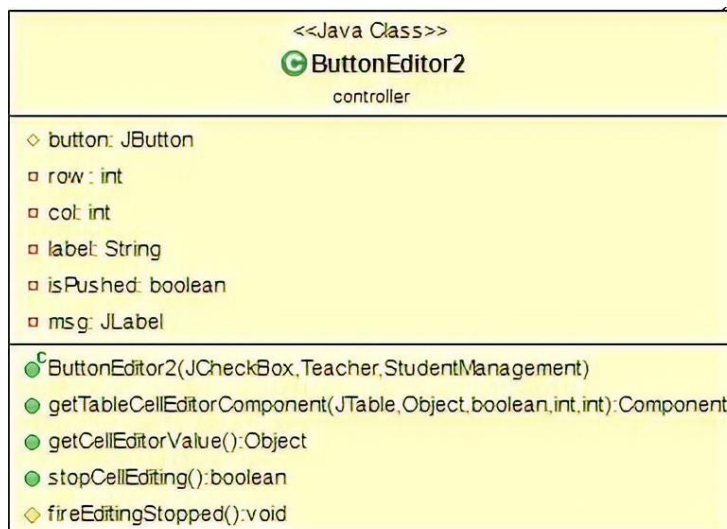


เป็นคลาสที่สืบทอดมาจากคลาส DefaultCellEditor ซึ่งหน้าที่ของคลาสนี้คือการสร้างปุ่มที่สามารถกดได้เพื่อที่จะนำไป render ใส่ใน Cell ของ JTable รวมถึงดักจับว่าให้ทำอะไร

เมธอด

1. ButtonEditor (คอนสตรัคเตอร์) จะทำการเรียกคอนสตรัคเตอร์ของคลาสแม่ผ่าน super และรับออบเจ็กต์ Teacher, ManagementGUI, StudentManagement มาเก็บไว้
2. getTableCellEditorComponent(JTable, Object, boolean, int, int) หลักๆคือจะเป็นการกำหนดข้อความให้กับปุ่มและรับตัวค่าแถวมาเก็บไว้ในตัวแปร row
3. getCellEditorValue() เป็นการเช็คค่าปุ่มถูกกดหรือไม่ผ่านตัวแปร isPushed หากถูกกดก็ให้ทำงานที่ต้องการ
4. stopCellEditing() ทำการกำหนดค่า isPushed ให้เป็น false แล้วเรียกเมธอด stopCellEditing ของคลาสแม่ผ่าน super
5. fireEditingStopped() ทำการเรียกเมธอด fireEditingStopped ของคลาสแม่

2. คลาส ButtonEditor2



เหมือนกับคลาส ButtonEditor ทุกประการยกเว้นการกำหนดว่าให้ทำงานอะไรถ้ากดปุ่ม
นี้

3. คลาส StudentManagement



เป็นคลาสที่เป็นตัวควบคุมการทำงานทั้งหมดของโปรแกรมรวมถึงทำหน้าที่ในการสร้าง
 ออบเจ็คคลาสอื่นๆจาก แพ็คเกจ view และ model รวมถึงติดต่อกับ database

เมธอด

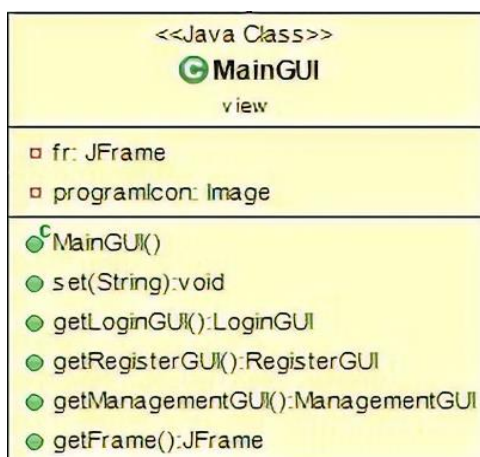
1. StudentManagement (คอนสตรัคเตอร์) จะทำหน้าที่ในการรับค่า host และ port และทำการเรียก เมธอด connectDB เพื่อติดต่อกับ database และกำหนดค่าให้กับตัวแปรในแต่ละออบเจ็คและเรียกเมธอดที่จะทำการตั้งค่า event ทั้งหมดของโปรแกรม
2. addSubjectGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า SubjectGUI
3. addInformationGraphGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า InformationGraphGUI
4. addInformationGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า InformationGUI
5. addLoginGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า LoginGUI
6. addManagementGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า ManagementGUI
7. addMyStudentGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า MyStudentGUI
8. addOverallScoreGraphGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า OverallScoreGraphGUI
9. addAddStudentGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า AddStudentGUI
10. addRegisterGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า RegisterGUI
11. addSettingGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า SettingGUI
12. addScoreGUIEvent() ทำหน้าที่ในการกำหนด event ทั้งหมดของหน้า ScoreGUI
13. setFileChooserFont(Component[]) ทำการกำหนด font ของหน้าต่าง FileChooser
14. importStudentInformationXLSX() ทำการอ่านข้อมูลไฟล์ xlsx และนำมาอัปโหลดนักเรียนเข้าไปในระบบ
15. exportStudentInformationXLSX() ทำการดึงหน้าต่างไฟล์ที่มีข้อมูลของนักเรียนที่มีในระบบให้ผู้ใช้โหลดได้

16. `calStat(ArrayList<Student>, int)` ทำหน้าที่ในการคำนวณค่า max, min, ค่าเฉลี่ย, ค่า SD
17. `createOverallScoreGraph(int)` ทำหน้าที่ในการสร้างกราฟที่จะแสดงว่าในวิชานั้นๆมีนักเรียนได้เกรดในแต่ละเกรดกี่คน
18. `createStudentGraph(String, HashMap<String, Double>, Object[], Object[], Object[])` ทำหน้าที่ในการสร้างกราฟที่จะแสดงว่านักเรียนคนนั้นได้คะแนนในแต่ละวิชาเท่าไรรวมถึงได้เกรดอะไรในวิชานั้น
19. `import_xlsx(int)` ทำการอ่านค่าจากไฟล์ xlsx มาอัปโหลดคะแนนของนักเรียนเข้ามาในระบบ
20. `export_xlsx(int)` ทำการดึงหน้าต่างโหลดไฟล์ที่มีข้อมูลคะแนนของนักเรียนที่มีในระบบให้ผู้ใช้โหลดได้
21. `deleteSubject(int)` ทำหน้าที่ลบวิชาที่มีอยู่
22. `updateScore(int)` ทำการดึงหน้าต่างให้ผู้ใช้สามารถแก้ไขคะแนนของนักเรียนได้โดยเลือกจากรหัสนักศึกษา
23. `addSubject(int)` ทำหน้าที่เพิ่มวิชา
24. `changePassword(String, String, String)` ทำหน้าที่เปลี่ยนรหัสผ่านของรหัสผู้ใช้ที่ล็อกอินอยู่
25. `deleteAccount(String)` ทำหน้าที่ลบรหัสผู้ใช้ที่ล็อกอินอยู่
26. `pullInformation(HashMap<String, String>, String)` ทำหน้าที่ดึงข้อมูลจากในระบบของนักเรียนหนึ่งคนมาแสดงในหน้า `addStudentGUI`
27. `connectDB(String, int)` ทำหน้าที่ติดต่อกับ database
28. `addStudent(HashMap<String, Double>)` ทำหน้าที่ในการเพิ่มนักเรียนหนึ่งคนเข้าไปในระบบ
29. `delete(String)` ทำหน้าที่ในการลบนักเรียนหนึ่งคนออกจากระบบ
30. `login_success()` ทำหน้าที่ในการดึงข้อมูลนักเรียนทั้งหมดจาก database มาเก็บไว้ในโปรแกรม
31. `updateTable()` ทำการอัปเดตการแสดงผลข้อมูลบน ตารางข้อมูลนักเรียน

32. updateScoreTable() ทำการอัปเดตการแสดงผลข้อมูลบน ตารางคะแนนของนักเรียน
33. updatePage() ทำการอัปเดตการแสดงผลของหน้า managementGUI ให้พื้นหลังของเมนูเป็นสีตามต้องการ
34. sortTable(boolean) ทำหน้าที่ในการเรียงนักเรียนบนตารางโดยจะไปเรียกเมธอด sort
35. sort(ArrayList<Student>, int, int) ทำหน้าที่ในการเรียงนักเรียนบนตารางโดยการแตกข้อมูลไปเรื่อยๆและรวมเข้าด้วยกันผ่านเมธอด merge
36. merge(ArrayList<Student>, int, int, int) ทำหน้าที่ในการรวมข้อมูลเข้าด้วยกันและเรียงไปด้วย

แพ็คเกจ view

1. คลาส MainGUI



ทำหน้าที่เป็น JFrame หลักของโปรแกรมโดยจะมีหลายๆ JPanel มาเปลี่ยนหน้าลงบน JFrame อันนี้

เมธอด

1. MainGUI (คอนสตรัคเตอร์) ทำการสร้าง JFrame ที่มีรูปพื้นหลัง
2. set(String) ทำหน้าที่ในการเปลี่ยนหน้าไปเป็นหน้าที่ต้องการ
3. เมธอด Setter และ Getter ของ Attribute ต่างๆ

2. คลาส AddStudentGUI

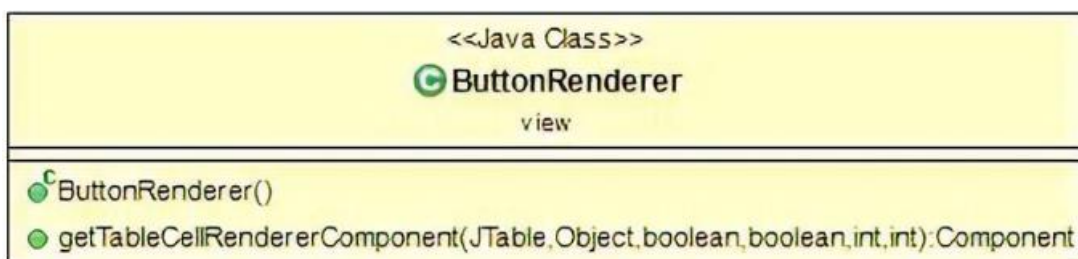
<<Java Class>>	
AddStudentGUI	
view	
□ p1: JDesktopPane	
□ left: JPanel	
□ right: JPanel	
□ picture: JPanel	
□ l1: JLabel	
□ l2: JLabel	
□ l3: JLabel	
□ l4: JLabel	
□ l5: JLabel	
□ l6: JLabel	
□ l7: JLabel	
□ l8: JLabel	
□ l9: JLabel	
□ l10: JLabel	
□ l11: JLabel	
□ l12: JLabel	
□ l13: JLabel	
□ l14: JLabel	
□ l15: JLabel	
□ l16: JLabel	
□ l17: JLabel	
□ pictureLabel: JLabel	
□ f1: JTextField	
□ f4: JTextField	
□ f5: JTextField	
□ f7: JTextField	
□ f8: JTextField	
□ f9: JTextField	
□ f10: JTextField	
□ f11: JTextField	
□ f12: JTextField	
□ f13: JTextField	
□ f14: JTextField	
□ f15: JTextField	
□ f16: JTextField	
□ f17: JTextField	
□ f2: JComboBox<String>	
□ f3: JComboBox<String>	
□ f6: DatePicker	
□ btn1: JButton	
□ btn2: JButton	
□ gbc: GridBagConstraints	
□ defaultImg: Image	
□ picturePath: String	
□ settings: DatePickerSettings	
□ title: JLabel	
□ titleo: JPanel	

● AddStudentGUI()
● getF4():JTextField
● setF4(JTextField):void
● getF5():JTextField
● setF5(JTextField):void
● getF7():JTextField
● setF7(JTextField):void
● getF8():JTextField
● setF8(JTextField):void
● getF9():JTextField
● setF9(JTextField):void
● getF10():JTextField
● setF10(JTextField):void
● getF11():JTextField
● setF11(JTextField):void
● getF12():JTextField
● setF12(JTextField):void
● getF13():JTextField
● setF13(JTextField):void
● getF14():JTextField
● setF14(JTextField):void
● getF15():JTextField
● setF15(JTextField):void
● getF16():JTextField
● setF16(JTextField):void
● getF17():JTextField
● setF17(JTextField):void
● getF2():JComboBox<String>
● setF2(JComboBox<String>):void
● getF3():JComboBox<String>
● setF3(JComboBox<String>):void
● getF6():DatePicker
● setF1(JTextField):void
● getPanel():JDesktopPane
● getBtn1():JButton
● getBtn2():JButton
● getPictureLabel():JLabel
● getPicturePath():String
● setPicturePath(String):void
● reset():void
● getF1():JTextField

เป็นหน้าที่เอาไว้เพิ่มนักเรียนจากข้อมูลที่ใช้กรอกเข้ามา

เมธอด

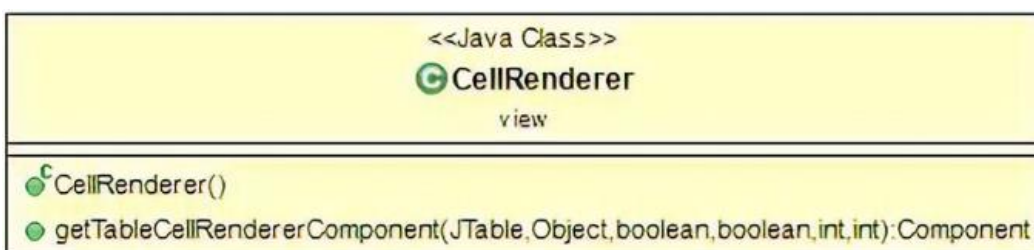
1. AddStudentGUI (คอนสตรัคเตอร์) จะทำการสร้าง JPanel ที่มี Component ต่างๆ มาวางไว้ตามต้องการ
 2. getPanel() จะทำการคืนค่า JPanel หลักของหน้านี้
 3. reset() จะทำการเซ็ต component ต่างๆของหน้านี้ให้เป็นช่องว่าง
 4. เมธอด Setter และ Getter ของ Attribute ต่างๆ
3. คลาส ButtonRenderer



เป็นคลาสที่สืบทอดมาจาก JLabel ที่มีการ implements TableCellRenderer มีหน้าที่ในการกำหนดฟอนต์และสีพื้นหลังให้แก่ cell ของตาราง (จะใช้กับ cell ที่เป็น JButton)

เมธอด

1. ButtonRenderer (คอนสตรัคเตอร์) ทำการกำหนดฟอนต์และทำให้ข้อความอยู่ตรงกลาง
 2. getTableCellRendererComponent(JTable, Object, boolean, boolean, int, int) ทำการกำหนดข้อความและมีการเช็คค่าหากเป็น column ที่กำหนดไว้ให้กำหนดให้มีสีพื้นหลังตามต้องการ
4. คลาส CellRenderer



เป็นคลาสที่สืบทอดมาจากคลาส DefaultTableCellRenderer มีหน้าที่ในการกำหนดฟอนต์และสีพื้นหลังให้แก่ cell ของตาราง

เมธอด

1. CellRenderer (คอนสตรัคเตอร์) ทำการกำหนดฟอนต์และทำให้ข้อความอยู่ตรงกลาง
2. getTableCellRendererComponent(JTable, Object, boolean, boolean, int, int) ทำการกำหนดข้อความและมีการเช็คค่าหากเป็น column ที่กำหนดไว้ให้กำหนดให้มีสีพื้นหลังตามต้องการ

5. คลาส Helper

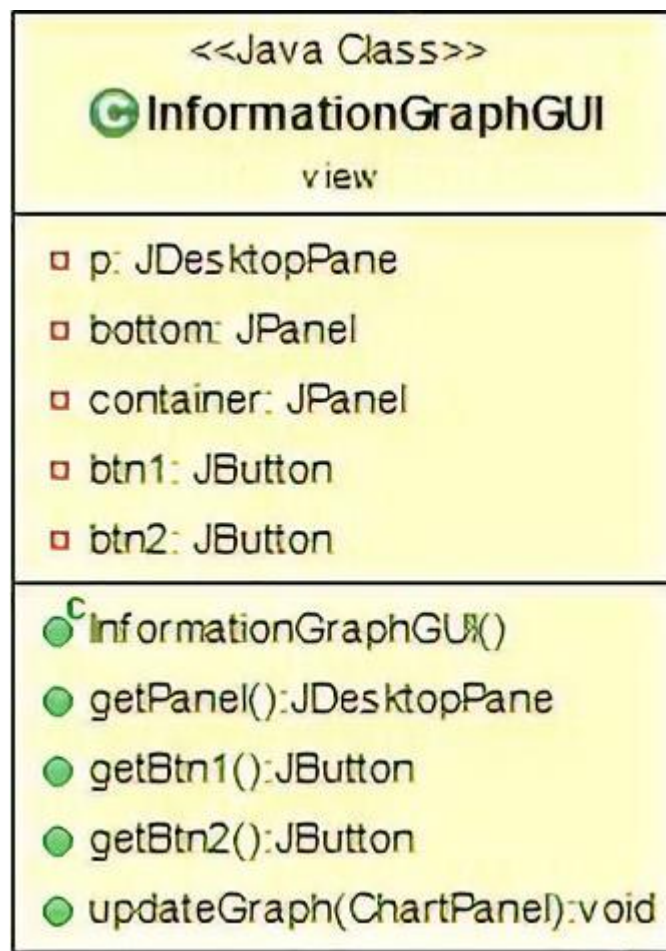


เป็นคลาสที่สร้างขึ้นเพื่อให้ง่ายต่อการสร้าง component ต่างๆ เพื่อจะได้ไม่ต้องกำหนดอะไรเดิมๆทุกครั้งการสร้าง component

เมธอด

1. createComboBox() ทำการสร้างและคืนค่า JComboBox
2. createComboBox(int) ทำการสร้างและคืนค่า JComboBox ที่เป็นขนาดฟอนต์ที่กำหนด
3. createPanel(String) ทำการสร้าง Panel ที่สามารถกำหนด path รูปภาพพื้นหลังได้
4. createPanel(String, String) ทำการสร้าง Panel ที่สามารถกำหนด path รูปภาพพื้นหลังได้รวมถึงสามารถกำหนดได้ว่าจะเป็นขนาดเท่าใด
5. createLabel(String) สร้าง JLabel
6. createLabel(String, boolean) สร้าง JLabel ที่เป็นตัวหนา
7. createLabel(String, Color) สร้าง JLabel ที่มีตัวอักษรสีตามที่ต้องการ
8. createLabel(String, int) สร้าง JLabel ที่มีขนาดตามที่กำหนด
9. createLabel(String, int, boolean) สร้าง JLabel ที่มีขนาดตามที่กำหนด และเป็นตัวหนา
10. createLabel(String, int, Color) สร้าง JLabel ที่มีขนาดและสีตามที่กำหนด
11. createLabel(String, String) สร้าง JLabel ที่มีรูปขนาด 70 * 70
12. createLabel(String, String, int, int) สร้าง JLabel ที่มีรูปขนาดกำหนดเองได้
13. createTextArea() สร้าง TextArea
14. createTextField(int) สร้าง TextField ขนาดตามต้องการ
15. createTextField(int, int) สร้าง TextField ขนาดตามต้องการ และ ขนาดของฟอนต์ตามต้องการ
16. createPasswordField(int) สร้าง PasswordField ที่มีขนาดตามต้องการ
17. createButton(String) สร้างปุ่มที่มีข้อความตามต้องการ
18. createButton(String, int) สร้างปุ่มที่มีข้อความและขนาดตามต้องการ
19. createButton(String, int, String, int, int) สร้างปุ่มที่มีขนาดตามต้องการและมีภาพพื้นหลังที่มีขนาดตามต้องการ
20. getImage(String) คืนค่า Image ที่ไปโหลดมาจากตำแหน่งที่กำหนด

6. คลาส InformationGraphGUI



เป็นหน้าที่แสดงผลข้อมูลคะแนนเป็นกราฟของนักเรียนหนึ่งคน

เมธอด

1. InformationGraphGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
3. updateGraph(CharPanel) ทำการนำ CharPanel ที่รับเข้ามาใส่ลงบน JPanel หลักเพื่อแสดงผล
4. เมธอด Setter และ Getter ของ Attribute ต่างๆ

7. คลาส InformationGUI

<<Java Class>>	
InformationGUI	
view	
□ p1: JDesktopPane	
□ left: JPanel	
□ right: JPanel	
□ picture: JPanel	
□ titlep: JPanel	
□ select: JPanel	
□ pictureLabel: JLabel	
□ l1: JLabel	
□ l2: JLabel	
□ l3: JLabel	
□ l4: JLabel	
□ l5: JLabel	
□ l6: JLabel	
□ l7: JLabel	
□ l8: JLabel	
□ l9: JLabel	
□ l10: JLabel	
□ l11: JLabel	
□ l12: JLabel	
□ l13: JLabel	
□ l14: JLabel	
□ l15: JLabel	
□ l16: JLabel	
□ l17: JLabel	
□ a1: JLabel	
□ a2: JLabel	
□ a3: JLabel	
□ a4: JLabel	
□ a5: JLabel	
□ a6: JLabel	
□ a7: JLabel	
□ a9: JLabel	
□ a10: JLabel	
□ a11: JLabel	
□ a12: JLabel	
□ a13: JLabel	
□ a14: JLabel	
□ a15: JLabel	
□ a16: JLabel	
□ a17: JLabel	
□ title: JLabel	
□ a8: JScrollPane	
□ information: HashMap<String,String>	
□ btn1: JButton	
□ btn2: JButton	
□ btn3: JButton	
□ picturePath: String	
□ gbc: GridBagConstraints	
□ studentID: String	

● InformationGUI()
● set(Student):void
● getPanel():JDesktopPane
● getBtn1():JButton
● getBtn2():JButton
● getBtn3():JButton
● getInformation():HashMap<String,String>
● getPicturePath():String
● getStudentID():String

เป็นหน้าที่แสดงข้อมูลทั้งหมดของนักเรียนหนึ่งคน

เมธอด

1. InformationGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
 2. set(Student) ทำการกำหนดค่าในแต่ละ component จากการไปดึงข้อมูลจากนักเรียน (Student) ที่รับเข้ามา
 3. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
 4. เมธอด Setter และ Getter ของ Attribute ต่างๆ
8. คลาส LoginGUI

<<Java Class>>

 **LoginGUI**

view

- btn1: JButton
- btn2: JButton
- desktop: JDesktopPane
- linkPanel: JDesktopPane
- link: JPanel
- p1: JPanel
- href: JPanel
- connectp: JPanel
- l1: JLabel
- l2: JLabel
- l3: JLabel
- l4: JLabel
- title: JLabel
- github: JLabel
- connect: JLabel
- unconnect: JLabel
- f1: JTextField
- f2: JPasswordField
- gbc: GridBagConstraints

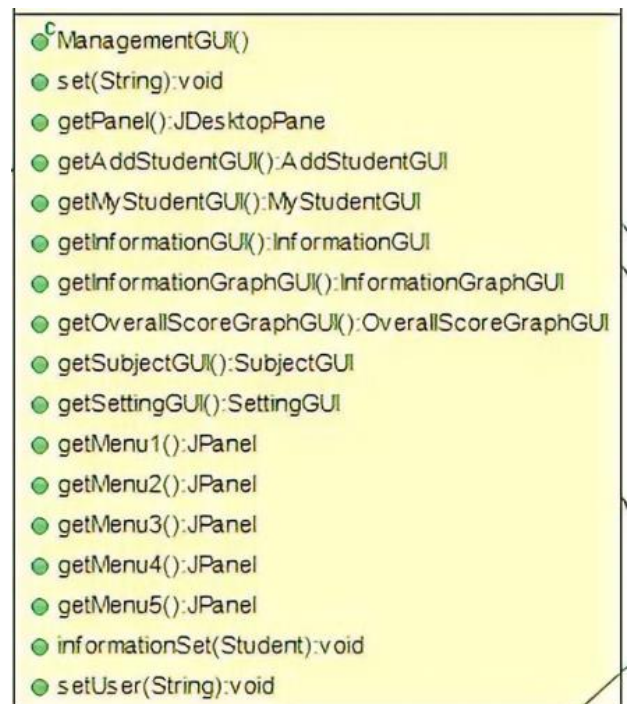
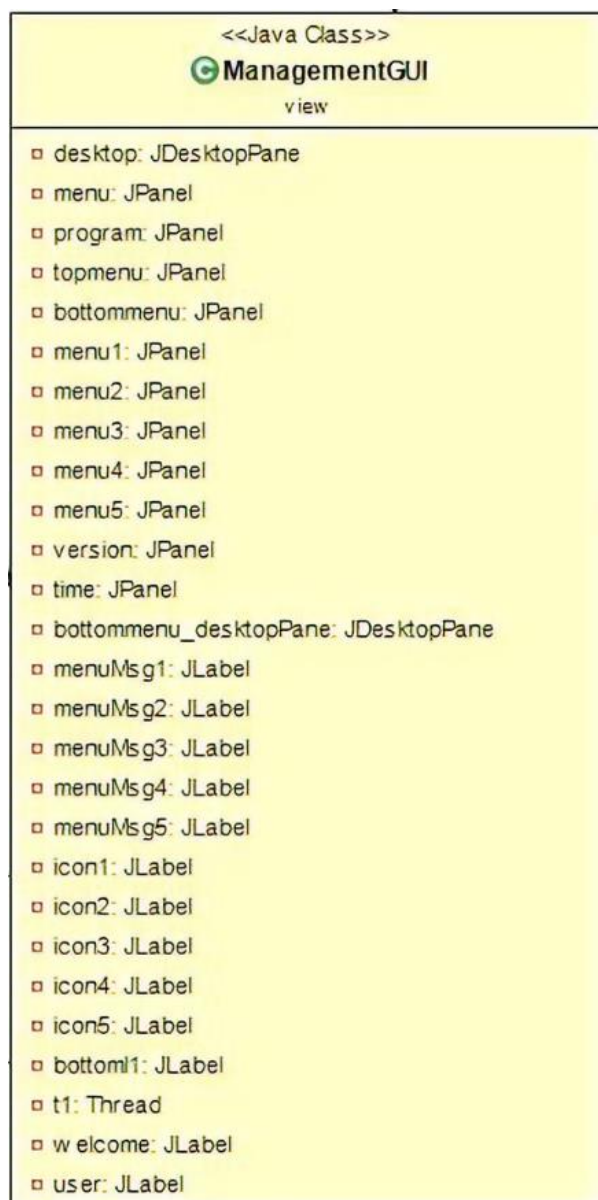
-  LoginGUI()
- getGUI():LoginGUI
- getPanel():JDesktopPane
- getBtn1():JButton
- getF1():JTextField
- getF2():JTextField
- getL3():JLabel
- getL4():JLabel
- getGithub():JLabel
- getBtn2():JButton
- connected(boolean):void

เป็นหน้าที่ให้ผู้ใช้งานล็อกอินและตั้งค่า database

เมธอด

1. LoginGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
3. connected(boolean) เป็นการกำหนดว่าถ้าติดต่อ database สำเร็จให้เปลี่ยนรูปแบบข้อมูลเป็นติ๊กถูก ถ้าไม่สำเร็จให้เป็นกากบาท

9. คลาส ManagementGUI

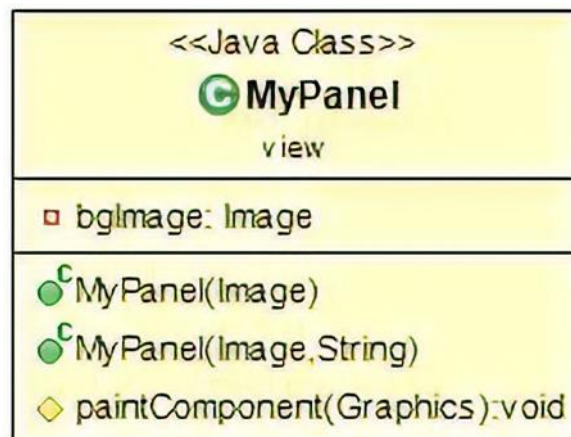


เป็นหน้าที่จะเป็นหน้าหลักของโปรแกรมโดยจะแสดงหลังจากที่ผู้ใช้งานล็อกอินเข้ามาในโปรแกรมแล้ว

เมธอด

1. ManagementGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. set(String) เป็นการเปลี่ยน Panel ที่อยู่ทางด้านขวาให้แสดงผลตามต้องการ
3. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
4. setUser(String) เช็ทว่าในขณะนี้โปรแกรมถูกล็อกอินอยู่ด้วย username อะไร

10. คลาส MyPanel

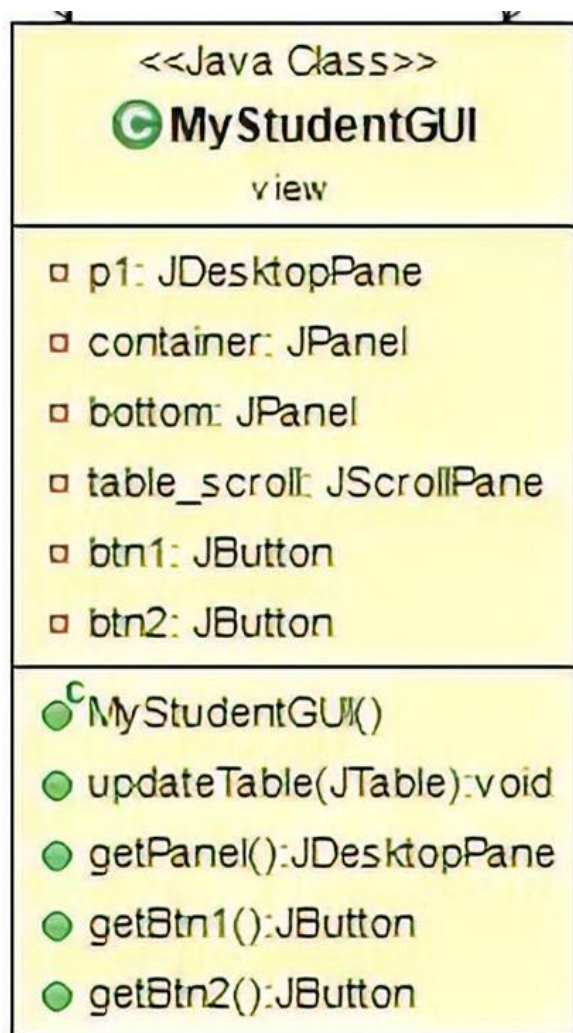


เป็นคลาสที่สืบทอดมาจาก JPanel เพื่อที่จะทำให้สามารถใส่ภาพพื้นหลังได้

เมธอด

1. MyPanel(Image) (คอนสตรัคเตอร์) ทำการรับรูปภาพและนำมา paint ใส่ในพื้นที่หลังขนาดเต็มหน้าต่างโปรแกรม
2. MyPanel(Image, String) (คอนสตรัคเตอร์) ทำการรับรูปภาพและนำมา paint ใส่ในพื้นที่หลังที่มีขนาดตามที่ต้องการ

11. คลาส MyStudentGUI

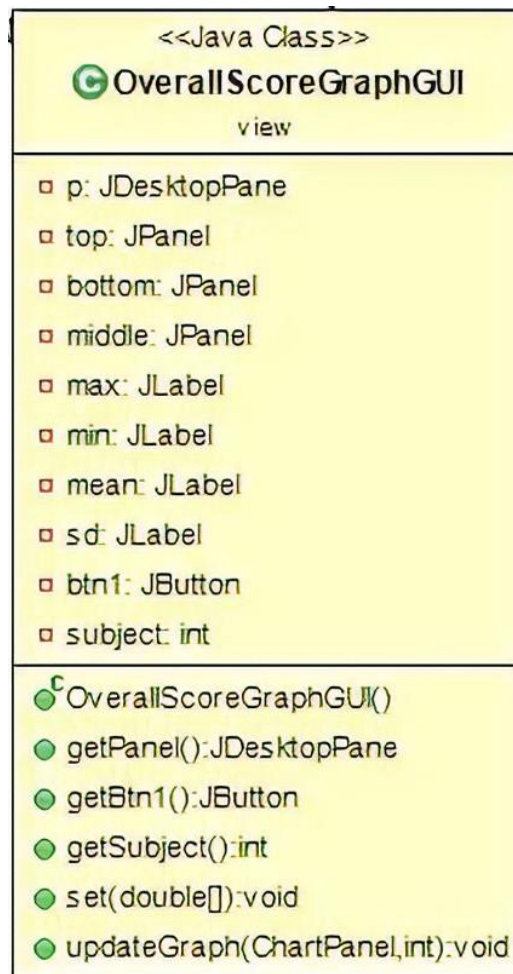


เป็นหน้าที่จะแสดงตารางที่จะแสดงผลรายชื่อนักเรียนที่มีอยู่ในระบบพร้อมกับข้อมูล
เล็กน้อยและในตารางจะมีปุ่มดูข้อมูลเพิ่มเติม และปุ่มลบนักเรียน

เมธอด

1. MyStudentGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. updateTable(JTable) ทำการอัปเดต ตารางรายชื่อนักเรียน
3. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
4. เมธอด Setter และ Getter ของ Attribute ต่างๆ

12. คลาส OverallScoreGraphGUI



เป็นหน้าที่จะแสดงข้อมูลว่าในวิชานี้มีนักเรียนได้เกรดอะไรบ้างออกมาเป็นกราฟและในวิชานั้นคะแนนที่มากและน้อยที่สุดรวมถึงค่าเฉลี่ยและค่า SD เป็นเท่าใด

เมธอด

1. `getPanel()` ทำการคืนค่า JPanel หลักของหน้านี้
2. `getSubject()` ทำการคืนค่าว่าขณะนี้หน้านี้แสดงข้อมูลของวิชาใดอยู่
3. `set(double[])` ทำการแสดงผลค่าคะแนนมากและน้อยสุดรวมถึงค่าเฉลี่ยและค่า SD
4. `updateGraph(ChartPanel)` ทำการอัปเดตการแสดงผลของกราฟ
5. เมธอด Setter และ Getter ของ Attribute ต่างๆ

13. คลาส RegisterGUI

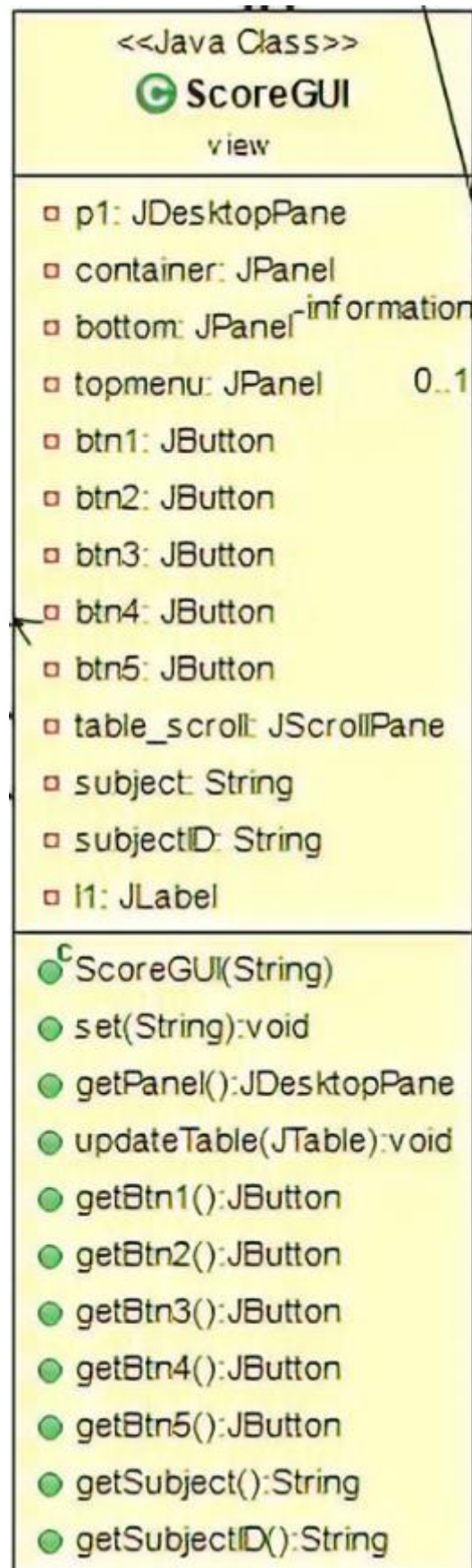


เป็นหน้าที่จะแสดงให้เห็นผู้ใช้งานกรอกข้อมูลสมัครสมาชิก

เมธอด

1. RegisterGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. เมธอด Setter และ Getter ของ Attribute ต่างๆ

14. คลาส ScoreGUI

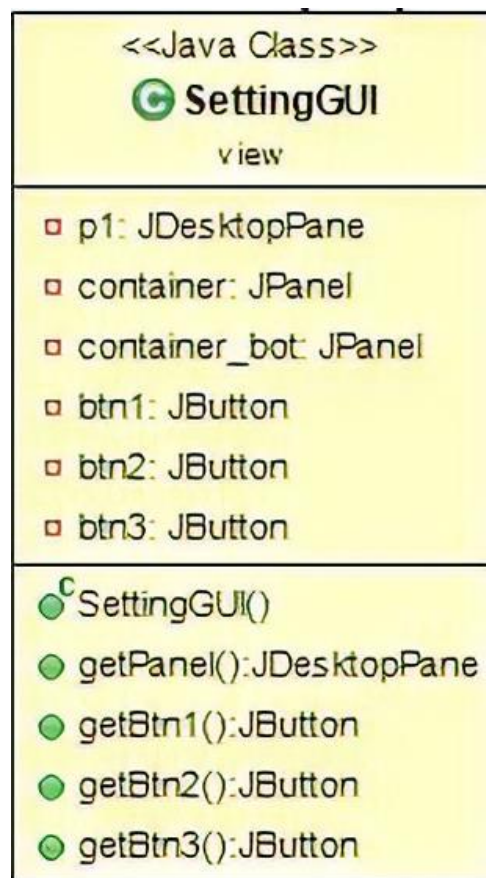


เป็นหน้าที่จะแสดงข้อมูลคะแนนของนักเรียนทุกคนในหนึ่ง

เมธอด

1. ScoreGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. set(String) ทำการกำหนดข้อความในส่วนหัวว่าเป็นวิชาใด
3. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
4. updateTable(JTable) ทำการอัปเดตตาราง
5. เมธอด Setter และ Getter ของ Attribute ต่างๆ

15. คลาส SettingGUI



เป็นหน้าที่แสดงเมนูการตั้งค่าต่างๆ

เมธอด

1. SettingGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. getPanel() ทำการคืนค่า JPanel หลักของหน้านี้
3. เมธอด Setter และ Getter ของ Attribute ต่างๆ

16. คลาส SubjectGUI

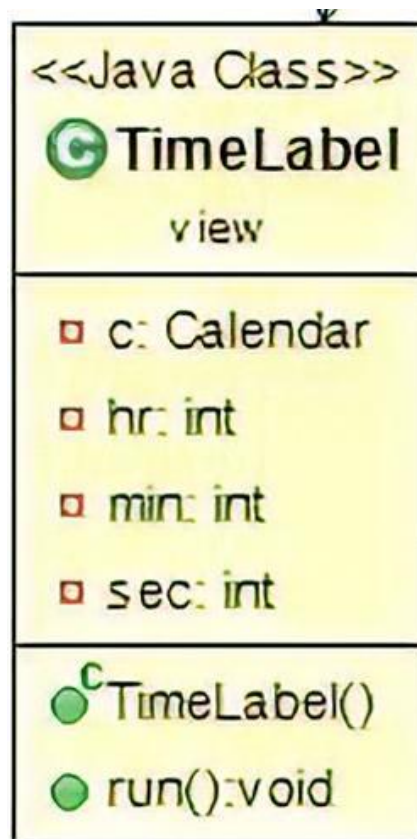


เป็นหน้าที่แสดงให้ผู้เลือกใช้ดูข้อมูลของวิชาอะไรและ ขณะนี้มีวิชาอะไรอยู่ในระบบ
บ้างรวมถึงการเพิ่มวิชาเข้าไป

เมธอด

1. SubjectGUI (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. setSubject(String, String, String) เป็นการตั้งว่า วิชาที่ 1, 2, 3 เป็นอะไรมีวิชาแล้ว
หรือยัง
3. เมธอด Setter และ Getter ของ Attribute ต่างๆ

17. คลาส TimeLabel



เป็นคลาสที่สืบทอดมาจาก JLabel และมีการ implements Runnable เพื่อที่จะรัน Thread ได้ เพราะเป็นข้อความที่จะแสดงเวลาปัจจุบันของผู้ใช้งาน

เมธอด

1. TimeLabel (คอนสตรัคเตอร์) ทำการสร้าง component ต่างๆ
2. run() ทำการดึงค่าเวลาในปัจจุบันของผู้ใช้งานออกมาแสดงผลอัปเดตตลอดเวลา

แหล่งอ้างอิง

เว็บไซต์ : <https://stackoverflow.com/>

ไลบรารี : <https://github.com/atarw/material-ui-swing>
<https://poi.apache.org/>
<https://mongodb.github.io/mongo-java-driver/>
<https://github.com/LGoodDatePicker/LGoodDatePicker>
<http://www.jfree.org/jfreechart/>

หน้าที่ของสมาชิกภายในกลุ่ม

นาย ภาณุวัฒน์ ภาวะโคตร 61070159

- ออกแบบหน้าต่างโปรแกรมและฟังก์ชันการทำงาน, ทำรายงาน

นาย วรภัทร ภัทรเมธาวรกุล 61070199

- ออกแบบหน้าต่างโปรแกรมและฟังก์ชันการทำงาน, ทำรายงาน

นาย วิศรุต พันธุ์ไม้ศรี 61070211

- ออกแบบหน้าต่างโปรแกรมและฟังก์ชันการทำงาน, ทำรายงาน

นาย ศรัณย์ ยันตะบุษย์ 61070218

- ออกแบบหน้าต่างโปรแกรมและฟังก์ชันการทำงาน, เขียนโค้ด, ทำรายงาน

นาย ศิริวิทย์ ยอดจิตร 61070221

- ออกแบบหน้าต่างโปรแกรมและฟังก์ชันการทำงาน, ทำรายงาน