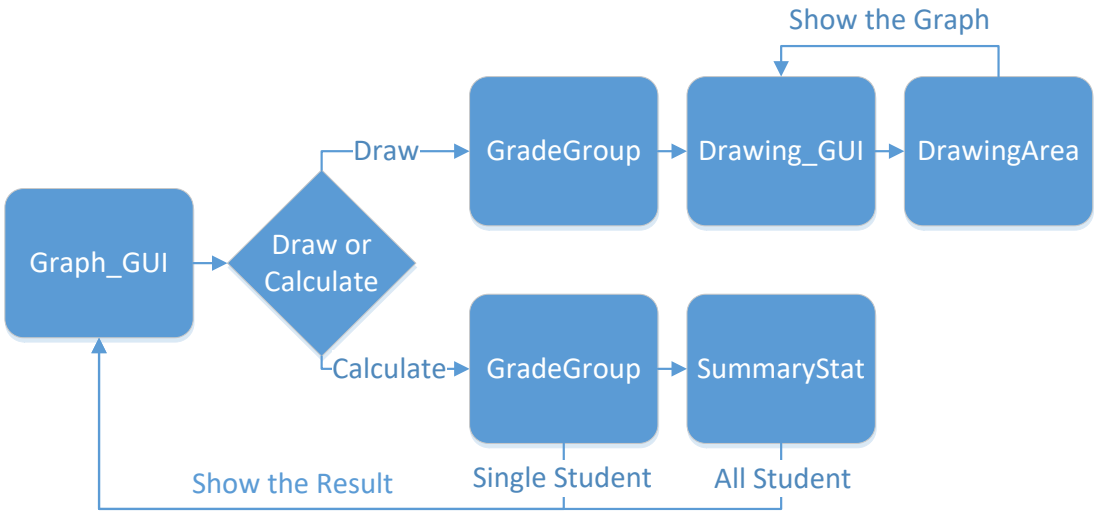
Canvas Graph Project Document

Group: Canvas Graph
Program Language: Java
Member: Zeming Wang, Tao Feng, Lan Chang
Date: 12/16/2016

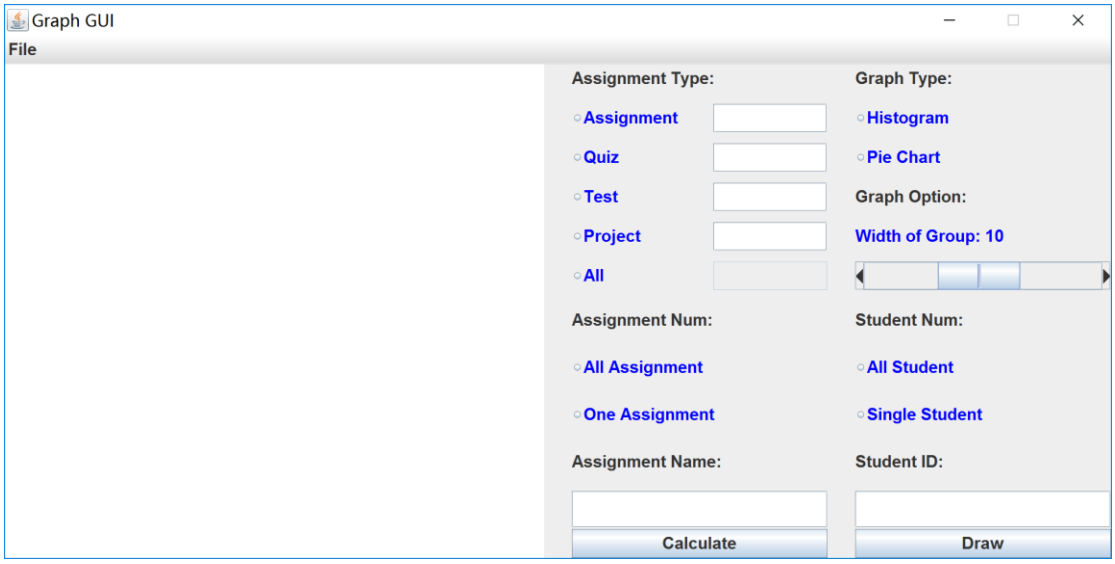
Introduction

The goal of our project is to design a program that generates graphs and shows summary statistics of various assignments, quizzes, tests and projects, and we also can save the graph to designated path.



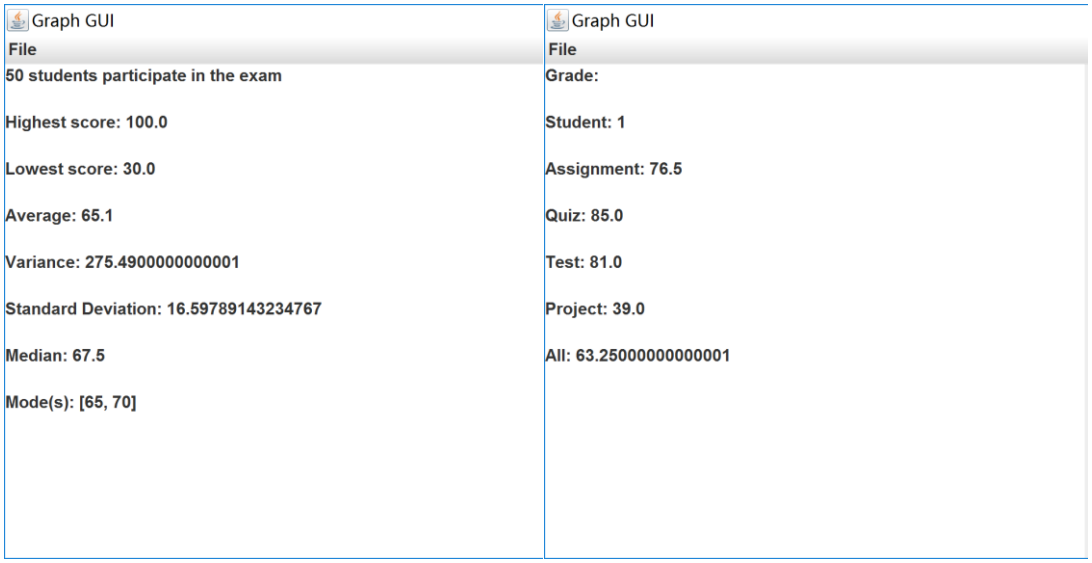
GUI

In the GUI part, there are two major parts. The left one will show the summary statistics. The right one is the selection area. We can decide:
(1) what kind of assignment, all assignment or one assignment, and its name;
(2) all student or single student, and his ID;
(3) what kind of graph;
(4) the width of the group.
There are also two buttons. We can show the summary statistics by pressing the "Calculate" and show the graph by pressing the "Draw".
The are some menu on the top. We can save the graph to designated path, and exit.



Summary statistics

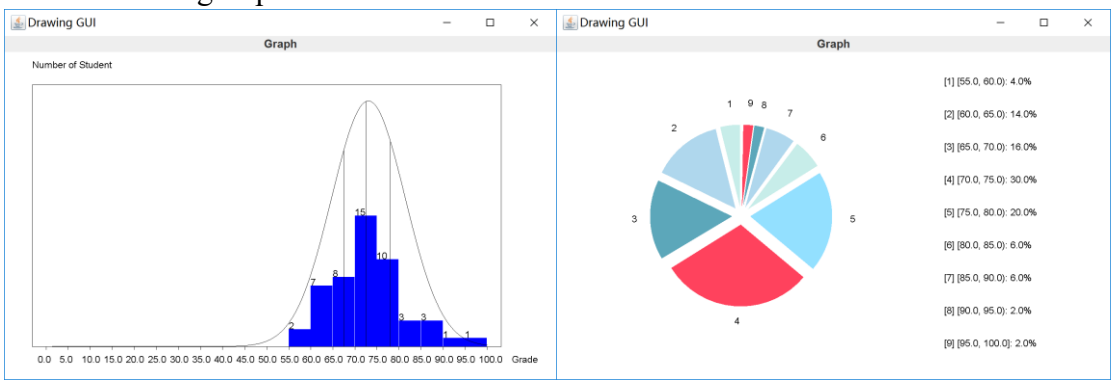
Our program can show the summary statistic for a category of assignments or all the assignments, and show the details of each category of assignment for single student.



Graph

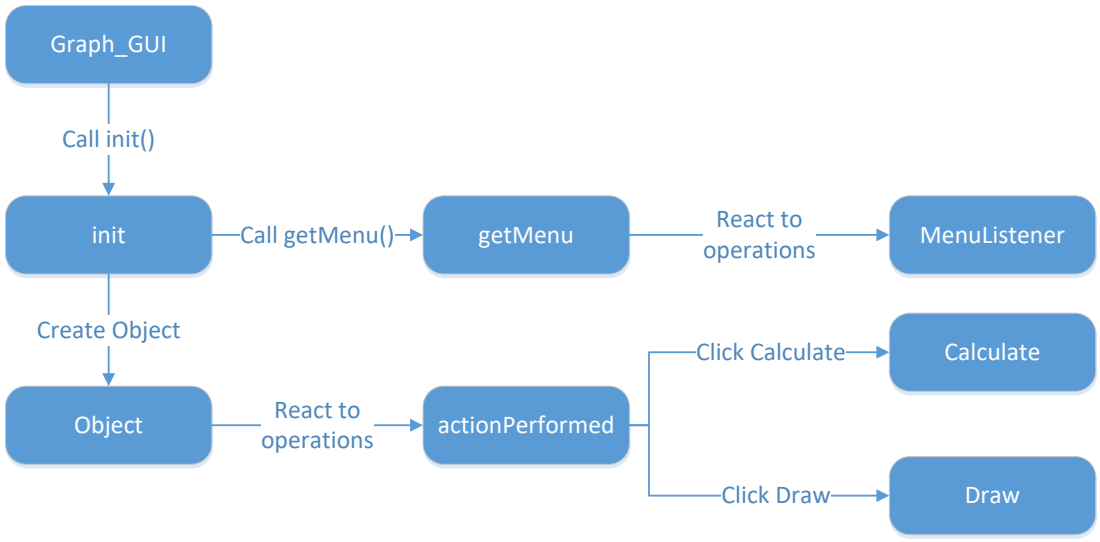
Our program could also display distribution of grades using histograms and pie chart (for an individual assignment, test, quiz or project, and for a category of assignments, tests, quizzes or projects and for all of them with different weights).
The left one is a histogram plot of our sample scores. We can change width of bands to get different graphs. And the curve in this graph is the normal distribution calculated through the variance and mean of our sample scores.

The right one is a pie chart of our sample scores. It illustrates numerical proportion of different score groups.



Part 1: Graph GUI

For the Graph GUI class, we have public methods of the constructor Graph_GUI(), the initiation function init(); actionPerformed(ActionEvent e) shows how the program reacts to the operations done by the user (like selecting a radio button or pressed a button) , and it will also call functions to handle with data and draw the graph; getMenu() will set the menu and menubar; public class MenuListener which make us be able to react to the operation on menu.



Graph_GUI:
(extends Jframe and implements ActionListener and AdjustmentListener)
Holding private elements for GUI.
Setting the basic setup of the GUI window.
Calling init function.

init:
Calling getMenu function, and set the return result as the menubar.
Setting up all buttons' colors, names, positions, font, and add actionListener to them.
Setting the input and output textfield on the GUI.
Placing all buttons and textfield on GUI.

getMenu:
Setting up all options' colors, names, font, and add actionListener to them.
MenuListener:(implements ActionListener)
Setting what should be done when "exit", "save", "open" operations are clicked.

actionPerformed:
Getting the input from textfield and buttons.
Calling corresponding functions or methods.

Part 2: SummaryStat

The SummaryStat class calculates the following statistics:

(1) mean: the average of all the scores in a certain task.
(2) mode: the most frequently appeared score(s) in a certain task.
(3) median: The "median" is the "middle" value in the list of scores.
(4) variance: variance is the expectation of the squared deviation of a random variable from its mean, and it informally measures how far a set of numbers are spread out from their mean.
(5) standard deviation: it is the square root of variance. A low standard deviation indicates that the scores tend to be close to the mean (also called the expected value) of the scores, while a high standard deviation indicates that the scores are spread out over a wider range of values.
(6) lowest score and highest score.

The methods in this class are listed as follow:

count(ArrayList<Double> list): get the number of students who has a score in a certain task.
min(ArrayList<Double> list): calculate the mininum score
max(ArrayList<Double> list): calculate the highest score
average(ArrayList<Double> list): calculate the average score
median(ArrayList<Double> list): calculate the median score
mode(final ArrayList<Double> list): find the mode(s) in a list of score
variance(ArrayList<Double> list): calculate variance
StdDev(ArrayList<Double> list): calculate standard deviation

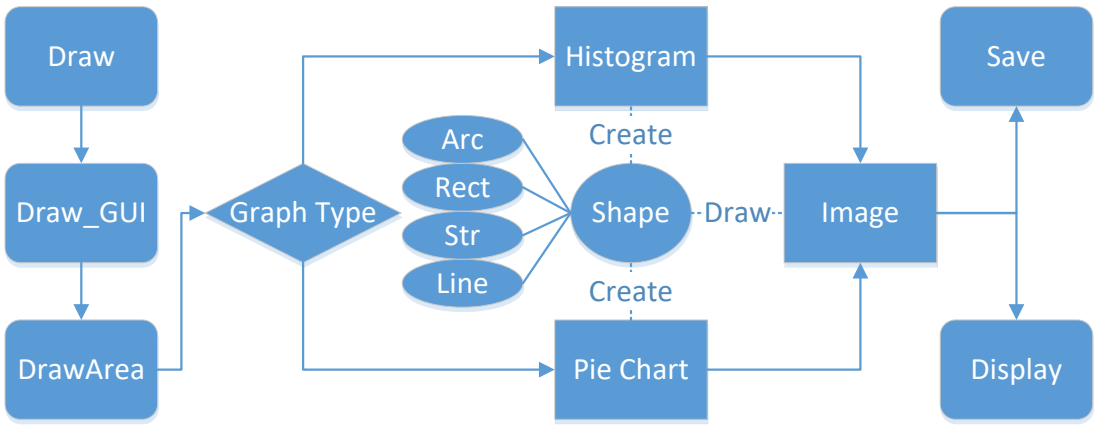
Part3: Graph

Class:
Drawing_GUI, DrawingArea, (abstract) Shape, Rect, Arc, Line, Str.

Method:
In the Drawing_GUI class, there are 2 methods, one is constructor, and the other is save().
The constructor sets up the GUI window to show the graph, and create DrawingArea object to draw the graph.
The save() method is to call the saveImage method in DrawingArea to save the graph.

In the DrawingArea class, there are 9 methods, they are constructor, paint(), drawingHistogram(), drawingPie(), clear(), variance(), average(), Gauss() and saveImage().
The constructor gets the parameters from Drawing_GUI.
The paint() sets up the background of the graph, decides to call drawingHistogram() or drawingPie() and paint the shapes on the graph.
The drawingHistogram() is to draw the histogram and distribution and save the graph into BufferedImage.
The drawingPie() is to draw the pie chart and save the graph into BufferedImage.
The clear() is to clear the graph.
The variance(), average(), Gauss() are to calculate the variance, average, distribution.
The saveImage() is to save the graph to designated path.

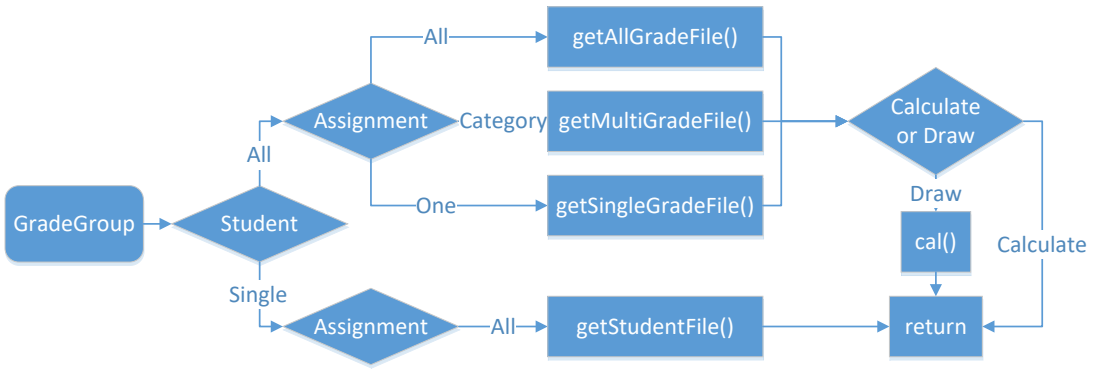
In the Shape, Rect, Arc, Line, Str classes, they are used to draw the histogram and pie chart.



Part 4: Grade

Class:
GradeGroup.

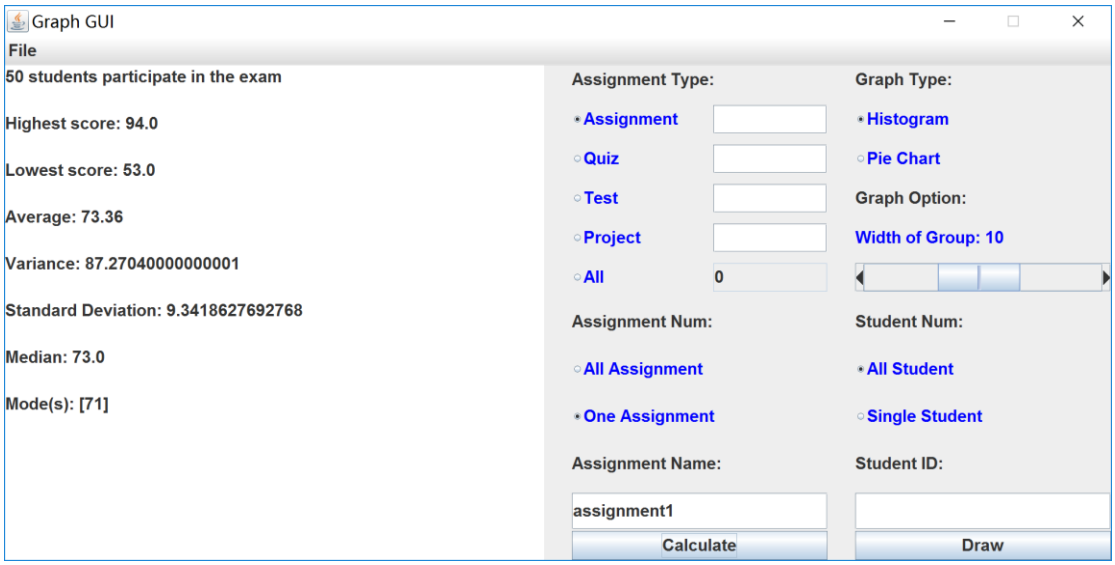
Method:
In the GradeGroup class, there are 12 methods. They are two constructors, getStudentFile(), getAllGradeFile(), getMultiGradeFile(), getSingleGradeFile(), cal(), getNum(), getGrade(), getDone(), getFull(), getStr().
One constructor is called by draw, the other is called by calculate. They decide which kind of grade do they want to get.
The getStudentFile() is to get the grade of single student and all assignments.
The getAllGradeFile() is to get the grade of all students and all assignments.
The getMultiGradeFile() is to get the grade of all students and a category of assignments.
The getSingleGradeFile() is to get the grade of all students and one assignment.
The cal() is to calculate how many students are there in every grade group.
The getNum(), getGrade(), getDone(), getFull(), getStr() are to return the numbers of students in each group, the grade, whether the calculation is done, the fullscore of this grade and the StringBuffer of one student and all assignments.



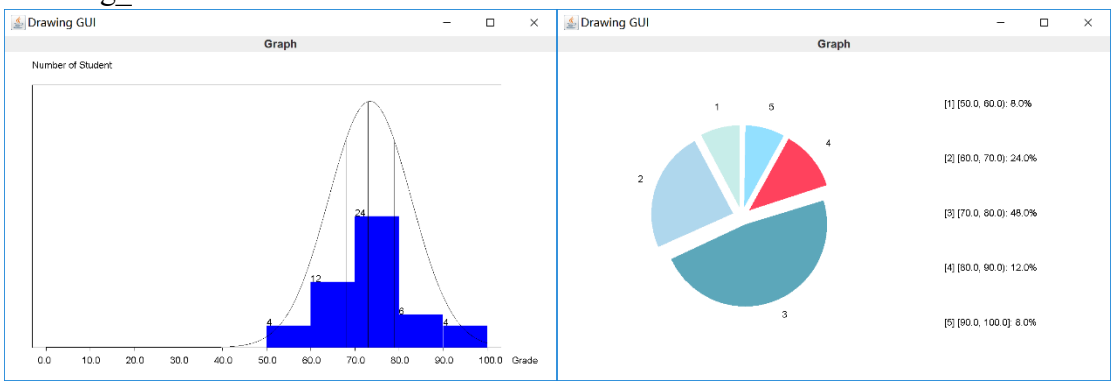
Example

Case 1:
Assignment Num: One Assignment;
Assignment Name: assignment1;
Student Num: All Student.

Graph_GUI:

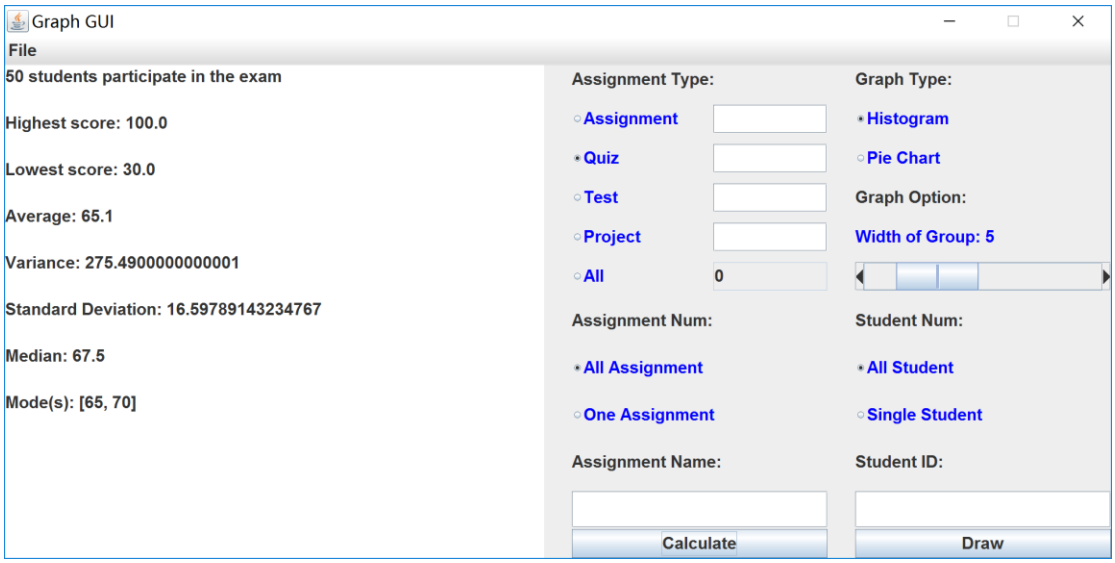


Drawing_GUI:

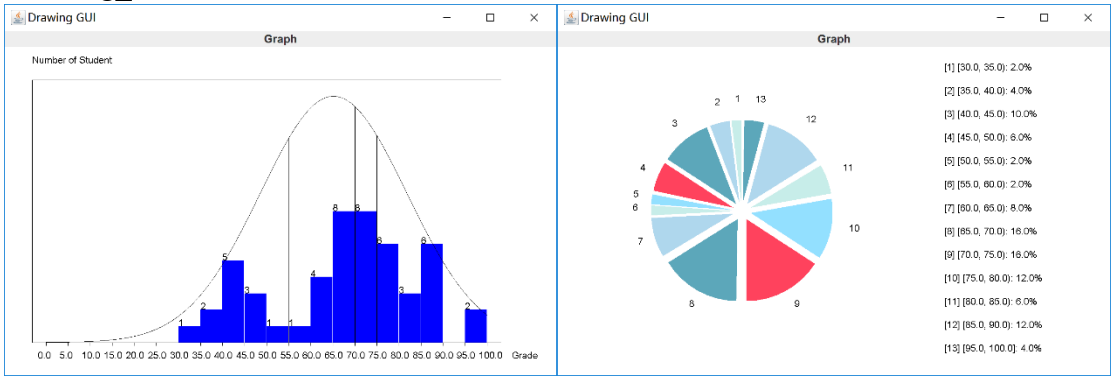


Case 2:
Assignment Type: Quiz,
Assignment Num: All Assignment,
Student Num: All Student.

Graph_GUI:

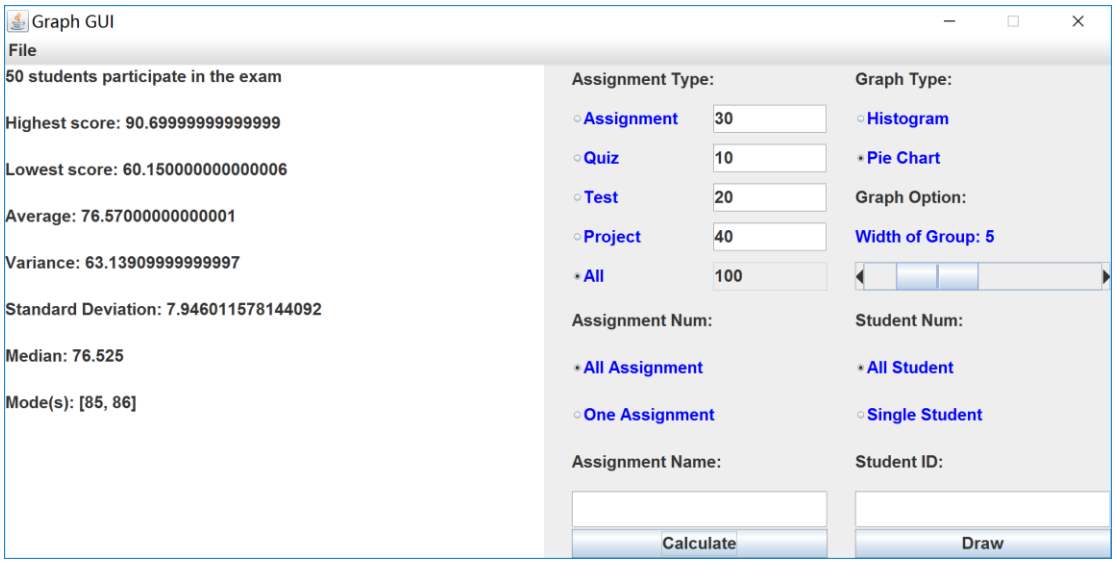


Drawing_GUI:

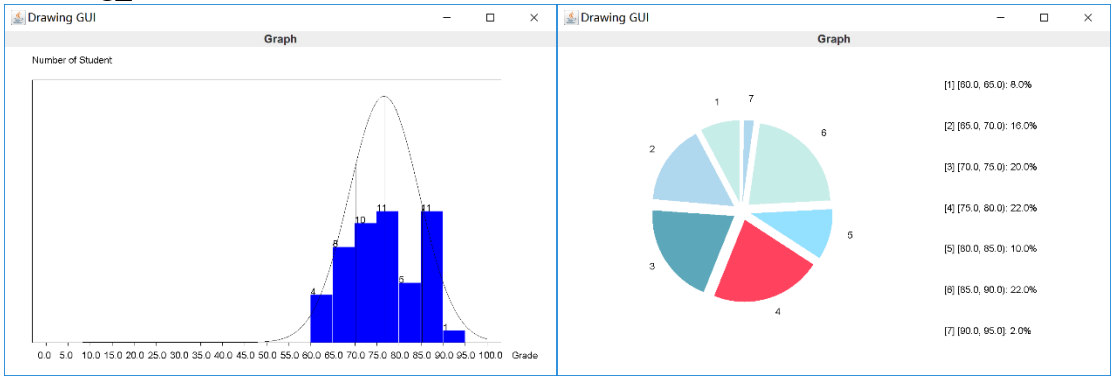


Case 3:
Assignment Type: All,
Assignment Num: All Assignment,
Student Num: All Student.

Graph_GUI:

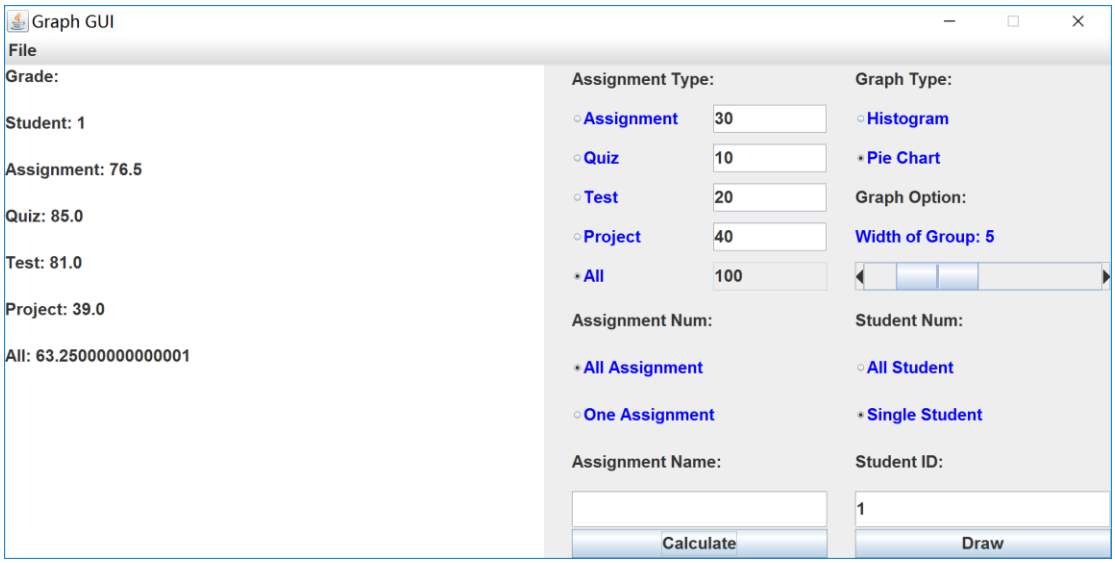


Drawing_GUI:

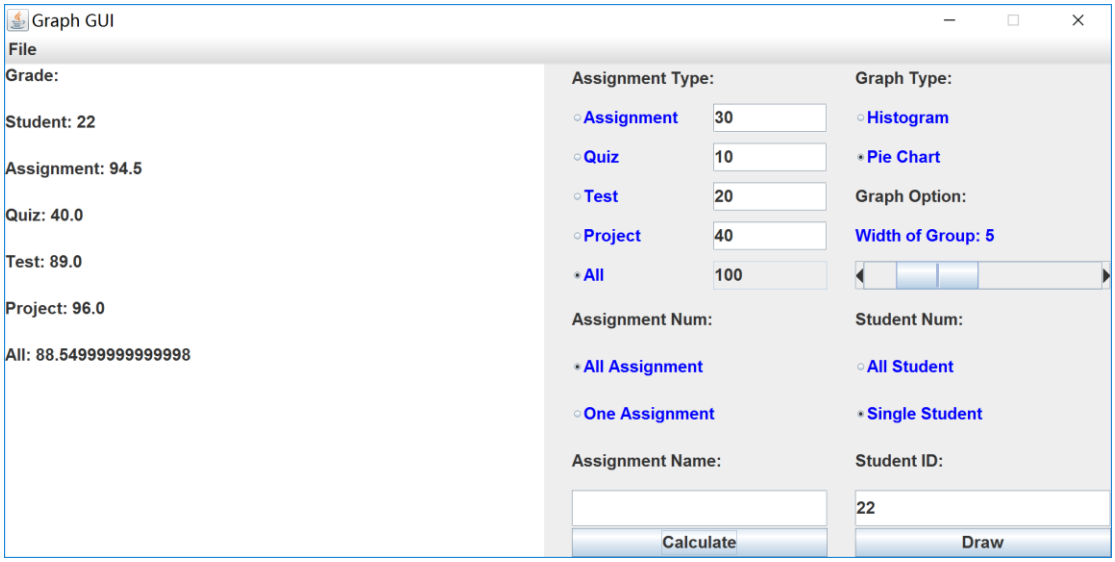


Case 4:
Assignment Type: All,
Assignment Num: All Assignment,
Student Num: Single Student.

Student ID: 1

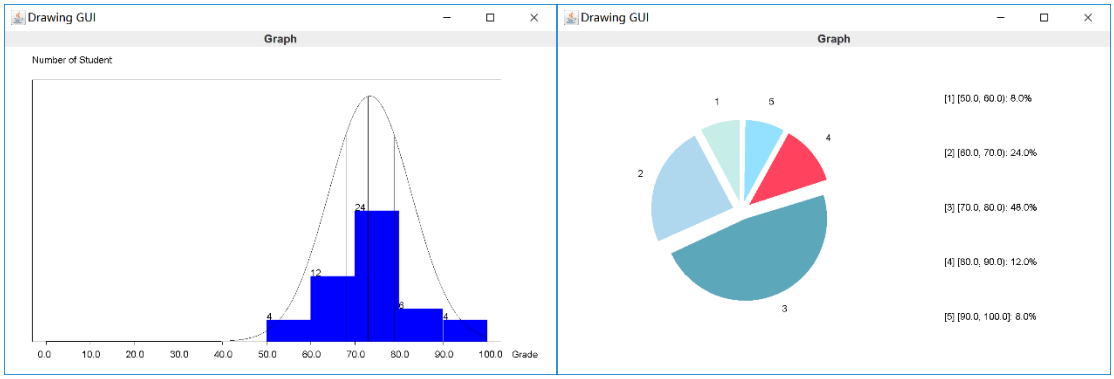


Student ID: 22

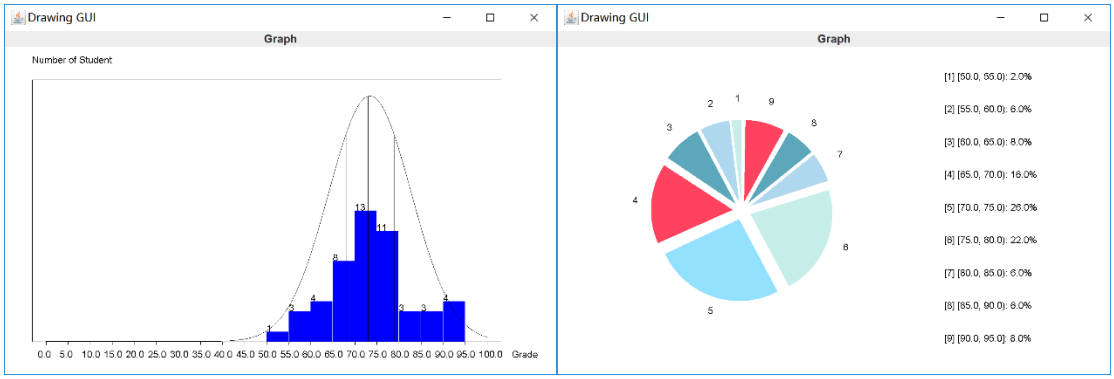


Case 5:
Different Width of Group.
(We can move the scroll bar of the width to get the width of group we want)

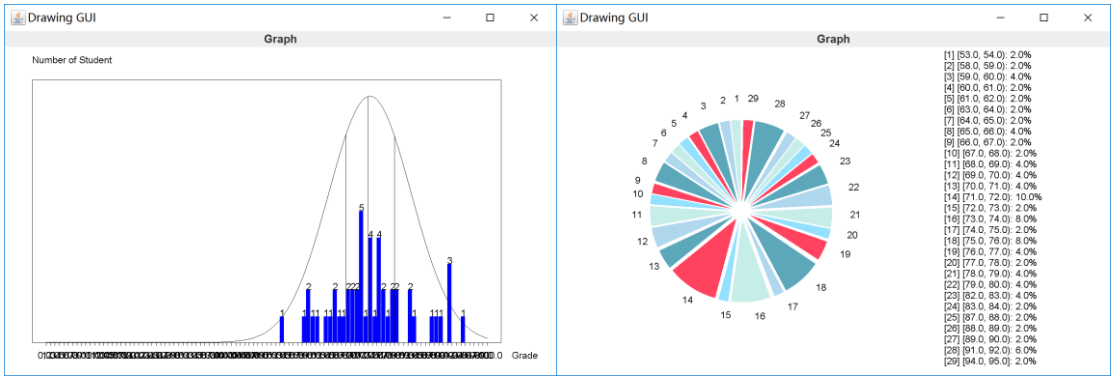
Width: 10



Width: 5

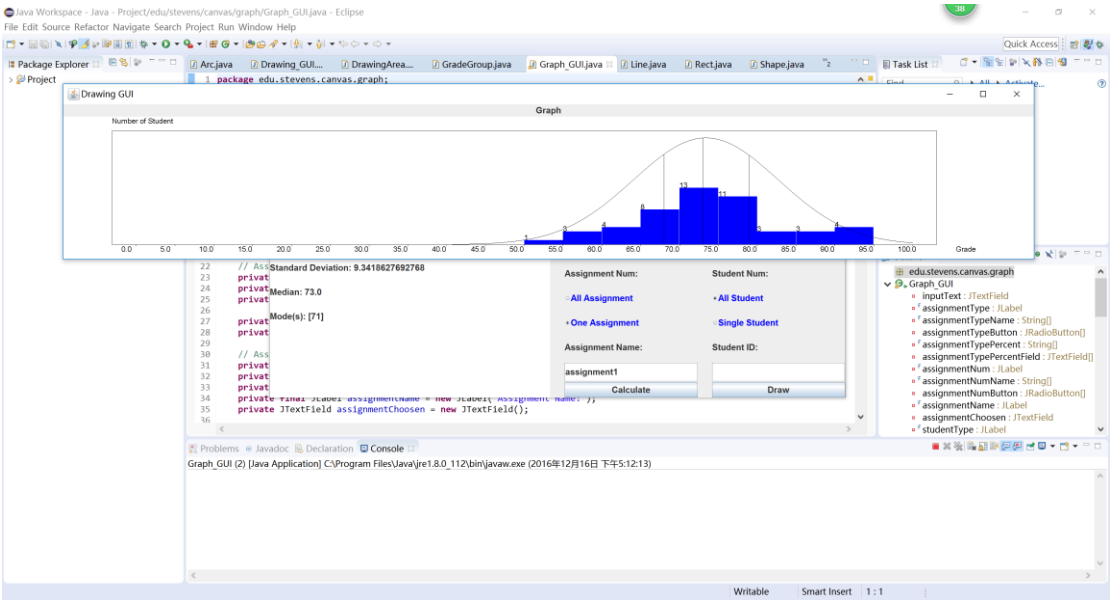


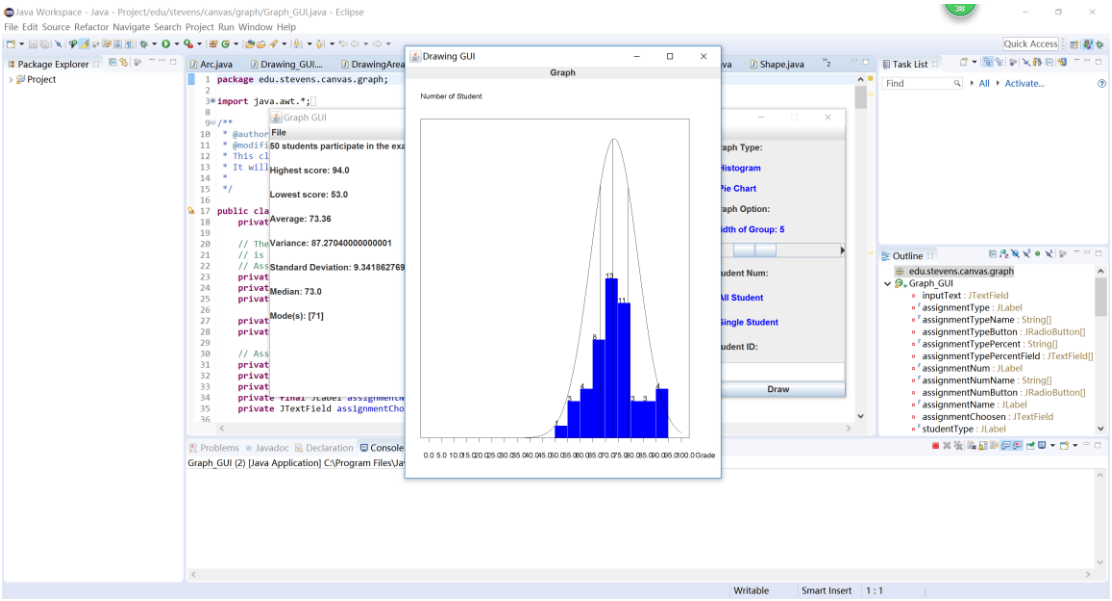
Width: 1



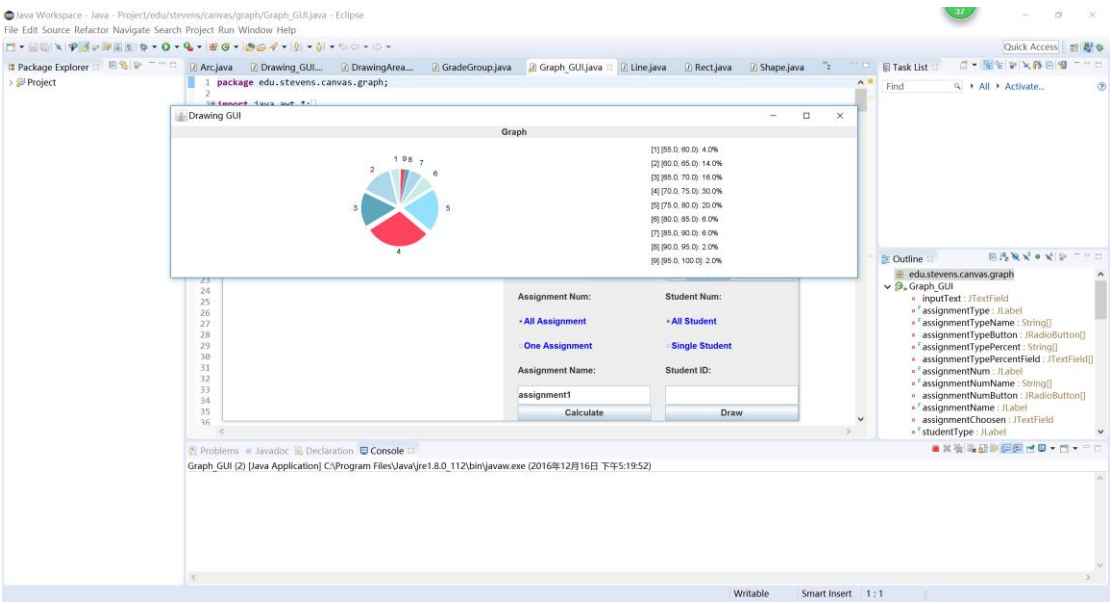
Case 6:
Different Size of Graph.
(We can adjust the size of the Drawing_GUI window, and at the same time, the graph will also change with the same scale)

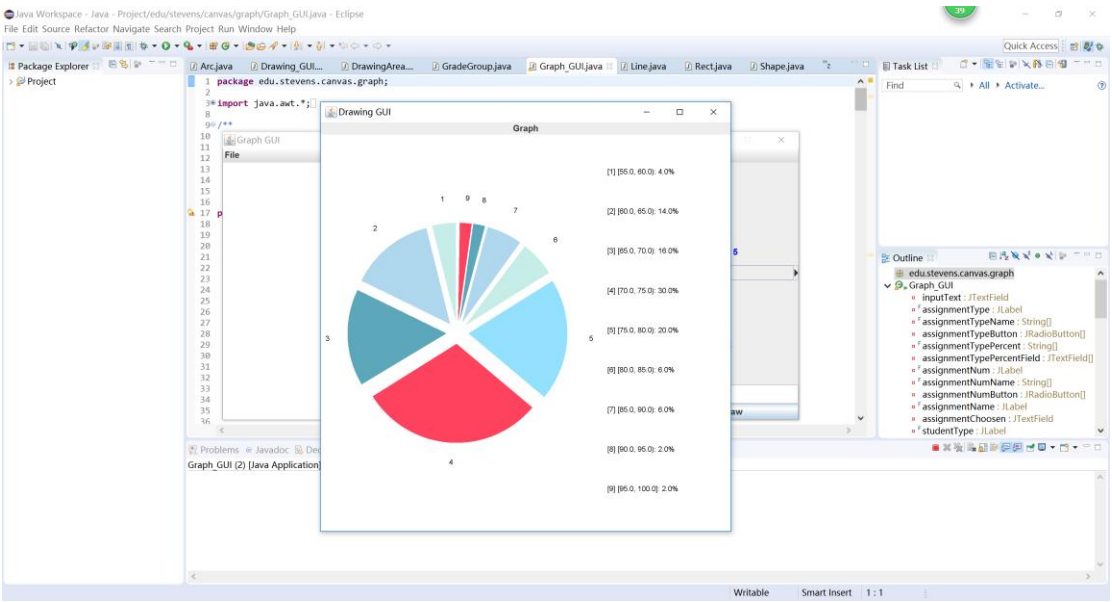
Histogram:





Pie Chart:

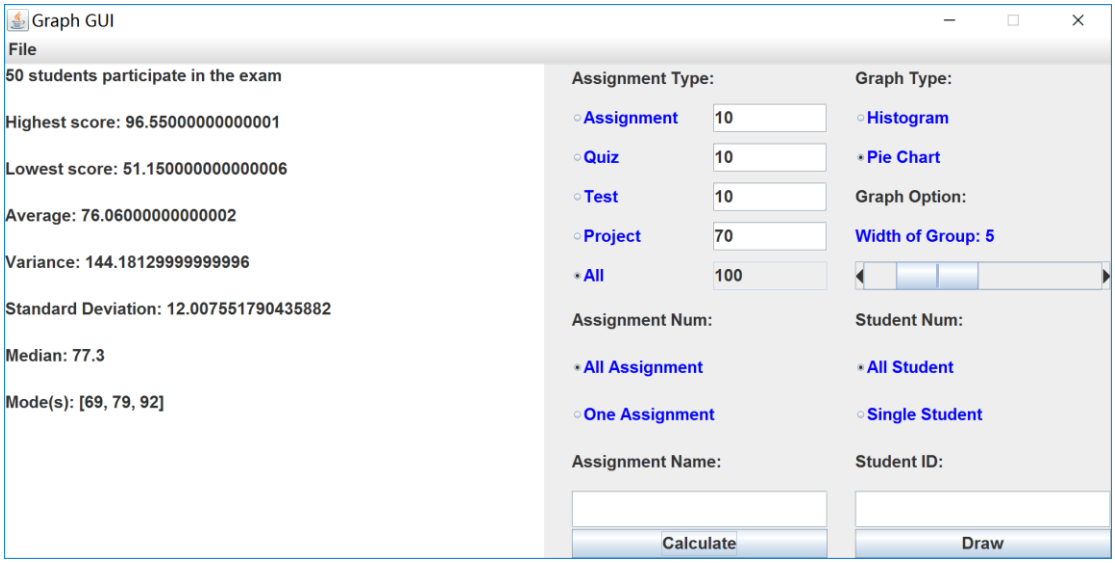

Drawing GUI
Graph
[1] [55.0, 60.0): 4.0%
[2] [60.0, 65.0): 14.0%
[3] [65.0, 70.0): 16.0%
[4] [70.0, 75.0): 30.0%
[5] [75.0, 80.0): 20.0%
[6] [80.0, 85.0): 6.0%
[7] [85.0, 90.0): 6.0%
[8] [90.0, 95.0): 2.0%
[9] [95.0, 100.0]: 2.0%
Assignment Num:
Student Num:
All Assignment
All Student
One Assignment
Single Student
Assignment Name:
Student ID:
assignment1
Calculate
Draw


Drawing GUI
Graph
[1] [55.0, 60.0): 4.0%
[2] [60.0, 65.0): 14.0%
[3] [65.0, 70.0): 16.0%
[4] [70.0, 75.0): 30.0%
[5] [75.0, 80.0): 20.0%
[6] [80.0, 85.0): 6.0%
[7] [85.0, 90.0): 6.0%
[8] [90.0, 95.0): 2.0%
[9] [95.0, 100.0]: 2.0%

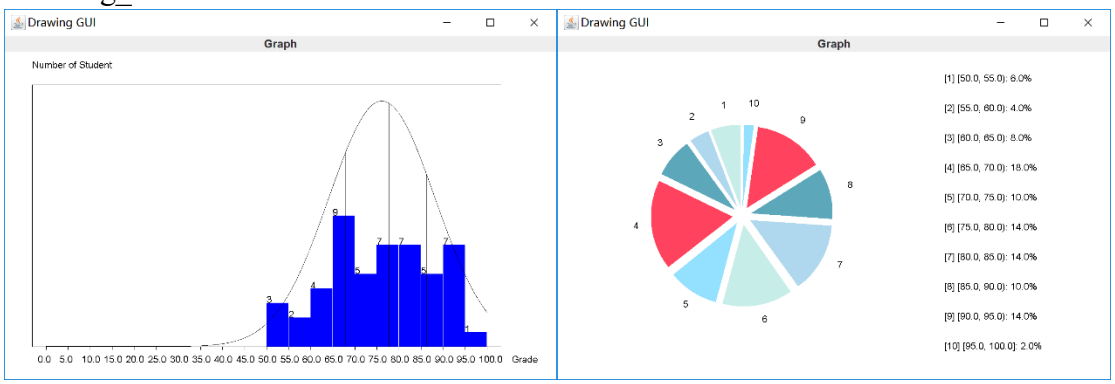
Case 7:
Different weight of assignment.
(We can adjustment the weight of each category of assignment)

Weight: 10%, 10%, 10%, 70%

Graph_GUI:

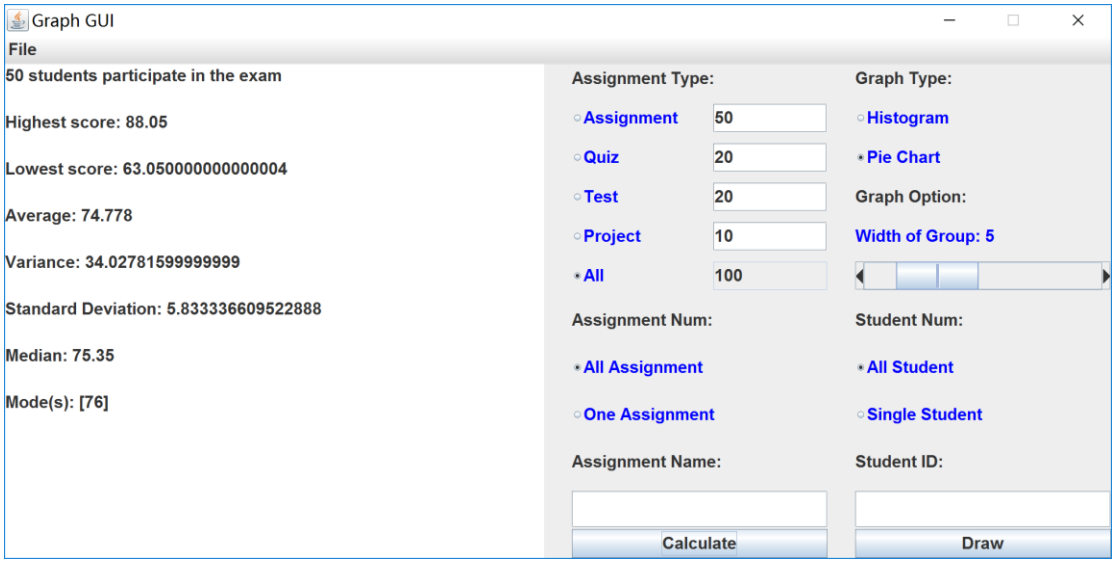


Drawing_GUI:

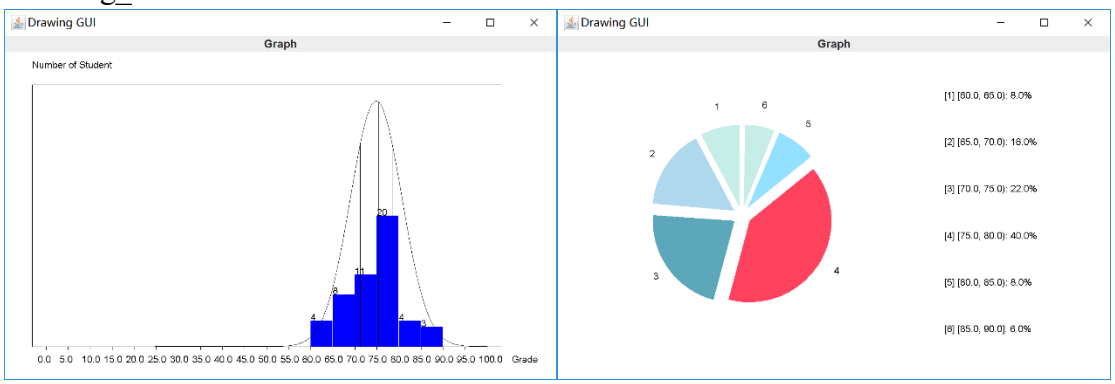


Graph_GUI:

Weight: 50%, 20%, 20%, 10%

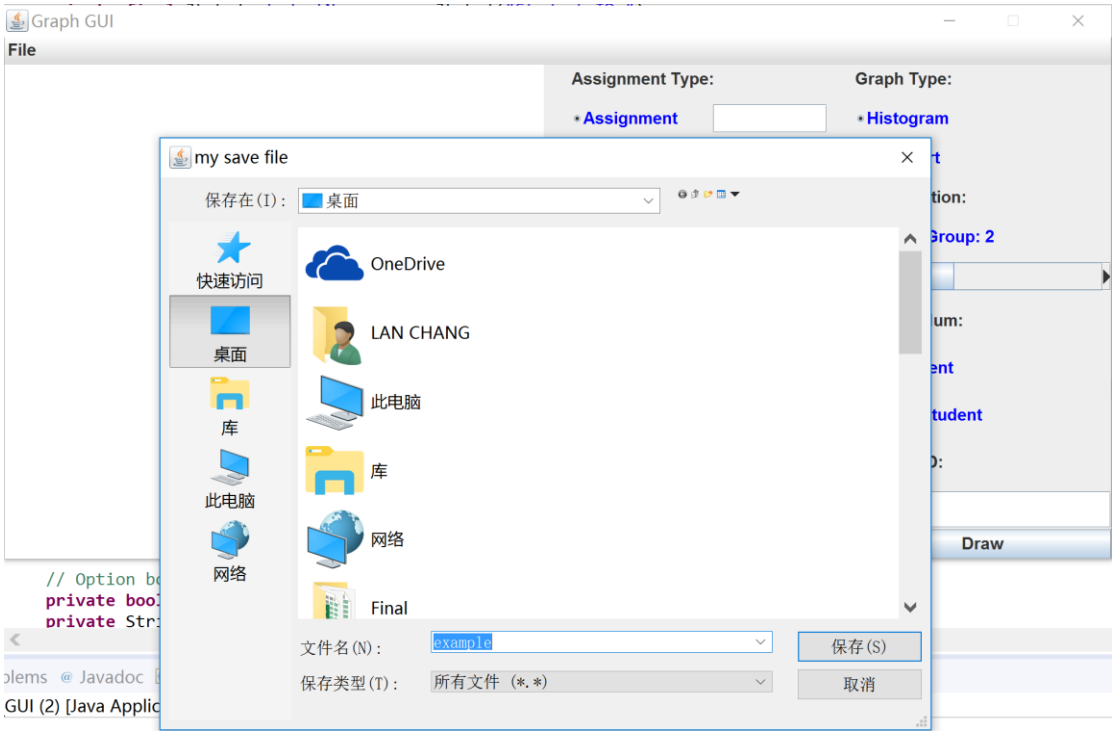


Drawing_GUI:

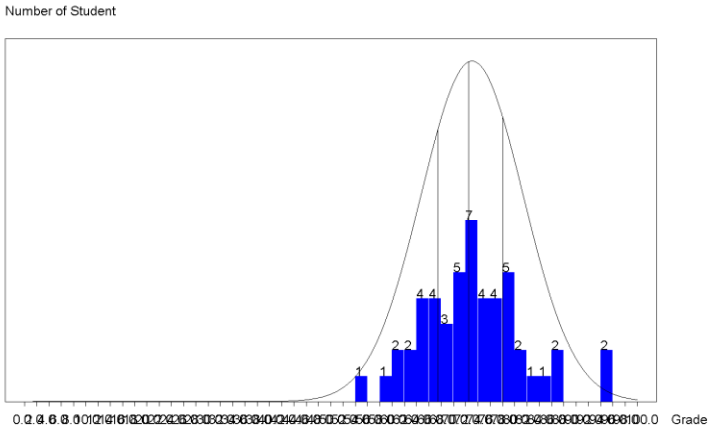


Case 8:
Save the graph.
(We can save the graph to designated path with any name)

Click “File” -> Click “Save the graph” -> Find the path you want to save.
For example: path: Desktop, name: example.

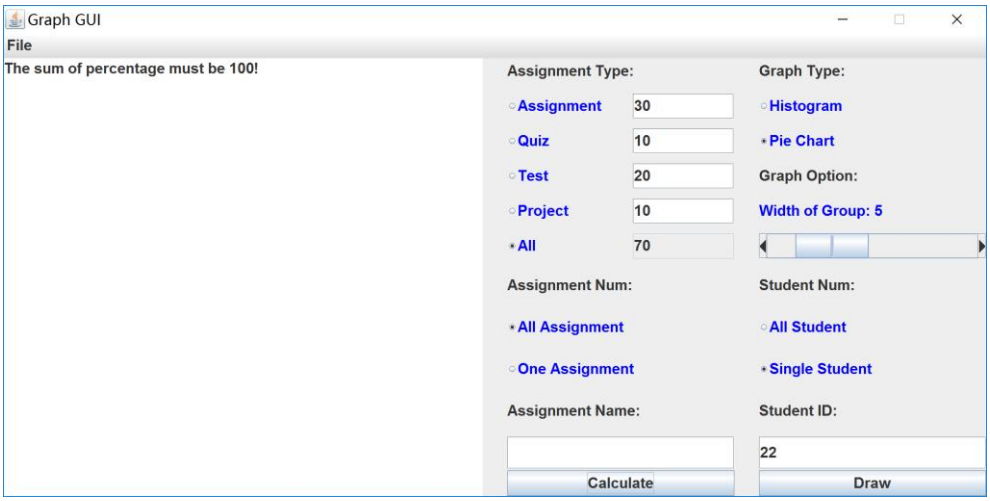


“example.png” file on the Desktop.

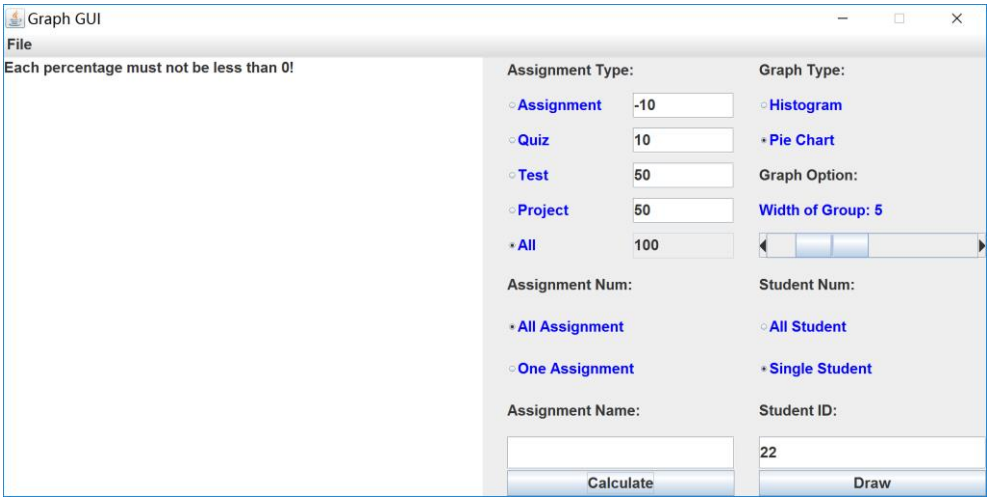


Case 9:
Showing Error.

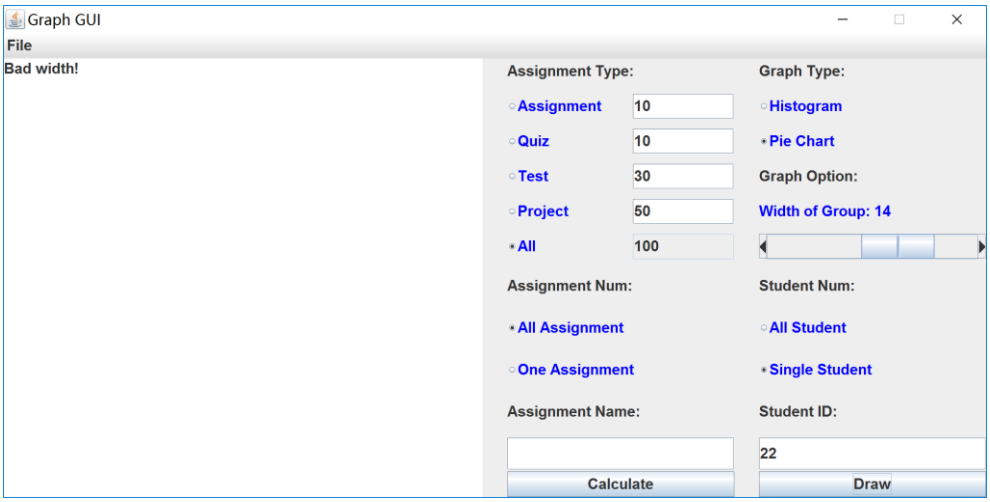
The sum of percentage is not 100.



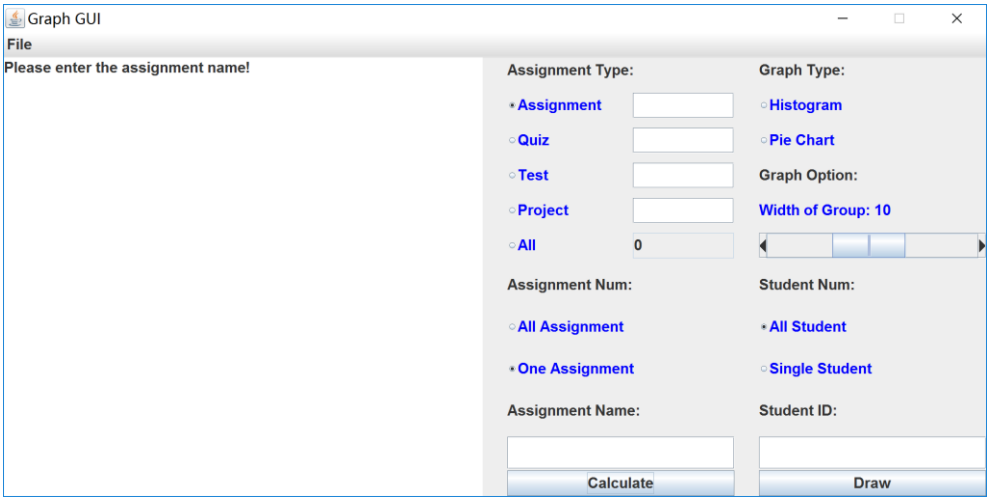
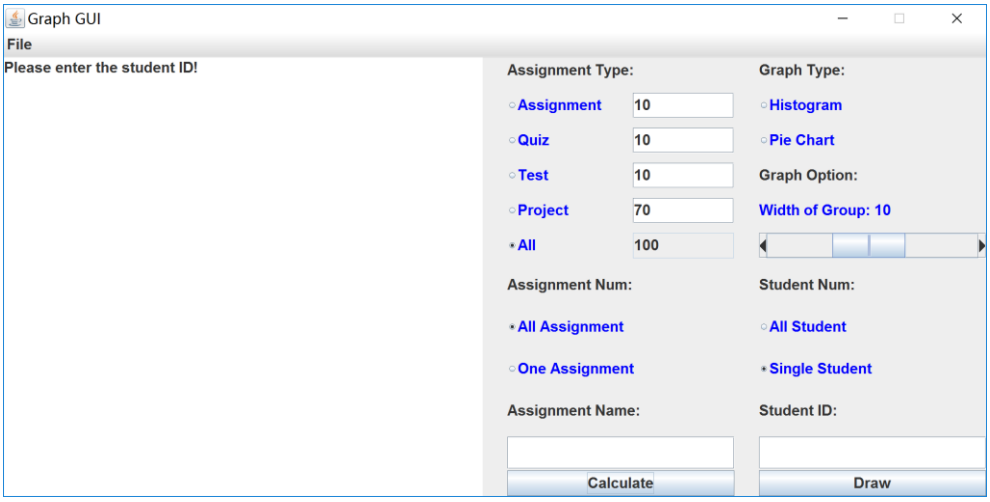
There is a percentage is less than 0.



The width is not good.



No assignment name entering.



No student ID entering.