Canvas Graph Project Document

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

1. Introduction

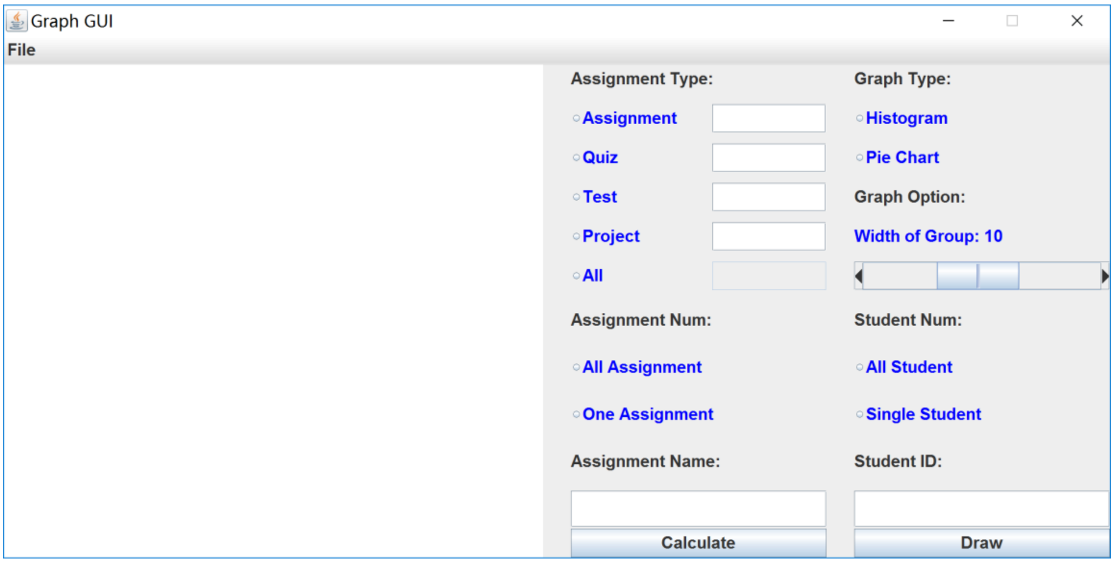
This program is to design a program that can generate some graphs and show the summary statistics of various assignments, quizzes, tests and projects, and we also can save the graph to designated path.

In the GUI part, there are two major parts. The left part will show the summary statistics. The right part is the selection area. We can decide:

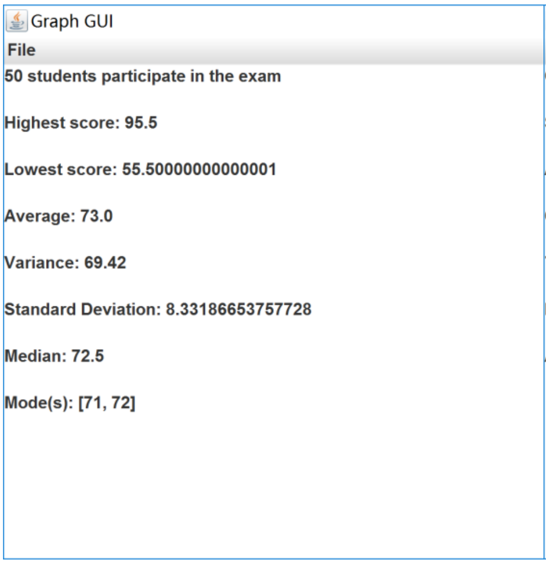
(1) what kind of assignment, all assignment or one assignment, and its name;
(2) all student or single student, and his ID;
(3) what kind of graph;
(4) the width of the group.

There are also two buttons. We can show the summary statistics by pressing the "Calculate" and show the graph by pressing the "Draw".

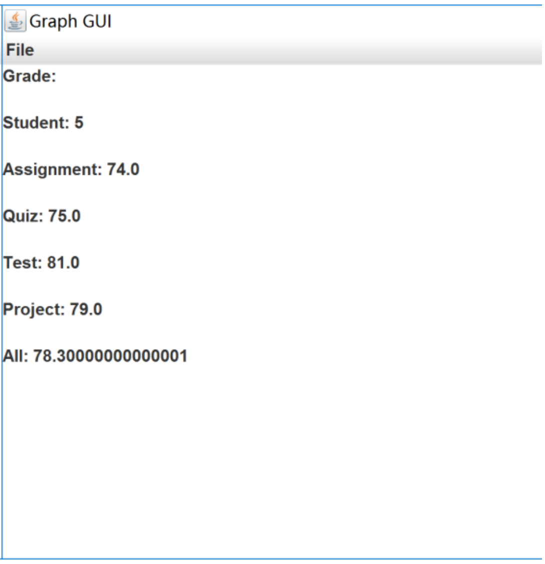
The are some menu on the top. We can save the graph to designated path, and exit.



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

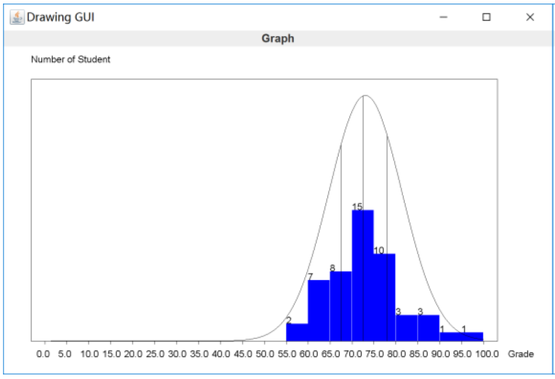


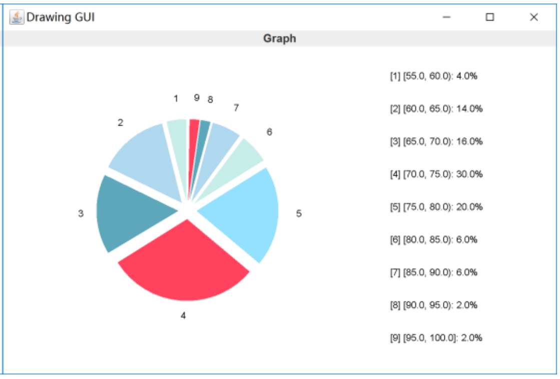
```
50 students participate in the exam

Highest score: 95.5

Lowest score: 55.50000000000001

Average: 73.0

Variance: 69.42

Standard Deviation: 8.33186653757728

Median: 72.5

Mode(s): [71, 72]
```



```
Grade:

Student: 5

Assignment: 74.0

Quiz: 75.0

Test: 81.0

Project: 79.0

All: 78.30000000000001
```

Our program could 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). For the histogram, 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. For the pie chart, it illustrates numerical proportion of different score groups.

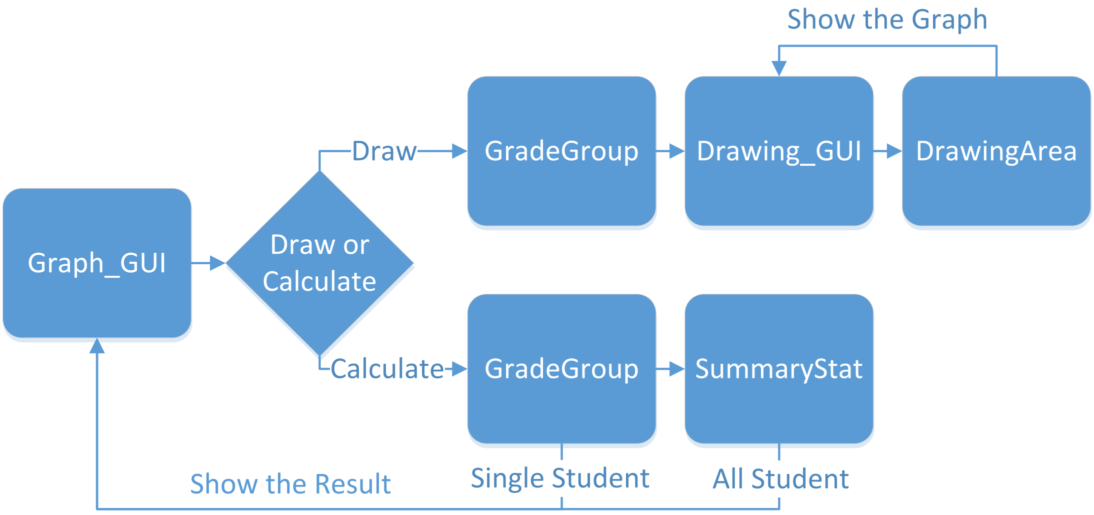




2. Components

2.1 Overview

In the Graph_GUI, you can choose the attributes about assignment, student, graph. And next, if you want to draw the graph, it will create a GradeGroup class to get the grade file and get the number of student in each grade group such as [85, 90), [90, 95) ... Then, it will create Drawing_GUI and DrawingArea to draw the graph.
If you want to show the summary statistics, it will also create a GradeGroup class to get the grade file and get the grade want you want. If you just want the single student's grade, it will return to the Graph_GUI, and if you want the summary statistics of all the students, it will use some methods in SummaryStat class to calculate, and then it will also return to the Graph_GUI.



2.2 Graph GUI

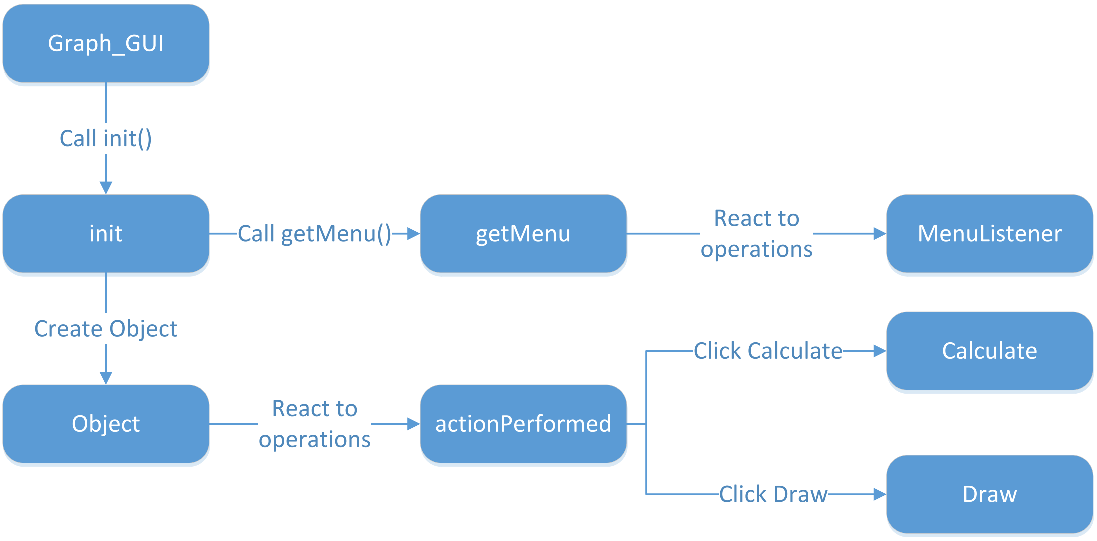
For the Graph GUI class, we have public methods of the constructor Graph_GUI(), the initiation function init(); actionPerformed(ActionEvent e) shows how the program reacts to the operations done by the user (like selecting a radio button or pressed a button) , and it will also call functions to handle 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.



2.3 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 minimum 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

2.4 Graph

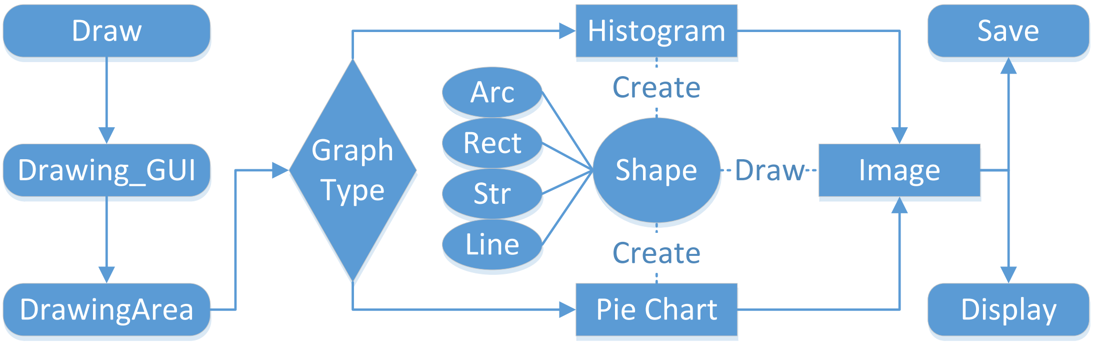
Class:
Drawing_GUI, DrawingArea, (abstract) Shape, Rect, Arc, Line, Str.

Method:
Drawing_GUI:.
Constructor: set up the GUI window to show the graph, and create DrawingArea object to draw the graph.
save(): call the saveImage method in DrawingArea to save the graph.

DrawingArea:
Constructor: get the parameters from Drawing_GUI.
paint(): set up the background, decide to call which graph's method and paint the shapes on the graph.
drawingHistogram(): draw the histogram and distribution and save it into BufferedImage.
drawingPie(): draw the pie chart and save the graph into BufferedImage.
clear(): clear the graph.
variance(), average(), Gauss(): calculate the variance, average, distribution.
saveImage(): save the graph to designated path.

Shape, Rect, Arc, Line, Str: be used to draw the histogram and pie chart.

Detail:
If you choose to draw the graph, it will create a Drawing_GUI object to create a window and then it will create a DrawingArea object to draw. The DrawingArea will decide which kind of graph to draw, if you want to draw the histogram, it will add some rectangles, lines and strings to ArrayList of shapes, and if you want to draw the pie chart, it will add some arcs and strings to ArrayList of shapes. Then, it will draw these shapes on the Graphics and display the graph on the Drawing_GUI window. Both of them will also create an BufferedImage to store the graph, if you choose to save the graph, it will get the path you want to save and then save the BufferedImage there.

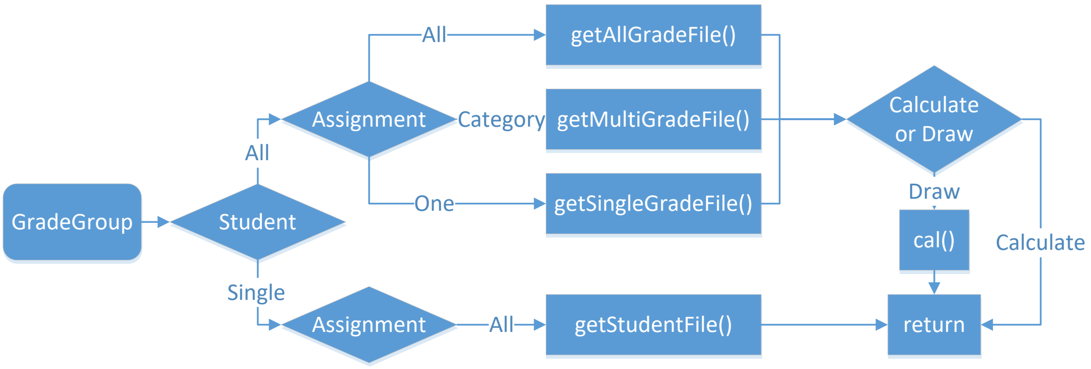


2.5 Grade

Class:
GradeGroup.

Method:
One constructor is called by draw, the other is called by calculate. They decide which kind of grade do they want to get.
getStudentFile(): get the grade of single student and all assignments.
getAllGradeFile(): get the grade of all students and all assignments.
getMultiGradeFile(): get the grade of all students and a category of assignments.
getSingleGradeFile(): get the grade of all students and one assignment.
cal(): calculate how many students are there in every grade group.
getNum(), getGrade(), getDone(), getFull(), getStr(): 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.

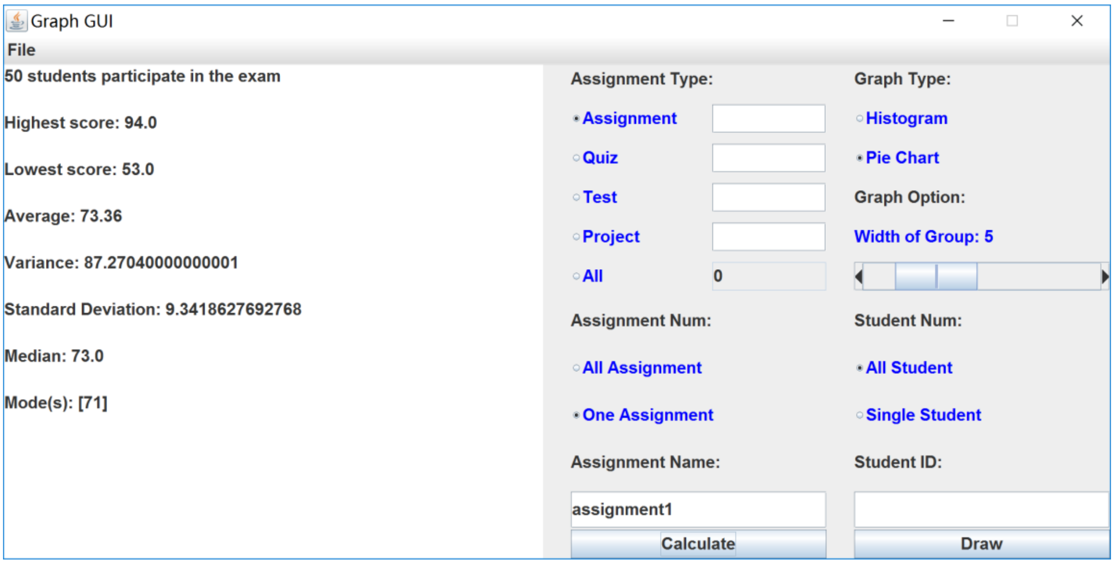
Detail:
Whatever you want to do, draw or calculate, the Graph_GUI will create a GradeGroup object to get the grade file and handle them. If you want the grade of all assignments and all students, it will call getAllGradeFile(). If you want the grade of a category of assignments and all students, it will call getMultiGradeFile(). If you want the grade of one assignment and all students, it will call getSingleGradeFile(). If you want the grade of all assignments and single student, it will call getStudentFile(). Then if you want to draw the graph, that means you need the numbers of students in each grade group, so it will call cal() to calculate, then it will return the ArrayList of number to Drawing_GUI to draw. If you want to calculate the summary statistics, it will return the ArrayList of grade to SummaryStat to calculate. Besides, for single student and all assignments, there is no need to draw the graph and calculate the summary statistics, so it will return this student's grade to Graph_GUI.



3. Example

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

Graph_GUI:



50 students participate in the exam

Highest score: 94.0

Lowest score: 53.0

Average: 73.36

Variance: 87.27040000000001

Standard Deviation: 9.3418627692768

Median: 73.0

Mode(s): [71]

Assignment Type:

● Assignment
○ Quiz
○ Test
○ Project
○ All 0

Graph Type:

○ Histogram
● Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

○ All Assignment
● One Assignment

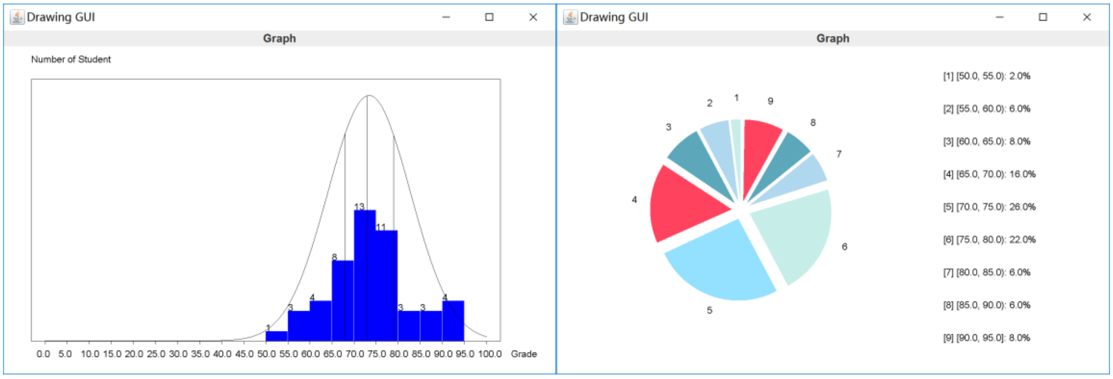
Student Num:

● All Student
○ Single Student

Assignment Name: assignment1

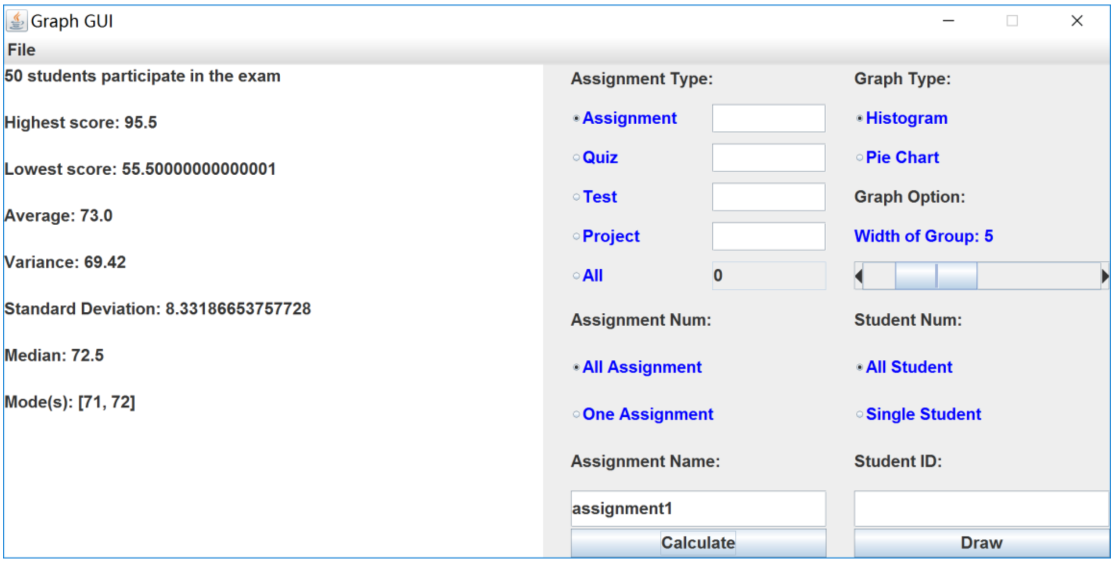
Student ID:

Drawing_GUI:



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

Graph_GUI:



```
50 students participate in the exam

Highest score: 95.5

Lowest score: 55.50000000000001

Average: 73.0

Variance: 69.42

Standard Deviation: 8.33186653757728

Median: 72.5

Mode(s): [71, 72]
```

Assignment Type:

● Assignment
○ Quiz
○ Test
○ Project
○ All 0

Graph Type:

● Histogram
○ Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

● All Assignment
○ One Assignment

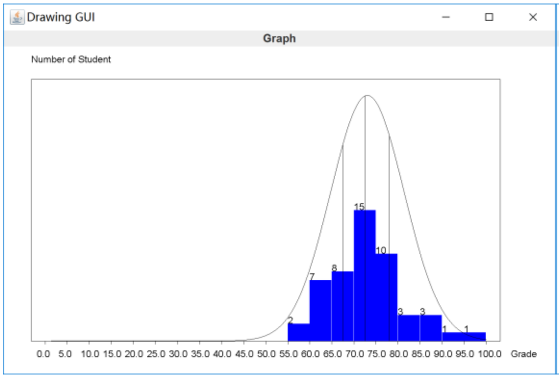
Student Num:

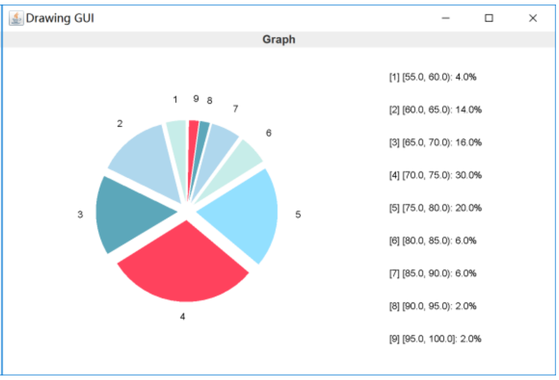
● All Student
○ Single Student

Assignment Name: assignment1

Student ID:

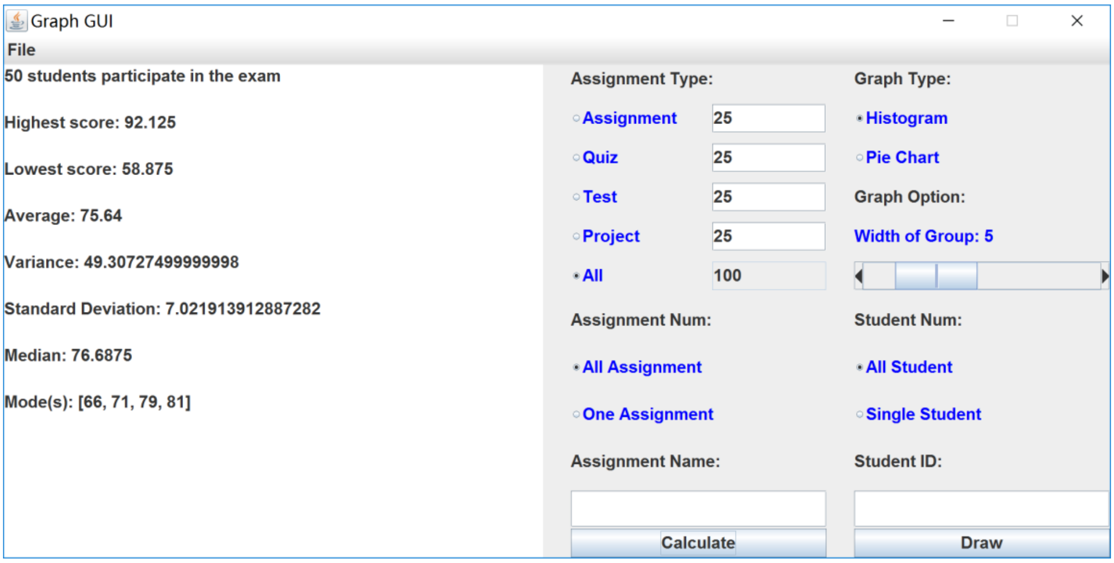
Drawing_GUI:





Case 3:
Assignment Type: All,
Weight: 25%, 25%, 25%, 25%,
Assignment Num: All Assignment,
Student Num: All Student.

Graph_GUI:



Graph GUI

File

50 students participate in the exam

Highest score: 92.125

Lowest score: 58.875

Average: 75.64

Variance: 49.30727499999998

Standard Deviation: 7.021913912887282

Median: 76.6875

Mode(s): [66, 71, 79, 81]

Assignment Type:

○ Assignment 25

○ Quiz 25

○ Test 25

○ Project 25

● All 100

Graph Type:

● Histogram

○ Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

● All Assignment

○ One Assignment

Assignment Name:

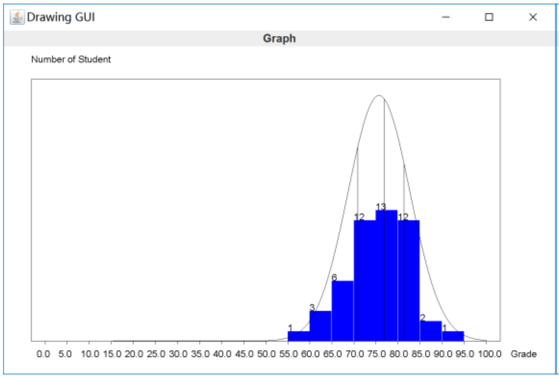
Student Num:

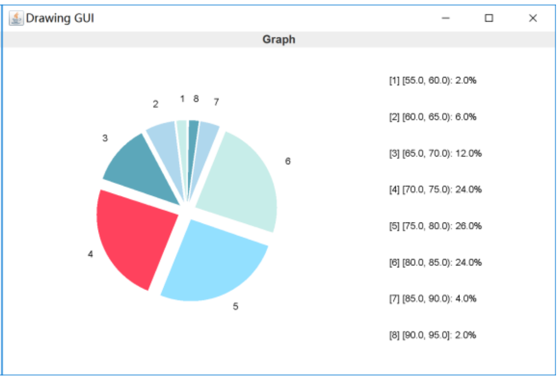
● All Student

○ Single Student

Student ID:

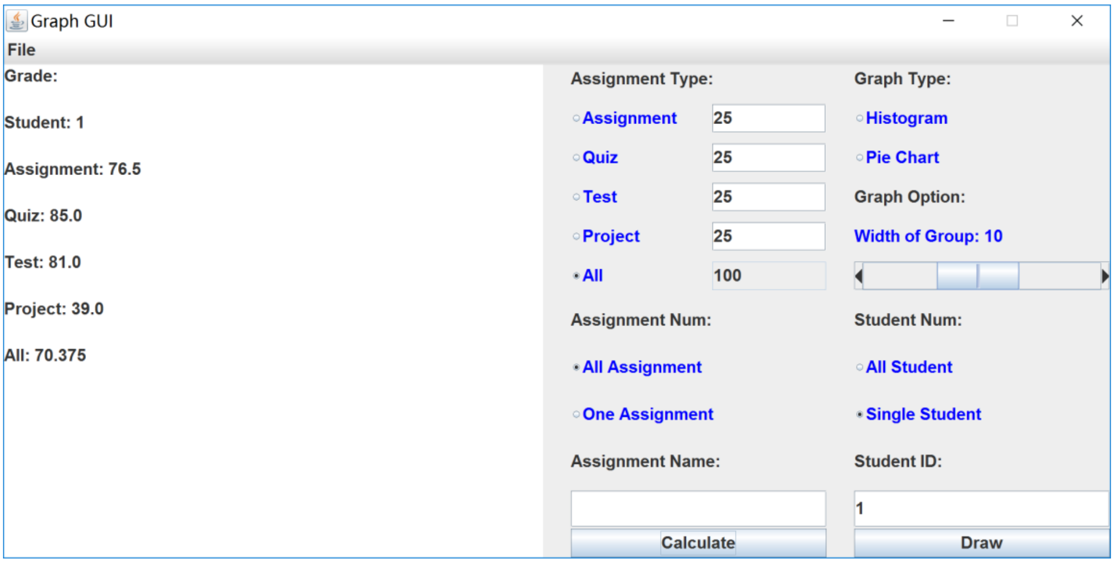
Drawing_GUI:





Case 4:
Assignment Type: All,
Weight: 25%, 25%, 25%, 25%,
Assignment Num: All Assignment,
Student Num: Single Student.

Student ID: 1



Grade:

Student: 1

Assignment: 76.5

Quiz: 85.0

Test: 81.0

Project: 39.0

All: 70.375

Assignment Type:
○ Assignment 25
○ Quiz 25
○ Test 25
○ Project 25
● All 100

Graph Type:
○ Histogram
○ Pie Chart

Graph Option:
Width of Group: 10

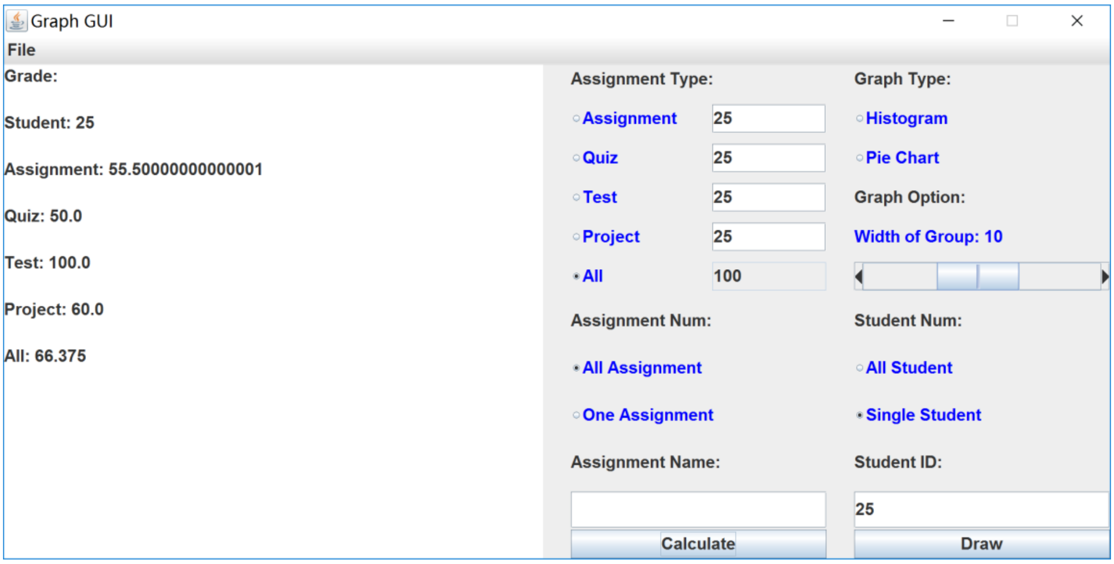
Assignment Num:
● All Assignment
○ One Assignment

Student Num:
○ All Student
● Single Student

Assignment Name:

Student ID: 1

Student ID: 25



Grade:

Student: 25

Assignment: 55.50000000000001

Quiz: 50.0

Test: 100.0

Project: 60.0

All: 66.375

Assignment Type:
○ Assignment 25
○ Quiz 25
○ Test 25
○ Project 25
● All 100

Graph Type:
○ Histogram
○ Pie Chart

Graph Option:
Width of Group: 10

Assignment Num:
● All Assignment
○ One Assignment

Student Num:
○ All Student
● Single Student

Assignment Name:

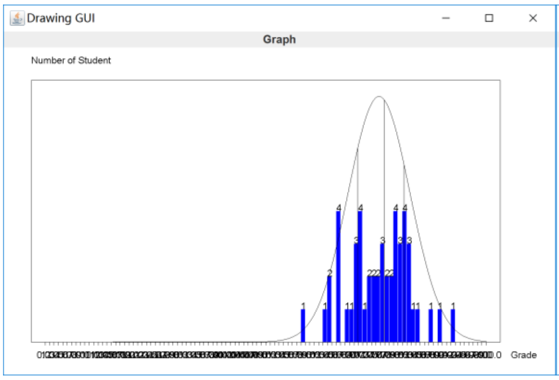
Student ID: 25

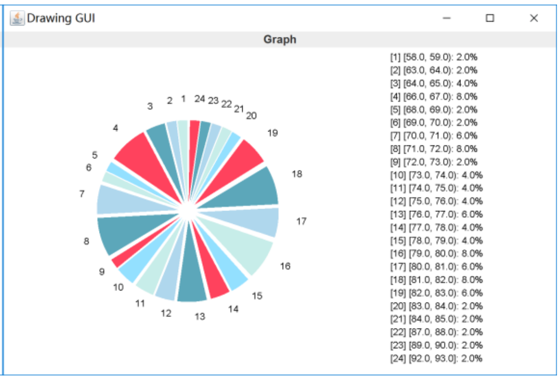
Case 5:

Different Width of Group.

(We can move the scroll bar of the width to get the width of group we want)

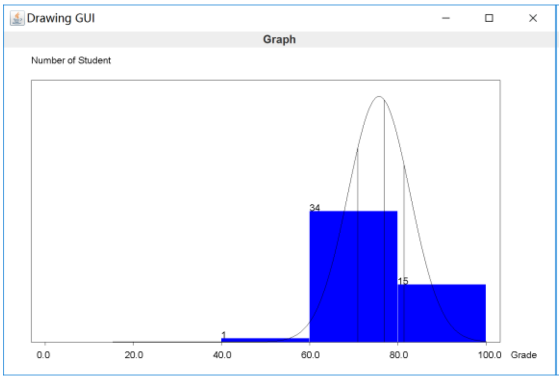
Width: 1

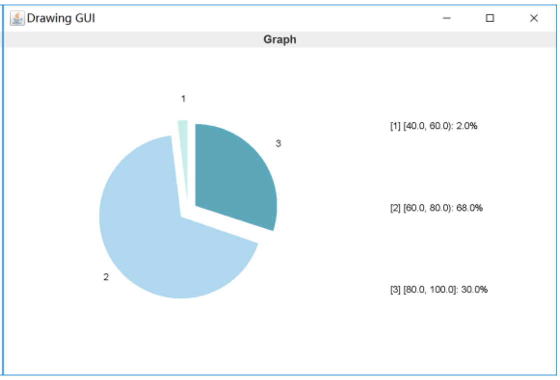




[1] [58.0, 59.0): 2.0%
[2] [63.0, 64.0): 2.0%
[3] [64.0, 65.0): 4.0%
[4] [66.0, 67.0): 8.0%
[5] [68.0, 69.0): 2.0%
[6] [69.0, 70.0): 2.0%
[7] [70.0, 71.0): 6.0%
[8] [71.0, 72.0): 8.0%
[9] [72.0, 73.0): 2.0%
[10] [73.0, 74.0): 4.0%
[11] [74.0, 75.0): 4.0%
[12] [75.0, 76.0): 4.0%
[13] [76.0, 77.0): 6.0%
[14] [77.0, 78.0): 4.0%
[15] [78.0, 79.0): 4.0%
[16] [79.0, 80.0): 8.0%
[17] [80.0, 81.0): 6.0%
[18] [81.0, 82.0): 8.0%
[19] [82.0, 83.0): 6.0%
[20] [83.0, 84.0): 2.0%
[21] [84.0, 85.0): 2.0%
[22] [87.0, 88.0): 2.0%
[23] [89.0, 90.0): 2.0%
[24] [92.0, 93.0]: 2.0%

Width: 20





[1] [40.0, 60.0): 2.0%

[2] [60.0, 80.0): 68.0%

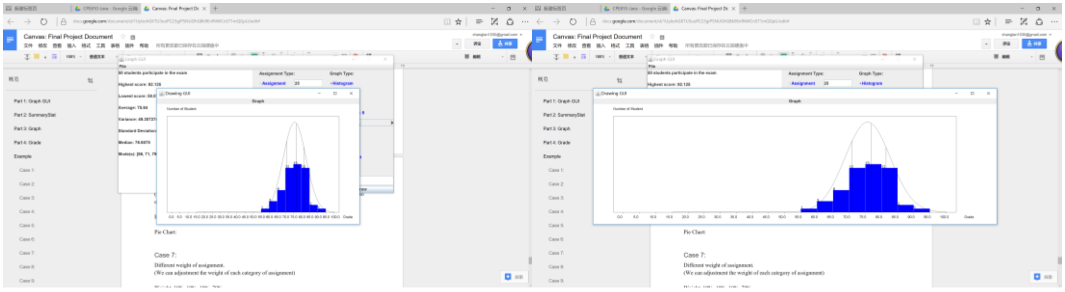
[3] [80.0, 100.0]: 30.0%

Case 6:

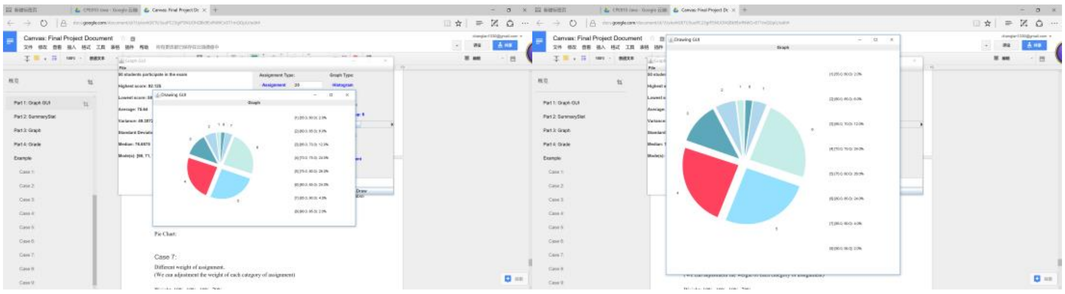
Different Size of Graph.

(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:

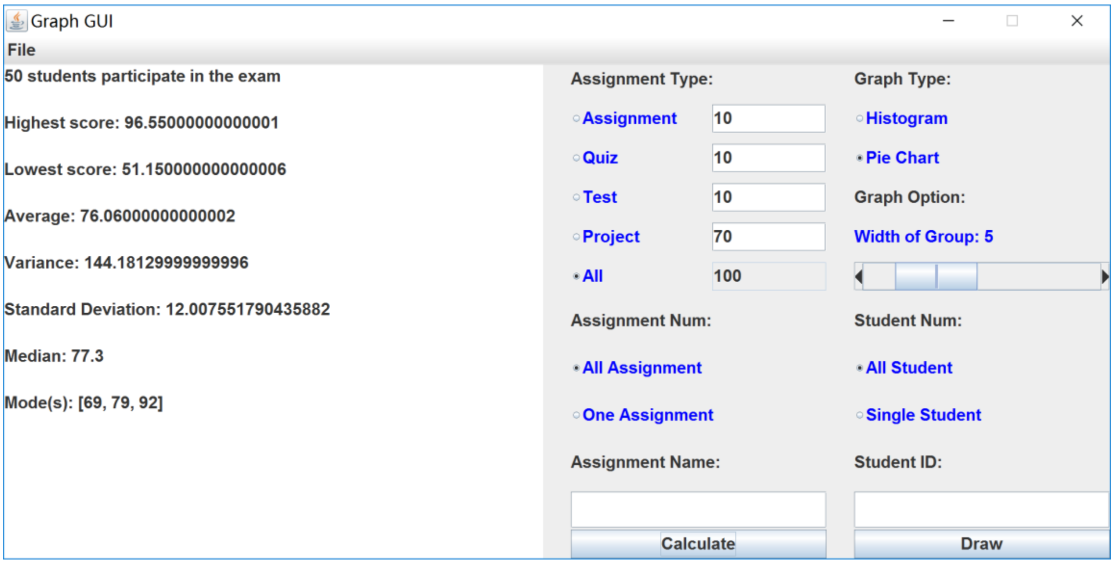


Case 7:

Different weight of assignment.

(We can adjustment the weight of each category of assignment)

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



```
50 students participate in the exam

Highest score: 96.55000000000001

Lowest score: 51.150000000000006

Average: 76.06000000000002

Variance: 144.18129999999996

Standard Deviation: 12.007551790435882

Median: 77.3

Mode(s): [69, 79, 92]
```

Assignment Type:

○ Assignment 10

○ Quiz 10

○ Test 10

○ Project 70

● All 100

Graph Type:

○ Histogram

● Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

● All Assignment

○ One Assignment

Assignment Name:

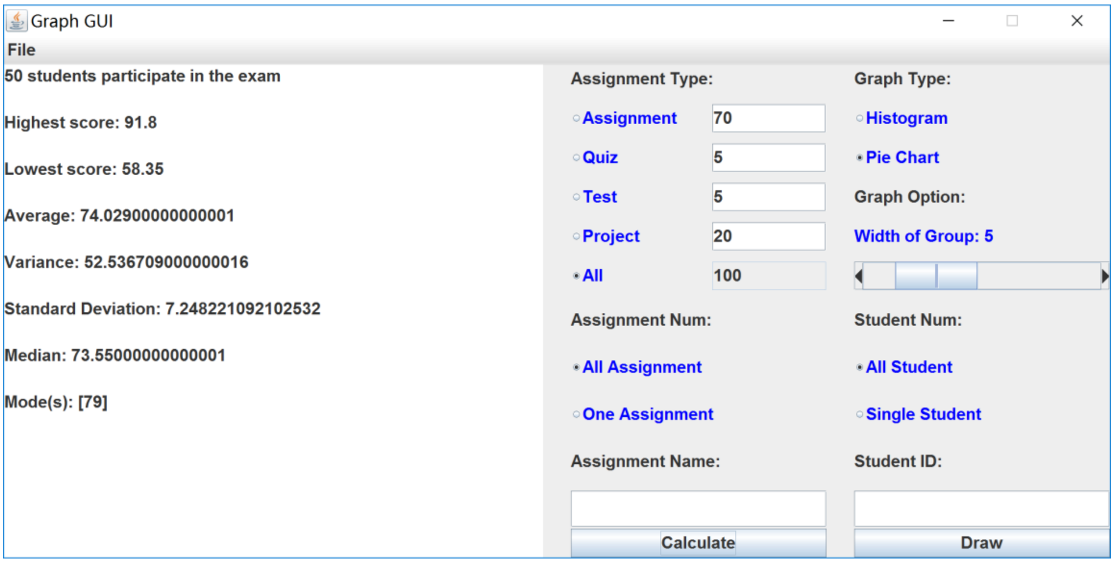
Student Num:

● All Student

○ Single Student

Student ID:

Weight: 70%, 5%, 5%, 20%



```
50 students participate in the exam

Highest score: 91.8

Lowest score: 58.35

Average: 74.02900000000001

Variance: 52.536709000000016

Standard Deviation: 7.248221092102532

Median: 73.55000000000001

Mode(s): [79]
```

Assignment Type:

○ Assignment 70

○ Quiz 5

○ Test 5

○ Project 20

● All 100

Graph Type:

○ Histogram

● Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

● All Assignment

○ One Assignment

Assignment Name:

Student Num:

● All Student

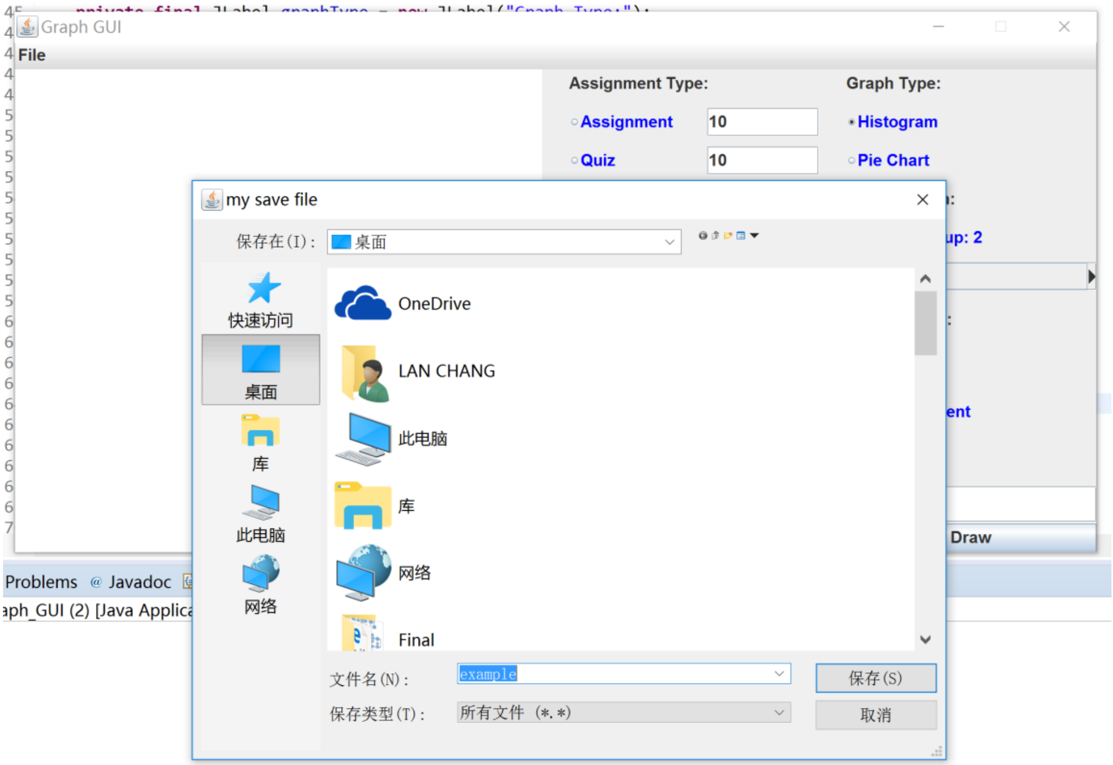
○ Single Student

Student ID:

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.



Assignment Type:
○ Assignment 10
○ Quiz 10

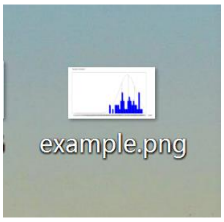
Graph Type:
● Histogram
○ Pie Chart

保存在(I): 桌面

文件名(N): example

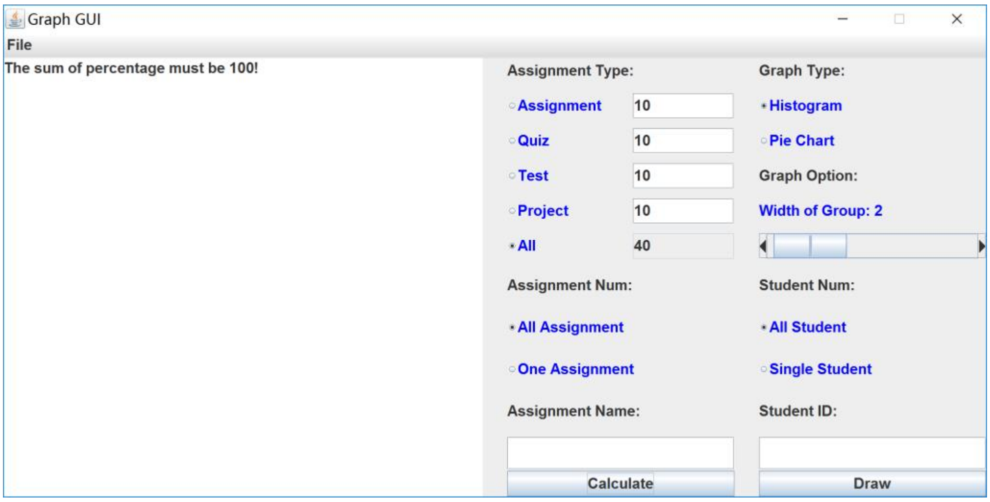
保存类型(T): 所有文件 (*.*)

“example.png” file on the Desktop.



Case 9:
Showing Error.

The sum of percentage is not 100.



The sum of percentage must be 100!

Assignment Type:
○ Assignment 10
○ Quiz 10
○ Test 10
○ Project 10
● All 40

Graph Type:
● Histogram
○ Pie Chart

Graph Option:
Width of Group: 2

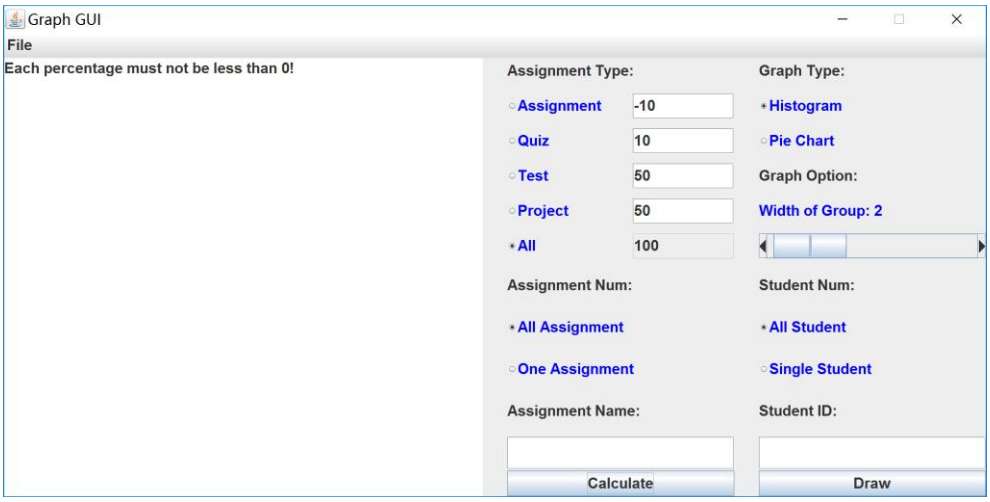
Assignment Num:
● All Assignment
○ One Assignment

Student Num:
● All Student
○ Single Student

Assignment Name:

Student ID:

There is a percentage is less than 0.



Each percentage must not be less than 0!

Assignment Type:
○ Assignment -10
○ Quiz 10
○ Test 50
○ Project 50
● All 100

Graph Type:
● Histogram
○ Pie Chart

Graph Option:
Width of Group: 2

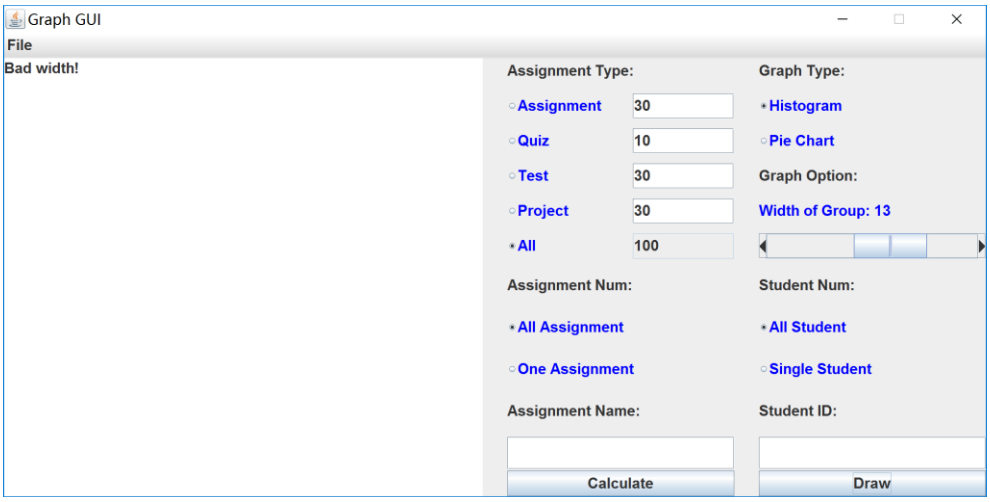
Assignment Num:
● All Assignment
○ One Assignment

Student Num:
● All Student
○ Single Student

Assignment Name:

Student ID:

The width is not good.



Bad width!

Assignment Type:
○ Assignment 30
○ Quiz 10
○ Test 30
○ Project 30
● All 100

Graph Type:
● Histogram
○ Pie Chart

Graph Option:
Width of Group: 13

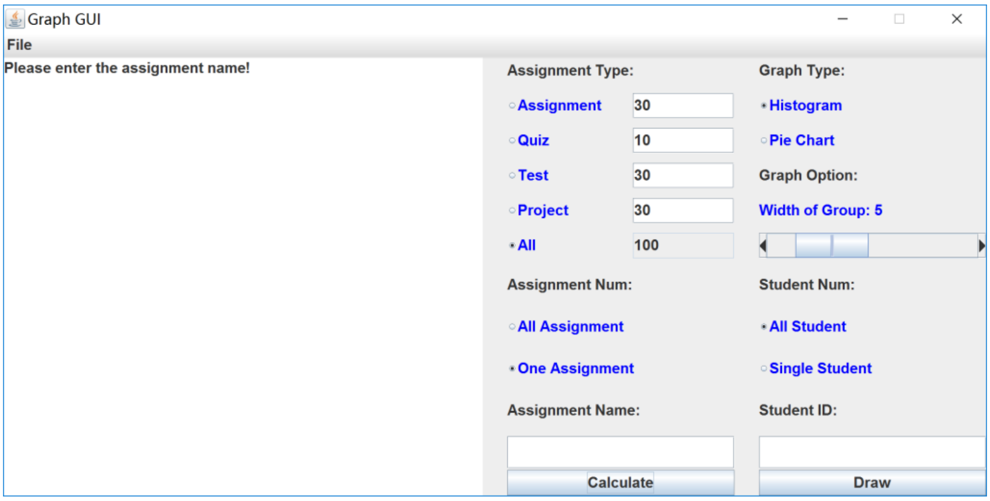
Assignment Num:
● All Assignment
○ One Assignment

Student Num:
● All Student
○ Single Student

Assignment Name:

Student ID:

No assignment name entering.



Please enter the assignment name!

Assignment Type:

○ Assignment 30

○ Quiz 10

○ Test 30

○ Project 30

● All 100

Graph Type:

● Histogram

○ Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

○ All Assignment

● One Assignment

Student Num:

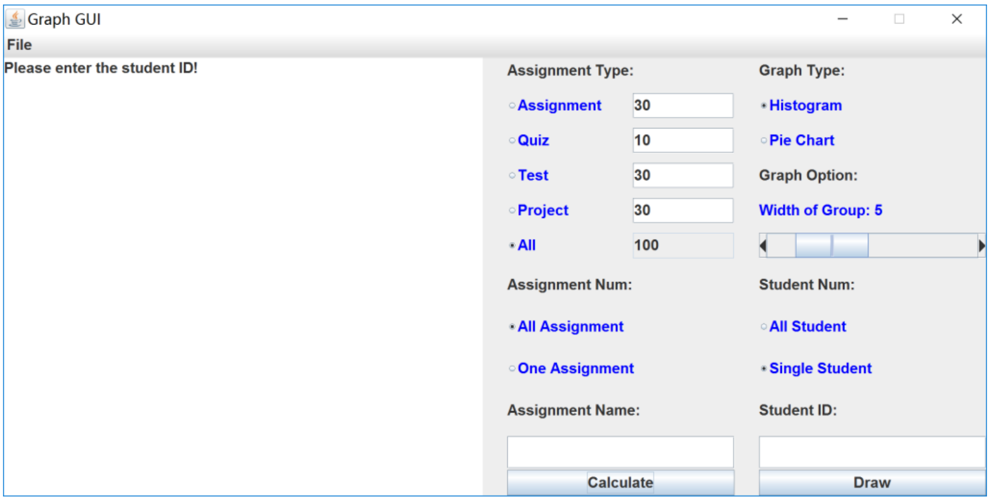
● All Student

○ Single Student

Assignment Name:

Student ID:

No student ID entering.



Please enter the student ID!

Assignment Type:

○ Assignment 30

○ Quiz 10

○ Test 30

○ Project 30

● All 100

Graph Type:

● Histogram

○ Pie Chart

Graph Option:

Width of Group: 5

Assignment Num:

● All Assignment

○ One Assignment

Student Num:

○ All Student

● Single Student

Assignment Name:

Student ID: