Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 2

по курсу «Программирование на языке Java»

на тему «Работа с коллекциями объектов»

Вариант 3

Выполнили

студенты группы 22ВВП1:

Юртаев Д.Ю.

Шатин Д.Д.

Приняли:

Юрова О.В.

Карамышева Н.С.

Пенза 2025

**Цель работы**

изучить библиотеку стандартных коллекций Java Collections Framework, позволяющую хранить различные структуры данных.

**Лабораторное задание**

Модифицировать приложение из предыдущей лабораторной работы, реализовав хранение данных таблицы с использованием библиотеки коллекций. Для этого реализовать класс RecIntegral, способный хранить одну запись таблицы. Для нечетных вариантов в качестве класса-коллекции выбрать ArrayList, для четных - LinkedList. Кроме того, добавить пару кнопок: очистить / заполнить, которые будут очищать таблицу и заполнять ее данными из коллекции соответственно.

|  |  |
| --- | --- |
| Номер варианта | Тип коллекции |
| 3 | ArrayList |

**Описание метода решения задачи**

В данной лабораторной работе была разработана модификация графического интерфейса для работы с данными, хранящимися в таблице, с использованием коллекций. Для хранения записей таблицы был реализован класс RecIntegral, который отвечает за сохранение одной записи с тремя параметрами: границей интеграла (A, B) и шагом (H), а также результатом вычислений для этих параметров.

Для работы с коллекциями был использован класс ArrayList<RecIntegral>, в который добавляются и из которого удаляются данные. Каждая запись добавляется в коллекцию при нажатии кнопки "ADD", а при нажатии кнопки "DELETE" удаляется выбранная строка из таблицы и соответствующая запись из коллекции.

В графическом интерфейсе таблица jTable1, отображает данные из коллекции. В таблице отображаются четыре столбца: «lower limit» (A), «higher limit» (B), «step» (H) и «result». Все данные, введенные пользователем, сохраняются и отображаются в таблице.

Кроме того, были добавлены две дополнительные кнопки:

1. **"CLEAN"** — эта кнопка очищает таблицу, но не затрагивает данные, хранящиеся в коллекции.
2. **"REFILL"** — эта кнопка перезаполняет таблицу данными из коллекции ArrayList, не изменяя саму коллекцию.

В кнопке "PERFORM" реализовано вычисление определенного интеграла для функции cos(x) на основе значений, взятых из таблицы. Для этого происходит считывание данных из строки таблицы в переменные (a, b, h), затем вычисляется интеграл методом трапеций, и результат записывается в таблицу в соответствующую колонку «result».

Таким образом, приложение теперь использует коллекцию ArrayList для хранения данных и позволяет пользователю управлять таблицей через интерфейс с дополнительными функциями очистки и перезаполнения таблицы. Листинг программы находится в приложении 1

**Результаты работы программы**

Для проверки корректности работы программы произвели записи значений, вычисления, а также проверили работу новых добавленных функций с использованием коллекции

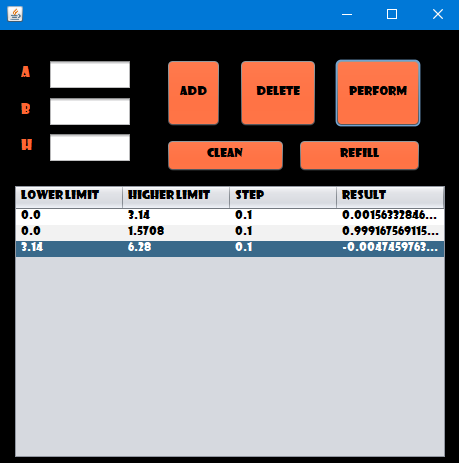


Рисунок 1 – Осуществили записи и произвели вычисления

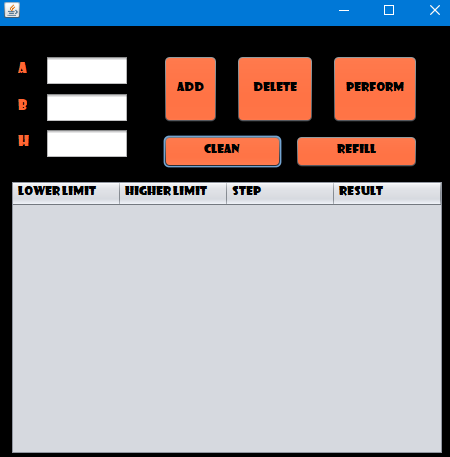


Рисунок 2 – Очистили записи с помощью кнопки «CLEAN»

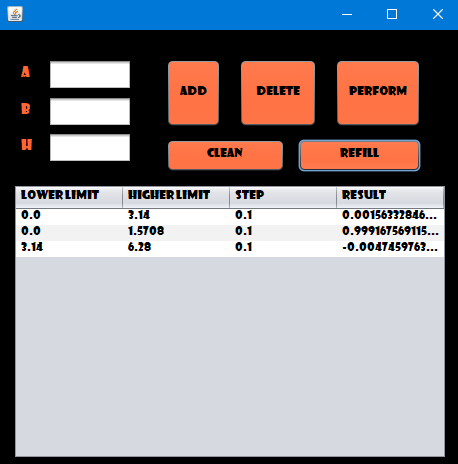


Рисунок 3 – Восстановили записи с помощью кнопки «REFILL»

**Вывод**

В ходе выполнения лабораторной работы была разработана программа, вычисляющая определенный интеграл функции. Были изучены основы разработки приложений, имеющих графический интерфейс пользователя, с использованием библиотеки Swing. Результаты работы программы совпали с ожидаемыми результатами, следовательно, программа работает без ошибок.

**Приложение 1 – Листинг**

**Класс NewJFrame**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template

\*/

package lab1\_java;

import javax.swing.table.DefaultTableModel;

import javax.swing.JOptionPane;

import java.awt.Font;

import javax.swing.table.JTableHeader;

import java.util.ArrayList;

/\*\*

\*

\* @author Ant

\*/

public class NewJFrame extends javax.swing.JFrame {

/\*\*

\* Creates new form NewJFrame

\*/

private ArrayList<RecIntegral> records; // List to store the records

public NewJFrame() {

initComponents();

records = new ArrayList<>(); // Initialize the list

JTableHeader tableHeader = jTable1.getTableHeader();

tableHeader.setFont(new Font("Showcard Gothic", Font.PLAIN, 12));

getContentPane().setBackground(new java.awt.Color(0, 0, 0)); // черный фон

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jTextField1 = new javax.swing.JTextField();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setBackground(new java.awt.Color(255, 204, 51));

jButton1.setBackground(new java.awt.Color(255, 102, 51));

jButton1.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jButton1.setText("ADD");

jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setBackground(new java.awt.Color(255, 102, 51));

jButton2.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jButton2.setText("DELETE");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton3.setBackground(new java.awt.Color(255, 102, 51));

jButton3.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jButton3.setText("PERFORM");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jLabel1.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jLabel1.setForeground(new java.awt.Color(255, 102, 51));

jLabel1.setText("a");

jLabel2.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jLabel2.setForeground(new java.awt.Color(255, 102, 51));

jLabel2.setText("b");

jLabel3.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jLabel3.setForeground(new java.awt.Color(255, 102, 51));

jLabel3.setText("h");

jTextField1.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jTextField2.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jTextField2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField2ActionPerformed(evt);

}

});

jTextField3.setFont(new java.awt.Font("Showcard Gothic", 0, 14)); // NOI18N

jTextField3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField3ActionPerformed(evt);

}

});

jTable1.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"lower limit", "higher limit", "step", "result"

}

) {

boolean[] canEdit = new boolean [] {

true, true, true, false

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane1.setViewportView(jTable1);

jButton4.setBackground(new java.awt.Color(255, 102, 51));

jButton4.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jButton4.setText("CLEAN");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jButton5.setBackground(new java.awt.Color(255, 102, 51));

jButton5.setFont(new java.awt.Font("Showcard Gothic", 0, 12)); // NOI18N

jButton5.setText("REFILL");

jButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton5ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addContainerGap(14, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 434, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(14, 14, 14))

.addGroup(layout.createSequentialGroup()

.addGap(22, 22, 22)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jLabel2)

.addGap(18, 18, 18))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel1)

.addGap(17, 17, 17)))

.addGroup(layout.createSequentialGroup()

.addComponent(jLabel3)

.addGap(16, 16, 16)))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jTextField3, javax.swing.GroupLayout.DEFAULT\_SIZE, 84, Short.MAX\_VALUE)

.addComponent(jTextField1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, 84, Short.MAX\_VALUE)

.addComponent(jTextField2, javax.swing.GroupLayout.Alignment.LEADING))

.addGap(34, 34, 34)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addComponent(jButton1)

.addGap(18, 18, 18)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 78, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(jButton3))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton4, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton5, javax.swing.GroupLayout.PREFERRED\_SIZE, 123, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(0, 0, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)

.addComponent(jButton2, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel1)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addComponent(jButton1, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jButton3, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(5, 5, 5)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel3)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton4, javax.swing.GroupLayout.DEFAULT\_SIZE, 33, Short.MAX\_VALUE)

.addComponent(jButton5, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)))

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 275, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

pack();

}// </editor-fold>

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Получаем модель таблицы

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

// Получаем индекс выделенной строки

int selectedRow = jTable1.getSelectedRow();

// Проверяем, выбрана ли строка

if (selectedRow != -1) {

// Удаляем выбранную строку из списка

records.remove(selectedRow);

// Удаляем выбранную строку из таблицы

model.removeRow(selectedRow);

} else {

// Если строка не выбрана, выводим сообщение

JOptionPane.showMessageDialog(this, "Выберите строку для удаления", "Ошибка", JOptionPane.WARNING\_MESSAGE);

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

// Получаем модель таблицы

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

// Получаем индекс выбранной строки

int selectedRow = jTable1.getSelectedRow();

// Проверяем, выбрана ли строка

if (selectedRow != -1) {

try {

// получаем значение из листа для выбранной строки

RecIntegral record = records.get(selectedRow);

record.calculate();

// обновляем таблицу с посчитанным значением

model.setValueAt(record.getResult(), selectedRow, 3);

} catch (Exception e) {

JOptionPane.showMessageDialog(this, "Ошибка в вычислениях", "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(this, "Выберите строку для вычисления", "Ошибка", JOptionPane.WARNING\_MESSAGE);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String niz = jTextField1.getText();

String verh = jTextField2.getText();

String shag = jTextField3.getText();

try {

double a = Double.parseDouble(niz);

double b = Double.parseDouble(verh);

double h = Double.parseDouble(shag);

// Add a new record to the list

RecIntegral record = new RecIntegral(a, b, h);

records.add(record);

// Add a new row to the table

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.addRow(new Object[]{a, b, h, ""});

// Clear the text fields

jTextField1.setText("");

jTextField2.setText("");

jTextField3.setText("");

} catch (NumberFormatException e) {

JOptionPane.showMessageDialog(this, "Ошибка в вводе чисел", "Ошибка", JOptionPane.ERROR\_MESSAGE);

}

}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0); // удаляем все строки

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0); // чистим существующие строки

for (RecIntegral record : records) {

model.addRow(new Object[]{record.getA(), record.getB(), record.getH(), record.getResult()});

}

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new NewJFrame().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JButton jButton5;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

// End of variables declaration

}

**Класс RecIntegral**

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package lab1\_java;

/\*\*

\*

\* @author Ant

\*/

public class RecIntegral {

private double a; // lower limit

private double b; // upper limit

private double h; // step

private double result; // result of the calculation

// Constructor

public RecIntegral(double a, double b, double h) {

this.a = a;

this.b = b;

this.h = h;

this.result = 0.0; // initially, no result

}

// Getters and Setters

public double getA() {

return a;

}

public void setA(double a) {

this.a = a;

}

public double getB() {

return b;

}

public void setB(double b) {

this.b = b;

}

public double getH() {

return h;

}

public void setH(double h) {

this.h = h;

}

public double getResult() {

return result;

}

public void setResult(double result) {

this.result = result;

}

// Method to calculate the result

public void calculate() {

double sum = 0.0;

double x = a;

while (x + h <= b) {

sum += (h / 2.0) \* (Math.cos(x) + Math.cos(x + h));

x += h;

}

if (x < b) {

double last\_h = b - x;

sum += (last\_h / 2.0) \* (Math.cos(x) + Math.cos(b));

}

this.result = sum; // Store the calculated result

}

}