

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **«СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | 09.02.07 Информационные системы и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

.

**Отчет по лабораторной работе № 5**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **на тему** | |  | Использование меню | | | | |
|  | | | | | |  | (наименование темы) |
|  |  | |  | | | | |
| **по дисциплине** | | | |  | Разработка мобильных приложений | | |
|  | | | | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Николаев Павел Юрьевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-301прог |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва, 2025**

Лабораторная работа №5

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.text.DecimalFormat;

import java.util.LinkedHashMap;

import java.util.Map;

import javax.swing.\*;

public class Main extends JFrame {

    private static final long serialVersionUID = 1L;

    // Map: unit name -> meters in one unit (1 unit = value meters)

    private static final Map<String, Double> UNIT\_TO\_METERS = new LinkedHashMap<>();

    static {

        UNIT\_TO\_METERS.put("миллиметр (mm)", 0.001);

        UNIT\_TO\_METERS.put("сантиметр (cm)", 0.01);

        UNIT\_TO\_METERS.put("дюйм (in)", 0.0254);

        UNIT\_TO\_METERS.put("фут (ft)", 0.3048);

        UNIT\_TO\_METERS.put("ярд (yd)", 0.9144);

        UNIT\_TO\_METERS.put("метр (m)", 1.0);

        UNIT\_TO\_METERS.put("километр (km)", 1000.0);

        UNIT\_TO\_METERS.put("миля (mi)", 1609.344); // statute mile

        UNIT\_TO\_METERS.put("морская миля (nmi)", 1852.0);

        UNIT\_TO\_METERS.put("аршин (arshin)", 0.7112);

        UNIT\_TO\_METERS.put("верста (versta)", 1066.8);

    }

    private JComboBox<String> fromCombo;

    private JComboBox<String> toCombo;

    private JTextField inputField;

    private JLabel resultLabel;

    private JTable tableAll;

    private DecimalFormat df = new DecimalFormat("#.##########");

    public Main() {

        setTitle("Конвертер длин — Java Swing");

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        setSize(750, 450);

        setLocationRelativeTo(null);

        initUI();

    }

    private void initUI() {

        JPanel topPanel = new JPanel(new FlowLayout(FlowLayout.LEFT, 10, 10));

        inputField = new JTextField();

        inputField.setPreferredSize(new Dimension(120, 26));

        topPanel.add(new JLabel("Значение:"));

        topPanel.add(inputField);

        fromCombo = new JComboBox<>(UNIT\_TO\_METERS.keySet().toArray(new String[0]));

        toCombo = new JComboBox<>(UNIT\_TO\_METERS.keySet().toArray(new String[0]));

        fromCombo.setPreferredSize(new Dimension(220, 26));

        toCombo.setPreferredSize(new Dimension(220, 26));

        topPanel.add(new JLabel("Из:"));

        topPanel.add(fromCombo);

        topPanel.add(new JLabel("В:"));

        topPanel.add(toCombo);

        JButton convertBtn = new JButton("Перевести");

        JButton swapBtn = new JButton("Поменять местами");

        JButton clearBtn = new JButton("Очистить");

        topPanel.add(convertBtn);

        topPanel.add(swapBtn);

        topPanel.add(clearBtn);

        add(topPanel, BorderLayout.NORTH);

        JPanel centerPanel = new JPanel(new BorderLayout(10, 10));

        resultLabel = new JLabel("Результат: ");

        resultLabel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

        centerPanel.add(resultLabel, BorderLayout.NORTH);

        String[] columns = {"Единица", "Значение"};

        Object[][] data = new Object[UNIT\_TO\_METERS.size()][2];

        int idx = 0;

        for (String u : UNIT\_TO\_METERS.keySet()) {

            data[idx][0] = u;

            data[idx][1] = "";

            idx++;

        }

        tableAll = new JTable(data, columns);

        JScrollPane scroll = new JScrollPane(tableAll);

        centerPanel.add(scroll, BorderLayout.CENTER);

        add(centerPanel, BorderLayout.CENTER);

        JPanel bottom = new JPanel(new GridLayout(2, 1));

        bottom.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));

        bottom.add(new JLabel("Введите числовое значение и выберите единицы. Нажмите 'Перевести' — " +

                "в таблице покажутся значения для всех единиц."));

        bottom.add(new JLabel("Примечание: аршин ≈ 0.7112 m, верста ≈ 1066.8 m."));

        add(bottom, BorderLayout.SOUTH);

        convertBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                doConvert();

            }

        });

        swapBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                int f = fromCombo.getSelectedIndex();

                int t = toCombo.getSelectedIndex();

                fromCombo.setSelectedIndex(t);

                toCombo.setSelectedIndex(f);

            }

        });

        clearBtn.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                inputField.setText("");

                resultLabel.setText("Результат: ");

                clearTable();

            }

        });

        inputField.addActionListener(new ActionListener() {

            @Override

            public void actionPerformed(ActionEvent e) {

                doConvert();

            }

        });

    }

    private void doConvert() {

        String text = inputField.getText().trim();

        if (text.isEmpty()) {

            JOptionPane.showMessageDialog(this, "Введите значение для конвертации.", "Ошибка", JOptionPane.WARNING\_MESSAGE);

            return;

        }

        double value;

        try {

            value = Double.parseDouble(text.replace(',', '.'));

        } catch (NumberFormatException ex) {

            JOptionPane.showMessageDialog(this, "Неверный формат числа.", "Ошибка", JOptionPane.ERROR\_MESSAGE);

            return;

        }

        String fromUnit = (String) fromCombo.getSelectedItem();

        String toUnit = (String) toCombo.getSelectedItem();

        double result = convert(value, fromUnit, toUnit);

        resultLabel.setText("Результат: " + df.format(value) + " " + unitShort(fromUnit) + " = " + df.format(result) + " " + unitShort(toUnit));

        fillAllTable(value, fromUnit);

    }

    private void clearTable() {

        for (int r = 0; r < tableAll.getRowCount(); r++) {

            tableAll.setValueAt("", r, 1);

        }

    }

    private void fillAllTable(double value, String fromUnit) {

        int r = 0;

        for (String targetUnit : UNIT\_TO\_METERS.keySet()) {

            double v = convert(value, fromUnit, targetUnit);

            tableAll.setValueAt(df.format(v), r, 1);

            r++;

        }

    }

    private static String unitShort(String longName) {

        int i1 = longName.indexOf('(');

        int i2 = longName.indexOf(')');

        if (i1 >= 0 && i2 > i1) {

            return longName.substring(i1 + 1, i2);

        }

        return longName;

    }

    public static double convert(double value, String fromUnit, String toUnit) {

        if (!UNIT\_TO\_METERS.containsKey(fromUnit) || !UNIT\_TO\_METERS.containsKey(toUnit)) {

            throw new IllegalArgumentException("Unknown unit");

        }

        double factorFrom = UNIT\_TO\_METERS.get(fromUnit);

        double factorTo = UNIT\_TO\_METERS.get(toUnit);

        double meters = value \* factorFrom;

        return meters / factorTo;

    }

    public static void main(String[] args) {

        try {

            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

        } catch (Exception ignored) {}

        SwingUtilities.invokeLater(new Runnable() {

            @Override

            public void run() {

                Main frame = new Main();

                frame.setVisible(true);

            }

        });

    }

}

