КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. А.Н.Туполева - КАИ

(КНИТУ-КАИ)

Отделение среднего профессионального образования института

Компьютерных технологий и защиты информации

«Колледж информационных технологий»

Лабораторная работа №2

по предмету:

Наладчик технологического оборудования

Выполнил:

студент 4 курса

Группы 4441

Зарипов Ислам Радифович

Казань 2019 г.

**Цель работы:**

Разработка программы позволяющий пользователю создавать, редактировать, сохранять текстовые файлы.

**Решение:**

Были установлены слушатели на кнопки «Новый», «Открыть», «Сохранить», «Закрыть». При нажатии на копку «Новый» в памяти инициализируется пустой файл, открывается возможность редактировать файл (см. Рис. 2). Если было вызвано событие открытия файла, то отображается окно выбора файла (см. Рис. 3), затем – содержание файла можно редактировать в текстовом поле программы. После изменения, файл можно сохранить, нажав на кнопку «Сохранить». Так же файл можно закрыть нажатие кнопки «Закрыть». Предусмотрена проверка изменения файла при закрытии файла (см. Рис. 4).

**Скриншоты работы программы на Java:**

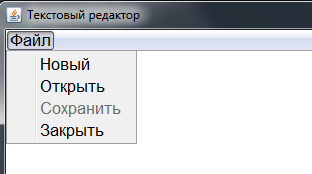
****

Рис. 1. Контекстное меню программы

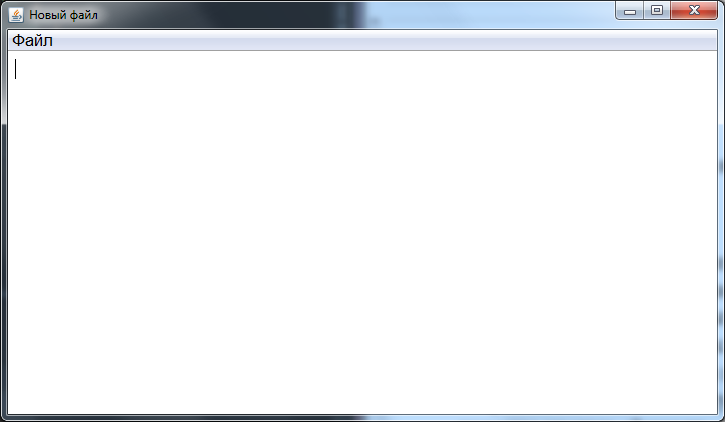


Рис. 2. Окно редактирования нового файла

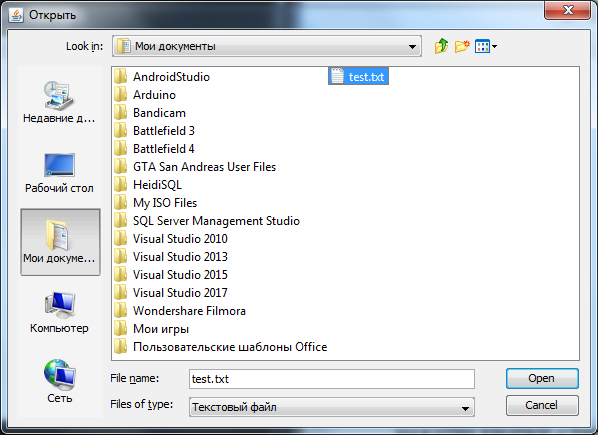


Рис. 3. Окно выбора файла для открытия

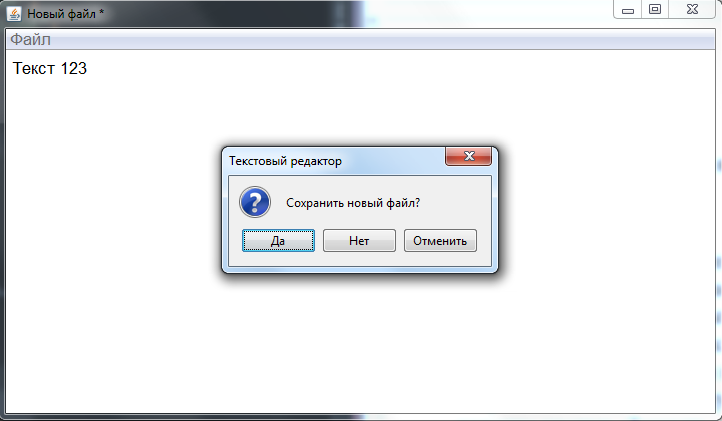


Рис. 4. Окно подтверждения «сохранить или нет»

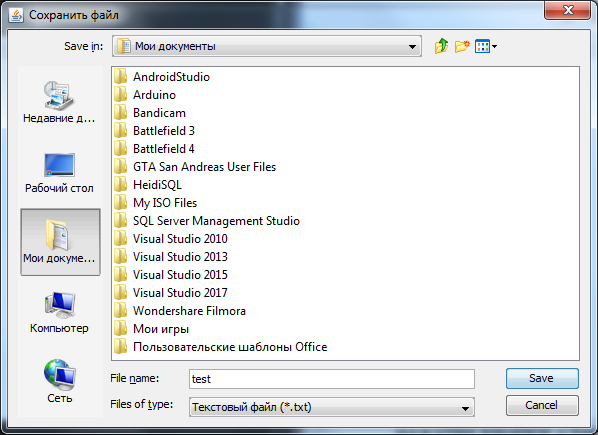


Рис. 5. Окно сохранения нового файла

**Вывод:**

В ходе лабораторной работы была разработана программа, позволяющая пользователю создавать, редактировать, сохранять текстовые файлы.

**Приложение**

**Файл Main.java:**

package ru.inkrot.kit.texteditor;

import javax.swing.\*;

public class Main {

public Main() {

Screen screen = new Screen();

screen.setVisible(true);

}

public static void main(String[] args) throws Exception {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

new Main();

}

}

**Файл Screen.java:**

package ru.inkrot.kit.texteditor;

import javax.swing.\*;

import javax.swing.filechooser.FileNameExtensionFilter;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

public class Screen extends JFrame implements ActionListener {

public final String ACTION\_NEW = "ACTION\_NEW";

public final String ACTION\_OPEN = "ACTION\_OPEN";

public final String ACTION\_SAVE = "ACTION\_SAVE";

public final String ACTION\_CLOSE = "ACTION\_CLOSE";

public String loadedFileText = null;

public boolean edited = false;

public static File currentFile = null;

private JTextArea textArea;

private JMenuItem saveBtn;

public Screen() {

setDefaultCloseOperation(WindowConstants.DO\_NOTHING\_ON\_CLOSE);

setLayout(new BorderLayout());

setSize(725, 422);

setLocationRelativeTo(null);

initGui();

closeEdit();

}

private void initGui() {

UIManager.put("OptionPane.cancelButtonText", "Отменить");

UIManager.put("OptionPane.yesButtonText", "Да");

UIManager.put("OptionPane.noButtonText", "Нет");

addWindowListener(new WindowListener() {

@Override

public void windowOpened(WindowEvent e) {}

@Override

public void windowClosing(WindowEvent e) {

if (checkIsSaved()) System.exit(0);

}

@Override

public void windowClosed(WindowEvent e) {}

@Override

public void windowIconified(WindowEvent e) {}

@Override

public void windowDeiconified(WindowEvent e) {}

@Override

public void windowActivated(WindowEvent e) {}

@Override

public void windowDeactivated(WindowEvent e) {}

});

Font font = new Font("Arial", 0, 17);

JMenuBar menuBar = new JMenuBar();

JMenu fileMenu = new JMenu("Файл");

fileMenu.setFont(font);

JMenuItem newBtn = new JMenuItem("Новый");

newBtn.setFont(font);

newBtn.addActionListener(this);

newBtn.setActionCommand(ACTION\_NEW);

fileMenu.add(newBtn);

JMenuItem openBtn = new JMenuItem("Открыть");

openBtn.setFont(font);

openBtn.addActionListener(this);

openBtn.setActionCommand(ACTION\_OPEN);

fileMenu.add(openBtn);

saveBtn = new JMenuItem("Сохранить");

saveBtn.setFont(font);

saveBtn.addActionListener(this);

saveBtn.setActionCommand(ACTION\_SAVE);

fileMenu.add(saveBtn);

JMenuItem closeBtn = new JMenuItem("Закрыть");

closeBtn.setFont(font);

closeBtn.addActionListener(this);

closeBtn.setActionCommand(ACTION\_CLOSE);

fileMenu.add(closeBtn);

textArea = new JTextArea();

textArea.setFont(font);

textArea.setBorder(BorderFactory.createCompoundBorder(

textArea.getBorder(),

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

textArea.setBounds(0, 0, 700, 400);

textArea.addKeyListener(new KeyListener() {

@Override

public void keyTyped(KeyEvent e) {}

@Override

public void keyPressed(KeyEvent e) {

}

@Override

public void keyReleased(KeyEvent e) {

if (loadedFileText != null) {

if (! loadedFileText.equals(textArea.getText())) {

edited = true;

if (currentFile != null) Screen.this.setTitle(currentFile.getAbsolutePath() + " \*");

else Screen.this.setTitle("Новый файл \*");

} else {

edited = false;

if (currentFile != null) Screen.this.setTitle(currentFile.getAbsolutePath());

else Screen.this.setTitle("Новый файл");

}

}

}

});

add(textArea, BorderLayout.CENTER);

menuBar.add(fileMenu);

setJMenuBar(menuBar);

}

@Override

public void actionPerformed(ActionEvent e) {

String action = e.getActionCommand();

if (action.equals(ACTION\_NEW)) {

if (! checkIsSaved()) return;

currentFile = null;

setTitle("Новый файл");

textArea.setText("");

edited = false;

loadedFileText = "";

openEdit();

} else if (action.equals(ACTION\_OPEN)) {

if (! checkIsSaved()) return;

JFileChooser chooser = new JFileChooser();

chooser.setDialogTitle("Открыть");

FileNameExtensionFilter filter = new FileNameExtensionFilter("Текстовый файл", "txt", "text");

chooser.addChoosableFileFilter(filter);

chooser.setAcceptAllFileFilterUsed(false);

int returnVal = chooser.showOpenDialog(this);

if (returnVal == JFileChooser.APPROVE\_OPTION) {

if (! chooser.getSelectedFile().exists()) {

JOptionPane.showMessageDialog(null, "Файл не найден");

return;

}

currentFile = chooser.getSelectedFile();

setTitle(chooser.getSelectedFile().getAbsolutePath());

try {

BufferedReader reader = new BufferedReader(new FileReader(currentFile));

String line;

StringBuilder sb = new StringBuilder();

while ((line = reader.readLine()) != null) sb.append(line).append('\n');

String text = sb.toString().substring(0, sb.length() - 1);

textArea.setText(text);

loadedFileText = text;

openEdit();

} catch (IOException e1) {

e1.printStackTrace();

}

}

} else if (action.equals(ACTION\_SAVE)) {

saveCurrentFile();

} else if (action.equals(ACTION\_CLOSE)) {

if (! checkIsSaved()) return;

textArea.setText("");

edited = false;

closeEdit();

}

}

public void closeEdit() {

setTitle("Текстовый редактор");

saveBtn.setEnabled(false);

textArea.setEnabled(false);

}

public void openEdit() {

saveBtn.setEnabled(true);

textArea.setEnabled(true);

}

// return: true - продолжить; false - отмена

private boolean checkIsSaved()

{

if (! edited) return true;

String what = currentFile == null ? "новый файл" : "\"" + currentFile.getAbsolutePath() + "\"";

int result = JOptionPane.showConfirmDialog(null, "Сохранить " + what + "?", "Текстовый редактор", JOptionPane.YES\_NO\_CANCEL\_OPTION);

if (result == JOptionPane.YES\_OPTION) {

saveCurrentFile();

return true;

} else if (result == JOptionPane.NO\_OPTION) {

return true;

} else return false;

}

public void saveCurrentFile() {

if (currentFile != null) {

writeFile(currentFile, textArea.getText());

setTitle(currentFile.getAbsolutePath());

edited = false;

} else {

JFileChooser fileChooser = new JFileChooser();

fileChooser.setDialogTitle("Сохранить файл");

FileNameExtensionFilter filter = new FileNameExtensionFilter("Текстовый файл (\*.txt)", "txt", "text");

fileChooser.setAcceptAllFileFilterUsed(false);

fileChooser.addChoosableFileFilter(filter);

int userSelection = fileChooser.showSaveDialog(this);

if (userSelection == JFileChooser.APPROVE\_OPTION) {

File selectedFile = fileChooser.getSelectedFile();

String path = selectedFile.getPath();

currentFile = new File(path.endsWith(".txt") ? path : path + ".txt");

writeFile(currentFile, textArea.getText());

setTitle(currentFile.getAbsolutePath());

edited = false;

}

}

}

public void writeFile(File file, String text) {

try (PrintWriter out = new PrintWriter(file)) {

out.println(text);

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

}