Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа № 2

По дисциплине ГИИС

Тема: «Разработка приложения «Адресная книга»»

Выполнил:

Студент 4 курса

Группы ИИ-16 (2)

Пешко А.С.

Проверил:

Михно Е.В.

Брест, 2021

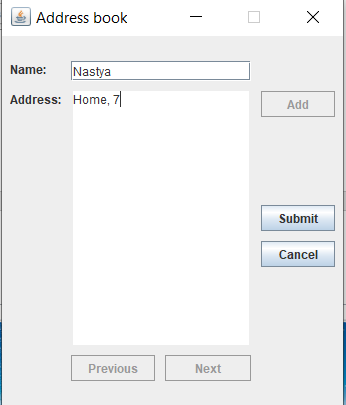
**Цель работы:**  Изучить базовые компоненты средства разработки графического интерфейса.

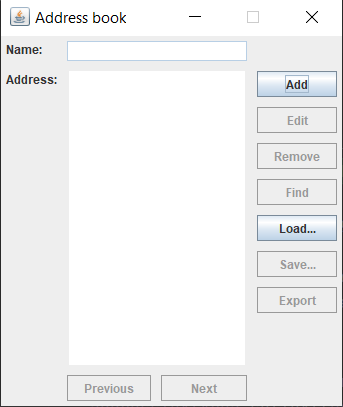
**Ход работы:**

**Задание:** Выполнить последовательно разработку приложения «Адресная книга», согласно учебному пособию.

public class AddressBook extends JFrame {  
  
 ArrayList<User> userBook;  
 Integer currentIndex = 0;  
 Integer mode = null;  
 JLabel nameLabel = new JLabel("Name:");  
 JLabel addressLabel = new JLabel("Address:");  
 JButton addButton = new JButton("Add");  
 JButton editButton = new JButton("Edit");  
 JButton removeButton = new JButton("Remove");  
 JButton findButton = new JButton("Find");  
 JButton loadButton = new JButton("Load...");  
 JButton saveButton = new JButton("Save...");  
 JButton exportButton = new JButton("Export");  
 JButton nextButton = new JButton("Next");  
 JButton previousButton = new JButton("Previous");  
 JButton submitButton = new JButton("Submit");  
 JButton cancelButton = new JButton("Cancel");  
 JTextField nameTextField = new JTextField(16);  
 JTextArea addressTextArea = new JTextArea(1, 16);  
 GridBagConstraints gbc = new GridBagConstraints();  
  
 public AddressBook() {  
 JPanel p = new JPanel();  
 p.setLayout(new GridBagLayout());  
 userBook = new ArrayList<>();  
 addressTextArea.setLineWrap(true);  
 addressTextArea.setWrapStyleWord(true);  
 addressTextArea.setEnabled(false);  
 nameTextField.setEnabled(false);  
 addButton.addActionListener(e -> {  
 nameTextField.setEnabled(true);  
 addressTextArea.setEnabled(true);  
 nameTextField.setText("");  
 addressTextArea.setText("");  
 mode = 0;  
 addButton.setEnabled(false);  
 editButton.setVisible(false);  
 removeButton.setVisible(false);  
 findButton.setVisible(false);  
 loadButton.setVisible(false);  
 saveButton.setVisible(false);  
 exportButton.setVisible(false);  
 nextButton.setEnabled(false);  
 previousButton.setEnabled(false);  
 layout(p, submitButton, 5, 5, 1, 1, true);  
 submitButton.setVisible(true);  
 layout(p, cancelButton, 5, 6, 1, 1, true);  
 cancelButton.setVisible(true);  
 });  
  
 submitButton.addActionListener(e -> {  
 if (nameTextField.getText().isEmpty() || addressTextArea.getText().isEmpty()) {  
 JOptionPane.showMessageDialog(AddressBook.this, "Please enter a name and address.",  
 "Error", JOptionPane.OK\_OPTION);  
 return;  
 }  
 if (mode.equals(0) || (!mode.equals(1) && userBook.get(currentIndex).getName() != nameTextField.getText()))  
 if (userBook.stream()  
 .filter(t -> t.getName().equals(nameTextField.getText()))  
 .collect(Collectors.toCollection(ArrayList::new)).size() != 0) {  
 JOptionPane.showMessageDialog(AddressBook.this, "Sorry, " + nameTextField.getText() + " is already in your address book",  
 "Error", JOptionPane.OK\_OPTION);  
 return;  
 }  
 if (mode == 0)  
 addNote(nameTextField.getText(), addressTextArea.getText());  
 else  
 editNote(currentIndex, nameTextField.getText(), addressTextArea.getText());  
 if (mode == 0)  
 JOptionPane.showMessageDialog(AddressBook.this, "Adding success");  
 else  
 JOptionPane.showMessageDialog(AddressBook.this, "Edit success");  
 activeButtons();  
 });  
  
 cancelButton.addActionListener(e -> {  
 if (userBook.size() > 0) {  
 nameTextField.setText(userBook.get(currentIndex).getName());  
 addressTextArea.setText(userBook.get(currentIndex).getAddress());  
 }  
 if (userBook.size() > 0) {  
 activeButtons();  
 } else {  
 addButton.setEnabled(true);  
 editButton.setEnabled(false);  
 removeButton.setEnabled(false);  
 findButton.setEnabled(false);  
 saveButton.setEnabled(false);  
 exportButton.setEnabled(false);  
 nextButton.setEnabled(false);  
 previousButton.setEnabled(false);  
  
 editButton.setVisible(true);  
 removeButton.setVisible(true);  
 findButton.setVisible(true);  
 loadButton.setVisible(true);  
 saveButton.setVisible(true);  
 exportButton.setVisible(true);  
 nextButton.setVisible(true);  
 previousButton.setVisible(true);  
 submitButton.setVisible(false);  
  
 cancelButton.setVisible(false);  
 nameTextField.setEnabled(false);  
 addressTextArea.setEnabled(false);  
 }  
 });  
  
 nextButton.addActionListener(e -> {  
 currentIndex = currentIndex + 1 != userBook.size() ? currentIndex + 1 : 0;  
 User current = userBook.get(currentIndex);  
 nameTextField.setText(current.getName());  
 addressTextArea.setText(current.getAddress());  
 });  
  
 previousButton.addActionListener(e -> {  
 currentIndex = (currentIndex - 1) >= 0 ? currentIndex - 1 : userBook.size() - 1;  
 User current = userBook.get(currentIndex);  
 nameTextField.setText(current.getName());  
 addressTextArea.setText(current.getAddress());  
 });  
  
 editButton.addActionListener(e -> {  
 nameTextField.setEnabled(true);  
 addressTextArea.setEnabled(true);  
 mode = 1;  
 addButton.setVisible(false);  
 editButton.setEnabled(false);  
 removeButton.setVisible(false);  
 findButton.setVisible(false);  
 loadButton.setVisible(false);  
 saveButton.setVisible(false);  
 exportButton.setVisible(false);  
 nextButton.setEnabled(false);  
 previousButton.setEnabled(false);  
 layout(p, submitButton, 5, 5, 1, 1, true);  
 submitButton.setVisible(true);  
 layout(p, cancelButton, 5, 6, 1, 1, true);  
 cancelButton.setVisible(true);  
 });  
  
 removeButton.addActionListener(e -> {  
 int res = JOptionPane.showConfirmDialog(AddressBook.this, "Are you sure?", "Warning", JOptionPane.YES\_NO\_OPTION, JOptionPane.WARNING\_MESSAGE);  
 if (res == JOptionPane.YES\_OPTION)  
 removeNote(currentIndex);  
 });  
  
 findButton.addActionListener((ActionEvent e) -> {  
 JFrame j = new JFrame();  
 j.setLocationRelativeTo(null);  
 JPanel pa = new JPanel();  
 pa.setLayout(new FlowLayout());  
 JLabel l = new JLabel("Enter the name of a contact:");  
 JTextField t = new JTextField(10);  
 JButton b = new JButton("Find");  
 b.addActionListener(e1 -> {  
 int indexFind = findByName(t.getText());  
 if (indexFind == -1)  
 JOptionPane.showMessageDialog(AddressBook.this, "Sorry, " + t.getText() + " is not in your address book.",  
 "", JOptionPane.OK\_OPTION);  
 else {  
 currentIndex = indexFind;  
 nameTextField.setText(userBook.get(indexFind).getName());  
 addressTextArea.setText(userBook.get(indexFind).getAddress());  
 }  
 AddressBook.this.setEnabled(true);  
 j.setVisible(false);  
 j.dispose();  
 });  
  
 pa.add(l);  
 pa.add(t);  
 pa.add(b);  
 j.add(pa);  
 j.setSize(500, 500);  
 j.addWindowListener(new WindowAdapter() {  
 public void windowClosing(WindowEvent e) {  
 AddressBook.this.setEnabled(true);  
 j.dispose();  
 }  
 });  
 j.setVisible(true);  
 j.setResizable(false);  
 j.pack();  
 setEnabled(false);  
 });  
  
 exportButton.addActionListener((e -> {  
 JFileChooser c = new JFileChooser();  
 c.addChoosableFileFilter(new FileNameExtensionFilter("vCard Files", "vcf"));  
  
 int rVal = c.showSaveDialog(AddressBook.this);  
 if (rVal == JFileChooser.APPROVE\_OPTION) {  
 String temp = "";  
 if (c.getFileFilter().getDescription().equals("vCard Files") && !c.getSelectedFile().toString().contains(".vcf"))  
 temp += ".vcf";  
 exportAddress(c.getSelectedFile().toString() + temp);  
 JOptionPane.showMessageDialog(AddressBook.this, "Export success");  
 }  
 loadButton.addActionListener(e -> {  
 JFileChooser c = new JFileChooser();  
 int rVal = c.showOpenDialog(AddressBook.this);  
 if (rVal == JFileChooser.APPROVE\_OPTION) {  
 JOptionPane.showMessageDialog(AddressBook.this, "Load success");  
 loadAddressBook(c.getSelectedFile().toString());  
 }  
 });  
  
 saveButton.addActionListener(e -> {  
 JFileChooser c = new JFileChooser();  
 c.addChoosableFileFilter(new FileNameExtensionFilter("Text document", "txt"));  
  
 int rVal = c.showSaveDialog(AddressBook.this);  
 if (rVal == JFileChooser.APPROVE\_OPTION) {  
 String temp = "";  
 if (c.getFileFilter().getDescription().equals("Text document") && !c.getSelectedFile().toString().contains(".txt"))  
 temp += ".txt";  
 saveAddressBook(c.getSelectedFile().toString() + temp);  
 JOptionPane.showMessageDialog(AddressBook.this, "Save success");  
 }  
 });  
  
 gbc.insets = new Insets(5, 5, 5, 5);  
 gbc.anchor = GridBagConstraints.NORTH;  
 layout(p, nameLabel, 0, 0, 1, 1, true);  
 layout(p, nameTextField, 1, 0, 4, 1, true);  
 layout(p, addressLabel, 0, 1, 1, 1, true);  
 layout(p, addressTextArea, 1, 1, 4, 9, false);  
 gbc.anchor = GridBagConstraints.CENTER;  
 layout(p, addButton, 5, 1, 1, 1, true);  
 layout(p, editButton, 5, 2, 1, 1, true);  
 layout(p, removeButton, 5, 3, 1, 1, true);  
 layout(p, findButton, 5, 4, 1, 1, true);  
 layout(p, loadButton, 5, 5, 1, 1, true);  
 layout(p, saveButton, 5, 6, 1, 1, true);  
 layout(p, exportButton, 5, 7, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 2, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 3, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 4, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 5, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 6, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 7, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 8, 1, 1, true);  
 layout(p, new JLabel(" "), 0, 9, 1, 1, true);  
 layout(p, previousButton, 1, 10, 2, 1, true);  
 layout(p, nextButton, 3, 10, 2, 1, true);  
  
 editButton.setEnabled(false);  
 removeButton.setEnabled(false);  
 findButton.setEnabled(false);  
 saveButton.setEnabled(false);  
 exportButton.setEnabled(false);  
 nextButton.setEnabled(false);  
 previousButton.setEnabled(false);  
 add(p);  
 }  
  
 public void activeButtons() {  
 if (mode == 0) {  
 addButton.setEnabled(true);  
 editButton.setEnabled(true);  
 editButton.setVisible(true);  
 } else {  
 editButton.setEnabled(true);  
 addButton.setEnabled(true);  
 addButton.setVisible(true);  
 }  
 removeButton.setVisible(true);  
 removeButton.setEnabled(true);  
 findButton.setEnabled(true);  
 loadButton.setEnabled(true);  
 saveButton.setEnabled(true);  
 exportButton.setEnabled(true);  
 findButton.setVisible(true);  
 loadButton.setVisible(true);  
 saveButton.setVisible(true);  
 exportButton.setVisible(true);  
 nextButton.setEnabled(true);  
 previousButton.setEnabled(true);  
  
 submitButton.setVisible(false);  
 cancelButton.setVisible(false);  
  
 nameTextField.setEnabled(false);  
 addressTextArea.setEnabled(false);  
 }  
  
 public void layout(JPanel p, JComponent c, Integer x, Integer y, Integer w, Integer h, boolean b) {  
 gbc.gridx = x;  
 gbc.gridy = y;  
 gbc.gridwidth = w;  
 gbc.gridheight = h;  
 gbc.fill = b ? GridBagConstraints.HORIZONTAL : GridBagConstraints.VERTICAL;  
 p.add(c, gbc);  
 }  
  
 public void addNote(String name, String address) {  
 userBook.add(new User(name, address));  
 currentIndex = userBook.size() - 1;  
 }  
  
 public void editNote(Integer index, String name, String address) {  
 userBook.get(index).edit(name, address);  
 }  
  
 public void removeNote(int index) {  
 userBook.remove(index);  
 if (index != userBook.size()) {  
 nameTextField.setText(userBook.get(index).getName());  
 addressTextArea.setText(userBook.get(index).getName());  
 } else if (index - 1 < 0) {  
 removeButton.setEnabled(false);  
 editButton.setEnabled(false);  
 findButton.setEnabled(false);  
 saveButton.setEnabled(false);  
 exportButton.setEnabled(false);  
 nextButton.setEnabled(false);  
 previousButton.setEnabled(false);  
 nameTextField.setText("");  
 addressTextArea.setText("");  
 } else {  
 currentIndex -= 1;  
 nameTextField.setText(userBook.get(index - 1).getName());  
 addressTextArea.setText(userBook.get(index - 1).getName());  
 }  
  
 }  
  
 public int findByName(String name) {  
 for (int i = 0; i < userBook.size(); i++)  
 if (userBook.get(i).getName().equals(name))  
 return i;  
 return -1;  
 }  
  
 public void loadAddressBook(String fileAddress) {  
 userBook = new ArrayList<>();  
 try {  
 Files.lines(Paths.get(fileAddress), StandardCharsets.UTF\_8).forEach(e -> {  
 ArrayList<String> list = Arrays.stream(e.split(" && ")).collect(Collectors.toCollection(ArrayList::new));  
 userBook.add(new User(list.get(0), list.get(1)));  
 });  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 if (userBook.size() > 0) {  
 editButton.setEnabled(true);  
 removeButton.setEnabled(true);  
 findButton.setEnabled(true);  
 saveButton.setEnabled(true);  
 exportButton.setEnabled(true);  
 nextButton.setEnabled(true);  
 previousButton.setEnabled(true);  
 currentIndex = userBook.size() - 1;  
 nameTextField.setText(userBook.get(currentIndex).getName());  
 addressTextArea.setText(userBook.get(currentIndex).getAddress());  
 }  
 }  
  
 public void saveAddressBook(String fileDestination) {  
 try {  
 File file = new File(fileDestination);  
 file.createNewFile();  
 try (FileWriter writer = new FileWriter(file)) {  
 for (User user : userBook)  
 writer.write(user.getName() + " && " + user.getAddress() + "\n");  
 writer.flush();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void exportAddress(String fileDestination) {  
 try {  
 File file = new File(fileDestination);  
 file.createNewFile();  
 try (FileWriter writer = new FileWriter(file)) {  
 writer.write("BEGIN:VCARD\n");  
 writer.write("VERSION:2.1\n");  
 String name = nameTextField.getText();  
 String address = addressTextArea.getText();  
 String fname;  
 String sname;  
 ArrayList<String> listName;  
 int i = name.indexOf(" ");  
 if (i != -1) {  
 listName = Stream.of(name.split("\\s+")).collect(Collectors.toCollection(ArrayList::new));  
 fname = listName.get(0);  
 sname = listName.get(listName.size() - 1);  
 } else {  
 listName = new ArrayList<>();  
 fname = name;  
 sname = "";  
 }  
 writer.write("N:" + sname + ";" + fname + "\n");  
 String temp = "";  
 for (String s : listName)  
 temp += s + " ";  
 if (!listName.isEmpty())  
 writer.write("FN:" + temp.trim() + "\n");  
 else  
 writer.write("FN:" + fname + "\n");  
 writer.write("ADR;HOME:;" + address + "\n");  
 writer.write("END:VCARD\n");  
 writer.flush();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static void main(String[] args) {  
 EventQueue.invokeLater(() -> {  
 AddressBook frame = new AddressBook();  
 frame.setSize(500, 500);  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setLocationRelativeTo(null);  
 frame.setVisible(true);  
 frame.setTitle("Address book");  
 frame.setResizable(false);  
 frame.pack();  
 });  
 }  
}

**Результат работы:**





**Вывод:** изучили базовые компоненты средства разработки графического интерфейса.