package messenger\_client**;**import java.awt.EventQueue**;**import javax.swing.JFrame**;**import javax.swing.JPanel**;**import javax.swing.border.EmptyBorder**;**import java.awt.GridBagLayout**;**import javax.swing.JLabel**;**import java.awt.GridBagConstraints**;**import javax.swing.JTextField**;**import java.awt.Insets**;**import java.awt.event.ActionEvent**;**import java.awt.event.ActionListener**;**import javax.swing.JButton**;**public class loginPage extends JFrame { //Наследуя от JFrame мы получаем всю функциональность окна for login  
  
 private JPanel contentPane**;** //создаем панель для авторизации пользователей  
 private JTextField textField**;** //создаем 2 текстовых поля для IP - адреса  
 private JTextField textField1**;** // и для имени пользователя  
  
 public static void main(String[] args) { //основная функция для реализации программы  
 EventQueue.*invokeLater*(new Runnable(){ //отправляет событие  
 // в конце списка событий Swings и обрабатывается после обработки всех предыдущих графических событий  
  
 public void run() {  
 try {  
 loginPage frame = new loginPage()**;** //создаем новый объект для окошка страницы логина  
 frame.setVisible(true)**;** // и задаем его видимым  
  
 //отлавливаем в случая наличия ошибки исключения  
 // и выводим сообщение в случаи переполнения стека памяти "Вылет по памяти"  
 } catch (Exception e) {  
 e.printStackTrace()**;** }  
 }  
 })**;** }  
  
//класс для создания страницы логина для авторизации  
 public loginPage() {  
  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*)**;** //№4  
 setBounds(**100, 100, 300, 120**)**;** //устанавливаем коорднаты в пространстве для расположения блока  
 // длинну и шириину блока логина  
  
 contentPane = new JPanel()**;** // создаем новый объект для контерйнера, который содержится внутри самой страницы для логина, и  
 // содержащий основные компоненты пользовательского интерфейса (лейблы, поля для ввода, кнопки)  
  
 contentPane.setBorder(new EmptyBorder(**5, 5, 5, 5**))**;** //устанавливаем значения координат расположения  
 // для объекта типа contentPane внутри JFrame  
 setContentPane(contentPane)**;** //Метод setContentPane(JPanel panel) позволяет заменить панель содержимого окна.  
 GridBagLayout gbl\_contentPane = new GridBagLayout()**;** // creating the new object for диалоговая панель, содержащая текстовые поля редактирования, переключатели, кнопки и т.д.  
 //№5 разбиваем панель на ячейки, у каждой свои координаты ширины и высоты, 0 - чтобы все они были одинаковые по размеру  
 gbl\_contentPane.columnWidths = new int[]{**0, 0, 0, 0**}**;** // ширина колонок диалоговой панели  
 gbl\_contentPane.rowHeights = new int[]{**0, 0, 0, 0, 0, 0, 0, 0, 0**}**;** gbl\_contentPane.columnWeights = new double[]{**0.0, 0.0, 1.0,** Double.*MIN\_VALUE*}**;** gbl\_contentPane.rowWeights = new double[]{**0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,** Double.*MIN\_VALUE*}**;** contentPane.setLayout(gbl\_contentPane)**;**//устанавливаем значения для слоев  
  
  
 // простейшее использование режима размещения GridBagLayout системы Layout Manager для орагнизации формы ввода.  
  
 /\*!!!! начало блока для лэйбла имени пользователя !!!!\*/  
 JLabel lblNickname = new JLabel("name of user")**;** //создаем конструктор объекта лейбл с надписью name of user  
 //соответствующие поля класса GridBagConstraints создаем  
 //панели, напоминающие по своему внешнему виду и поведению стандрантые диалоговые панели Windows  
 GridBagConstraints gbc\_lblNickname = new GridBagConstraints()**;** //задать значения полей объекта класса GridBagConstraints #6  
 gbc\_lblNickname.anchor = GridBagConstraints.*WEST***;** gbc\_lblNickname.insets = new Insets(**0, 0, 5, 0**)**;** gbc\_lblNickname.gridx = **0;** gbc\_lblNickname.gridy = **4;** contentPane.add(lblNickname**,** gbc\_lblNickname)**;** //добавляем на страницу контента лейбл с надписью и поля объектов выше  
 /\*!!!! конец блока для лэйбла имени пользователя !!!!\*/  
  
 /\*!!!! начало блока для поля ввода имени пользователя !!!!\*/  
 // The constructor of the class are : JTextField() : constructor that creates a new TextField.  
 textField = new JTextField()**;** //создаем новый объект для текстового поля - для ввода имя пользователя  
 GridBagConstraints gbc\_textField = new GridBagConstraints()**;** //все тоже самое, что и в пред случаи  
 gbc\_textField.insets = new Insets(**0, 0, 5, 0**)**;** gbc\_textField.fill = GridBagConstraints.*HORIZONTAL***;** gbc\_textField.gridx = **2;** gbc\_textField.gridy = **4;** contentPane.add(textField**,** gbc\_textField)**;** textField.setColumns(**10**)**;** //устанавливаем значение для этого поля в виде 10 колонки (колонок)  
 /\*!!!! конец блока для поля ввода имени пользователя !!!!\*/  
  
 /\*!!!! начало блока для лэйбла ip - адрес сервера !!!!\*/  
 //те же процедуры для второго лейбла с текстовой надписью  
 JLabel lblNickname1 = new JLabel("ip - адрес сервера")**;** GridBagConstraints gbc\_lblip = new GridBagConstraints()**;** gbc\_lblip.anchor = GridBagConstraints.*WEST***;** gbc\_lblip.insets = new Insets(**0, 0, 5, 0**)**;** gbc\_lblip.gridx = **0;** gbc\_lblip.gridy = **1;** contentPane.add(lblNickname1**,** gbc\_lblip)**;** /\*!!!! конец блока для лэйбла ip - адрес сервера !!!!\*/  
  
 /\*!!!! начало блока для поля ввода ip - адрес сервера !!!!\*/  
 //те же процедуры для текстового поля для ввода ip - адреса локалхоста  
 textField1 = new JTextField()**;** GridBagConstraints gbc\_textField1 = new GridBagConstraints()**;** gbc\_textField1.insets = new Insets(**0, 0, 5, 0**)**;** gbc\_textField1.fill = GridBagConstraints.*HORIZONTAL***;** gbc\_textField1.gridx = **2;** gbc\_textField1.gridy = **1;** contentPane.add(textField1**,** gbc\_textField1)**;** textField1.setColumns(**10**)**;** /\*!!!! конец блока для поля ввода ip - адрес сервера !!!!\*/  
  
 /\* создаем кнопку залогинивания пользователя\*/  
 JButton btnNewButton = new JButton("Login")**;** //новый объект JButton для кнопки с названиием "Login"  
 GridBagConstraints gbc\_btnNewButton = new GridBagConstraints()**;** //задать значения полей объекта класса GridBagConstraints #6  
 gbc\_btnNewButton.gridx = **2;** gbc\_btnNewButton.gridy = **7;** contentPane.add(btnNewButton**,** gbc\_btnNewButton)**;** //добавить на панель кнопку  
  
 //добавляем для кнопки функцию действия при ее нажатии  
 btnNewButton.addActionListener(new ActionListener() { //#7  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 String ipText = textField1.getText()**;** //передаем значение из поля ввода для ip - address  
 String nameText = textField.getText()**;** //передаем значение из поля ввода для имени пользователя  
 // ClientWindow clientWindow = new ClientWindow(ipText,nameText);  
 new ClientWindow(ipText**,**nameText)**;** //создаем новый объект для клиентского окна с переданными выше параметрами  
 }  
 })**;** }  
}