Министерство науки и образования Украины

Одесский национальный политехнический университет

Кафедра системного программного обеспечения

Объяснительная записка к курсовой работе

с дисциплины «Объектно-ориентированное программирование»

Группа: АС-071

Выполнил: Розгон А. С.

Приняла: Кунгурцев А. Б.

Одесса 2009 г.

**Аннотация**

Данная курсовая работа заключалась в написании учебной системы на языке программирования Java, что и было сделано. Учебная система представляет собой совокупность пяти основных классов и около десятка вспомогательных. Она предполагает регистрацию, выдачу учебного материала, проверку знаний, способы выдачи и принятия информации, детально описанные в задании для каждого варианта отдельно. Также имеются общие характеристики системы для всех вариантов, и они являются базовыми, индивидуальные только расширяют их. Именно это повлияло на распределение функций по классам. На основе задания и программы была составлена диаграмма классов, иллюстрирующая все аспекты вышеперечисленного, и поэтому была распечатана отдельном листе формата А1. В данной пояснительной записке указана инструкция использования, некоторые особенности поведения программы, приведены коды всех классов и файлов, а также имеется объяснение к диаграмме классов на А1.

**Содержание**

1. Задание
2. Анализ требований
3. Структуры классов
4. Коды классов
5. Структура файлов
6. Инструкция пользователя и контрольный пример

Список использованной литературы

**1. Задание**

Завдання №8.

Учбовий матеріал – функції з параметрами-масивами мови С++.

Треба реалізувати наступні вимоги до програмних класів:1.4, 2.1, 3.6, 4.1, 5.4

Навчальна програма повинна складатися з наступних класів.

**1.Клас представлення навчального матеріалу.**

Клас обов’язково реалізує наступні функції.

Читання навчального матеріалу з файла.

Надання студенту порції навчального матеріалу.

Перехід до наступної порції матеріалу.

Клас додатково може реалізувати наступні функції відповідно до конкретного завдання до курсової роботи.

Матеріал подається за схемою, що задається викладачем для кожного конкретного випадку.

**2.Клас контролю знань студента.**

Клас обов’язково реалізує наступні функції.

Читання завдання до контролю з файлу.

Читання еталонних відповідей з файлу.

Надання студенту завдань до контролю знань.

Оцінювання кожної відповіді.

Клас додатково може реалізувати наступні функції відповідно до конкретного завдання до курсової роботи.

Завдання типу «вибір альтернативної відповіді».

**3.Клас реєстрації студента та видачі підсумкового документу.**

Клас обов’язково реалізує наступні функції.

Приймає дані щодо студента.

Видає підсумковий документ.

Клас додатково може реалізувати наступні функції відповідно до конкретного завдання до курсової роботи.

Підсумковий документ містить загальний час навчання.

**4.Клас введення учбового матеріалу, контрольних завдань, еталонів відповідей та іншої інформації.**

Клас обов’язково реалізує наступні функції.

Введення учбового матеріалу у форматі, що задано іншими пунктами завдання. Введення контрольних завдань у форматі, що задано іншими пунктами завдання.Введення еталонів відповідей у форматі, що задано іншими пунктами завдання.

Клас додатково може реалізувати наступні функції відповідно до конкретного завдання до курсової роботи.

Клас приймає дані тільки у діалоговому режимі.

**5.Клас контролю за ходом навчального процесу викладачем.**

Клас обов’язково реалізує наступні функції.

Перегляд результатів навчання для кожного студента.

Клас додатково може реалізувати наступні функції відповідно до конкретного завдання до курсової роботи.

Отримання даних щодо тем, які успішно пройшов конкретний студент.

Завдання на курсову роботу видається на початку 3 семестру.

Докладний зміст курсової роботи характеризується типовим завданням.

**2. Анализ тренований:**

- Очевидно, что нужно ввести два типа пользователя: студент и

преподаватель, так как их функции и возможности граничат и

конфликтуют.

- Система должна содержать пять главных классов и несколько

второстепенных;

- Система должна выполнять все пункты задания;

**3. Структуры классов**

На диаграмме классов, приведенной отдельно от объяснительной записке на А1, мы можем видеть поочередность вызова классов, их взаимосвязи, функции и переменные, поэтому дальше описаны детали того, что мы увидели:

1. Class Main – чтобы работать с системой, программа должна считывать с определенных файлов определенные данные(а именно файлы Control.txt Answers.txt, указанные в задании, но сначала они должны быть записаны, что и реализует этот класс. Помимо этого этот класс вызывает класс, отвечающий за регистрацию – ButtonFrame11. Все переменные этого класса – строки, которые записываются в выше перечисленные классы, а функция writeData () – это реализует.
2. Class ButtonFrame11 – класс, отвечающий за правильное расположения окна регистрации и вызов самого класса регистрации – ButtonPanel11.
3. Class ButtonPunel11 – класс, отвечающий за регистрацию, вход в систему и авторизацию. Содержит в себе элементы окна регистрации и внутренний класс ColorAction11, который отвечает за получение и обработку данных, полученных от пользователя. В зависимости от этого он может вызвать, а может и не вызвать класс ButtonFrame. Единственный метод этого класса – ActionPerformed () – отвечает за считывание данных, сравнения их с существующими, создания или не создания различных файлов в различных ситуациях.
4. Class ButtonFrame – класс, содержащий в себе элементы главного окна, отвечающий за распределения возможностей между студентом и преподавателем, отвечающий за правильное расположение главного окна. В нем имеется встроенный класс ColorAction – отвечающий за идентификацию действия, который вызывает пользователь. И в зависимости от этого может вызвать следующие классы: JFrame1, JFrame2, CheckBoxFrame, MyTimer, MyTimer1.
5. Class JFrame1 – это класс, который отвечает за расположения окна частичного вывода материала. Его главная задача – вызов класса ButtonPanel1.
6. Class ButtonPanel1 – класс, содержащий все элементы окна, и отвечающий за чтение и вывод информации на экран. Для этого он имеет функцию readData () – отвечающую за чтение, и встроенный класс – Color1Action – отвечающий за преобразования на экране, что, в свою очередь, осуществляет функция actionPefrormed ().
7. Class JFrame2 – это класс, который отвечает за расположения окна изменение порядка вывода материала. Его главная задача – вызов класса ButtonPanel2.
8. Class ButtonPanel2 – – класс, содержащий все элементы окна, и отвечающий за принятие и обработку информации, введенной пользователем. Для этого он имеет функцию getPrevios() – отвечающую за преобразование информации, и функцию actionPefrormed (), которая отвечает за считывание.
9. Class CheckBoxFrame – это класс, содержащий все элементы и свойства окна тестирование знаний, который отвечает за вывод информации из файла на экран, принятия информации пользователя, формулирования и вывод результатов на экран, а также их последующая запись в файл. Функция addRadioButton () – отвечает за добавления кнопки и элементов идентификации. Функция actionPerformed () – отвечает за чтение и сравнение информаций пользователя и конкретного задания.
10. Class MyTimer – это класс, отвечающий за выход из системы и вывод на экран результатов работы системы, для чего в нем предусмотрена функция readData () – для чтения информации из файла, а также функция actionPerformed () – для выхода с системы.
11. Class MyTimer1 – класс, отвечающий за те же функции, что и класс MyTimer, содержащий те же элементы, но имеется существенное исключение : MyTimer – выдавал информацию о текущем клиенте, а класс MyTimer1 – выдает информацию о любом студенте.
12. Class Device – отвечает за первоначальную запись и чтение элементов в файле, так как они являются объектами класса, это реализует функции readData () и writeData (), а доступ к элементам класса функция getName ().

**4. Коды классов**

package javaapplication20;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.net.\*;

import java.lang.reflect.\*;

public class Main

{

public static void main(String[] args)throws IOException

{

DataInputStream inData = new DataInputStream(System.in);

String T0="Оцените утверждение : массив в С++ никогда не передается по значению";

String T1="Верно ли, что размер массива важен при объявлении параматра?";

String T2="Действительно ли, что указатель указывает на нулевой элемент массива, при передачи массива";

String T3="Оцените утверждение: изменение значения аргумента внутри функции затрагивает сам переданный объект и его локальную копию ";

String T4="Верно ли, размер массива не является частью типа параметра ";

String T5="Действительно ли, функция знает реальный размер передаваемого массива?";

String T6="Правда ли, что в конце каждого массива имеется , который указывает на конец массива";

String T7="Оцените утверждение: другой способ сообщить функции размер массива-параметра – объявить параметр как ссылку ";

String T8="Верно ли, что использование абстрактный контейнерный тип - это один из способов получения размера передаваемого массива";

String T9="Правда ли, что параметры шаблона заключаются в круглые скобки";

String T10="3";

String T11="2";

String T12="3";

String T13="1";

String T14="3";

String T15="2";

String T16="2";

String T17="3";

String T18="3";

String T19="2";

try{

FileOutputStream OS=new FileOutputStream (new File("C:\\USER\\Control.txt"));

DataOutputStream ods = new DataOutputStream (OS);

ods.writeUTF(T0);

ods.writeUTF(T1);

ods.writeUTF(T2);

ods.writeUTF(T3);

ods.writeUTF(T4);

ods.writeUTF(T5);

ods.writeUTF(T6);

ods.writeUTF(T7);

ods.writeUTF(T8);

ods.writeUTF(T9);

ods.close();

FileOutputStream OS1=new FileOutputStream (new File("C:\\USER\\Answers.txt"));

DataOutputStream ods1 = new DataOutputStream (OS1);

ods1.writeUTF(T10);

ods1.writeUTF(T11);

ods1.writeUTF(T12);

ods1.writeUTF(T13);

ods1.writeUTF(T14);

ods1.writeUTF(T15);

ods1.writeUTF(T16);

ods1.writeUTF(T17);

ods1.writeUTF(T18);

ods1.writeUTF(T19);

ods1.close();

}

catch (IOException e)

{

System.out.print("Can not read file55555"+e);

}

ButtonFrame11 frame1 = new ButtonFrame11 ();

frame1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame1.setVisible(true);

Device[] staff = new Device[10];

staff[0] = new Device("Массив в С++ никогда не передается по значению, а только как указатель на его первый, точнее нулевой, элемент");

staff[1] = new Device("Размер массива неважен при объявлении параметра");

staff[2] = new Device("Передача массивов как указателей имеет следующие особенности:");

staff[3] = new Device("изменение значения аргумента внутри функции затрагивает сам переданный объект, а не его локальную копию. Если такое поведение нежелательно, программист должен позаботиться о сохранении исходного значения. Можно также при объявлении функции указать, что она не должна изменять значение параметра, объявив этот параметр константой: ");

staff[4] = new Device("размер массива не является частью типа параметра. Поэтому функция не знает реального размера передаваемого массива. Компилятор тоже не может это проверить");

staff[5] = new Device("При проверке типов параметров компилятор способен распознать, что в обоих случаях тип аргумента int\* соответствует объявлению функции. Однако контроль за тем, не является ли аргумент массивом, не производится.");

staff[6] = new Device("По принятому соглашению C-строка является массивом символов, последний элемент которого равен нулю. Во всех остальных случаях при передаче массива в качестве параметра необходимо указывать его размер. Это относится и к массивам символов, внутри которых встречается 0. Обычно для такого указания используют дополнительный параметр функции.");

staff[7] = new Device("Другой способ сообщить функции размер массива-параметра – объявить параметр как ссылку. В этом случае размер становится частью типа, и компилятор может проверить аргумент в полной мере.");

staff[8] = new Device("Еще один способ получить размер переданного массива в функции – использовать абстрактный контейнерный тип. (Такие типы были представлены в главе 6. В следующем подразделе мы поговорим об этом подробнее");

staff[9] = new Device("Параметры шаблона заключаются в угловые скобки. Ключевое слово class означает, что идентификатор Type служит именем параметра, при конкретизации шаблона функции putValues() он заменяется на реальный тип – int, double, string и т.д. (В главе 10 мы продолжим разговор о шаблонах функций");

Device[] staff3 = new Device[10];

staff3[0] = new Device("Передача массивов в С++");

staff3[1] = new Device("Размер массива ");

staff3[2] = new Device("Передача массивов как указателей имеет следующие особенности...");

staff3[3] = new Device("Свойство 1");

staff3[4] = new Device("Свойство 2");

staff3[5] = new Device("Расмотрение примера");

staff3[6] = new Device("Дополнительный параметер функции");

staff3[7] = new Device("Объявление параметра как ссылку");

staff3[8] = new Device("Абстрактный контейнерный тип");

staff3[9] = new Device("Шаблоны");

try

{

PrintWriter out = new PrintWriter( new FileWriter("device.dat"));

writeData(staff,out);

out.close();

PrintWriter out1 = new PrintWriter( new FileWriter("device1.dat"));

writeData(staff3,out1);

out1.close();

}

catch(IOException exception){

exception.printStackTrace();

}

}

static void writeData(Device[] e, PrintWriter out)

throws IOException

{

out.println(e.length);

for(int i=0;i<e.length;i++)

e[i].writeData(out);

}

}

class ButtonFrame extends JFrame

{

public ButtonFrame()

{

Toolkit kit = Toolkit.getDefaultToolkit();

Dimension screenSize = kit.getScreenSize();

int screenHeight = screenSize.height;

int screenWidth = screenSize.width;

setSize(screenWidth/2,screenHeight/2);

setLocation(screenWidth/4,screenHeight/4);

Image img = kit.getImage("icon.gif");

setIconImage(img);

setTitle("CenteredFrame");

Container contantPane = getContentPane();

JMenuBar menuBar = new JMenuBar();

JMenu stMenu = new JMenu("Студент");

JMenuItem WorkSpace = new JMenuItem("Просмотреть материал",new ImageIcon("page\_bookmark.gif"));

stMenu.add(WorkSpace);

menuBar.add(stMenu);

JButton yellowButton = new JButton("WorkSpace");

JButton greenButton = new JButton("I\_want\_to\_see");

JButton blueButton = new JButton("I am a teacher");

JButton redButton = new JButton("Check\_Knowlarge");

JButton ExitButton = new JButton("Exit");

JPanel wer= new JPanel();

wer.add(yellowButton);

wer.add (blueButton);

wer.add(redButton);

wer.add(ExitButton);

wer.add(greenButton);

//contantPane.add(wer,BorderLayout.SOUTH);

contantPane.add(menuBar,BorderLayout.NORTH);

ColorAction yellowAction = new ColorAction(2);

ColorAction blueAction = new ColorAction(3);

ColorAction redAction = new ColorAction(4);

ColorAction ExitAction = new ColorAction(5);

ColorAction greenAction = new ColorAction(6);

yellowButton.addActionListener(yellowAction);

blueButton.addActionListener(blueAction);

redButton.addActionListener(redAction);

ExitButton.addActionListener(ExitAction);

greenButton.addActionListener(greenAction);

WorkSpace.addActionListener(yellowAction);

JMenuItem Check = new JMenuItem("Проверить знания",new ImageIcon("icon\_security.gif"));

stMenu.add(Check);

Check.addActionListener(redAction);

JMenuItem Exit = new JMenuItem("Выход",new ImageIcon("action\_stop.gif"));

stMenu.add(Exit);

Exit.addActionListener(ExitAction);

JMenu prMenu = new JMenu("Преподаватель");

JMenuItem teacher = new JMenuItem("Изменить",new ImageIcon("page\_down.gif"));

prMenu.add(teacher);

teacher.addActionListener(blueAction);

JMenuItem teacher1 = new JMenuItem("Посмотреть",new ImageIcon("icon\_world\_dynamic.gif"));

prMenu.add(teacher1);

teacher1.addActionListener(greenAction);

menuBar.add(prMenu);

try

{

FileInputStream is = new

FileInputStream(new

File("C:\\USER\\teacher.txt"));

DataInputStream ids = new DataInputStream(is);

String S2=ids.readUTF();

n=Integer.parseInt(S2);

}

catch(IOException exception){

System.out.println("bgbvhgfj");

}

if(n==0)

{

teacher.setEnabled(false);

teacher1.setEnabled(false);

System.out.println(n);

}

else

{

Check.setEnabled(false);

Exit.setEnabled(false);

System.out.println(n);

}

}

private class ColorAction implements ActionListener

{

public ColorAction(int c)

{

ccc=c;

}

public void actionPerformed(ActionEvent event)

{

if(ccc==2)

{

JFrame1 frame2 = new JFrame1 ();

frame2.setVisible(true);

}

if(ccc==3)

{

JFrame2 frame3 = new JFrame2 ();

frame3.setVisible(true);

}

if(ccc==4)

{

CheckBoxFrame frame3 = new CheckBoxFrame();

frame3.setVisible(true);

}

if(ccc==5)

{

MyTimer frame3 = new MyTimer();

frame3.setVisible(true);

}

if(ccc==6)

{

MyTimer1 frame3 = new MyTimer1();

frame3.setVisible(true);

}

}

private int ccc;

}

private int n=0;

}

class MyTimer extends JFrame

{

public MyTimer()

{

setSize(650,350);

setTitle("Your answer");

try

{

FileInputStream is = new

FileInputStream(new

File("C:\\USER\\NOW.txt"));

DataInputStream ids = new DataInputStream(is);

S2=ids.readUTF();

GregorianCalendar d = new GregorianCalendar();

int year=d.get(Calendar.YEAR);

int mouth=d.get(Calendar.MONTH);

int day=d.get(Calendar.DAY\_OF\_MONTH);

int hour=d.get(Calendar.HOUR);

int minute=d.get(Calendar.MINUTE);

int seconds=d.get(Calendar.SECOND);

FileInputStream is1 = new

FileInputStream(new

File("C:\\USER\\Time.txt"));

DataInputStream ids1 = new DataInputStream(is1);

String S22=ids1.readUTF();

int year\_0=Integer.parseInt(S22);

String S33=ids1.readUTF();

int mouth\_0=Integer.parseInt(S33);

String S44=ids1.readUTF();

int day\_0=Integer.parseInt(S44);

String S55=ids1.readUTF();

int hour\_0=Integer.parseInt(S55);

String S66=ids1.readUTF();

int minute\_0=Integer.parseInt(S66);

String S77=ids1.readUTF();

int seconds\_0=Integer.parseInt(S77);

if(seconds<seconds\_0)

{

seconds\_1=seconds-seconds\_0+60;

minute=minute-1;

}

else

{

seconds\_1=seconds-seconds\_0;

}

if(minute<minute\_0)

{

minute\_1=minute-minute\_0+60;

hour=hour-1;

}

else

{

minute\_1=minute-minute\_0;

}

if(hour<hour\_0)

{

hour\_1=hour-hour\_0+24;

day=day-1;

}

else

{

hour\_1=hour-hour\_0;

}

if(day<day\_0)

{

if(mouth==0||mouth==2||mouth==4||mouth==6||mouth==8||mouth==10)

g=31;

if(mouth==3||mouth==5||mouth==7||mouth==9||mouth==11)

g=30;

if(mouth==1)

g=28;

if(mouth==1 && year==2012)

g=29;

day\_1=day-day\_0+g;

mouth=mouth-1;

}

else

{

day\_1=day-day\_0;

}

if(mouth<mouth\_0)

{

mouth\_1=mouth - mouth\_0+12;

year=year-1;

}

else

{

mouth\_1=mouth - mouth\_0;

}

year\_1=year-year\_0;

FileInputStream is12 = new

FileInputStream(new

File("C:\\Student\\"+S2+".txt"));

ids13 = new DataInputStream(is12);

String E=ids13.readUTF();

int f=Integer.parseInt(E);

df=new Device[f];

m1=new int [f];

FileInputStream is121 = new

FileInputStream(new

File("C:\\Student1\\"+S2+".txt"));

ids131 = new DataInputStream(is121);

for(int r123=0;r123<f;r123++)

{

String E1=ids131.readUTF();

int f1=Integer.parseInt(E1);

m1[r123]=f1;

}

BufferedReader in = new BufferedReader(new FileReader("device1.dat"));

newStaff1=readData(in);

in.close();

for(int g1=0;g1<f;g1++)

{

r3=m1[g1];

df[r34]=newStaff1[r3];

r34++;

}

label9= new JLabel("FFF");

for(int g1=0;g1<f;g1++)

{

String rrr=df[g1].getName();

hhh=hhh+"<br>"+rrr;

}

label9.setText("<html><br> Вы правильно ответили на следующие темы(это - "+(f\*10)+"%)<br>"+hhh+"</html>");

Container contentPane = getContentPane ();

JPanel panel = new JPanel();

JPanel panel1 = new JPanel();

label = new JLabel ("Вниманиe "+S2+" , вы прошли тестирование");

label1 = new JLabel ("Ваше время пребывания:");

label2 = new JLabel (year\_1+" лет ");

label3 = new JLabel (mouth\_1+" месяцев ");

label4 = new JLabel (day\_1+" дней ");

label5 = new JLabel (hour\_1+" часов ");

label6 = new JLabel (minute\_1+" минут ");

label7 = new JLabel (seconds\_1+" секунд ");

contentPane.add(label,BorderLayout.NORTH);

panel1.add(label1);

panel1.add(label2);

panel1.add(label3);

panel1.add(label4);

panel1.add(label5);

panel1.add(label6);

panel1.add(label7);

panel1.add(label9);

replace2 = new JButton("OK");

panel.add(replace2);

contentPane.add(panel,BorderLayout.SOUTH);

contentPane.add(panel1,BorderLayout.CENTER);

}

catch (IOException e)

{

System.out.print("Can not read file66666"+e);

}

replace2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event2)

{

System.exit(0);

}

});

}

private JLabel label;

private JLabel label1;

private JLabel label2;

private JLabel label3;

private JLabel label4;

private JLabel label5;

private JLabel label6;

private JLabel label7;

private JLabel label9;

private JButton replace2;

private String S2;

private String hhh=new String("");

private int seconds\_1;

private int minute\_1;

private int hour\_1;

private int day\_1;

private int mouth\_1;

private int year\_1;

private int g;

private int r34=0;

private int r3=0;

private DataInputStream ids13;

private Device [] df;

private int [] m1;

private DataInputStream ids131;

private Device [] newStaff1=new Device[10];

static Device[] readData(BufferedReader in)

throws IOException

{

int n = Integer.parseInt(in.readLine());

Device [] e=new Device[n];

for(int i=0;i<n;i++)

{

e[i] = new Device();

e[i].readData(in);

}

return e;

}

}

class MyTimer1 extends JFrame

{

public MyTimer1()

{

setSize(650,350);

setTitle("His answer");

try

{

S2 =JOptionPane.showInputDialog("Введите логин искомого студента");

FileInputStream is12 = new

FileInputStream(new

File("C:\\Student\\"+S2+".txt"));

ids13 = new DataInputStream(is12);

String E=ids13.readUTF();

int f=Integer.parseInt(E);

df=new Device[f];

m1=new int [f];

FileInputStream is121 = new

FileInputStream(new

File("C:\\Student1\\"+S2+".txt"));

ids131 = new DataInputStream(is121);

for(int r123=0;r123<f;r123++)

{

String E1=ids131.readUTF();

int f1=Integer.parseInt(E1);

m1[r123]=f1;

}

BufferedReader in = new BufferedReader(new FileReader("device1.dat"));

newStaff1=readData(in);

in.close();

for(int g1=0;g1<f;g1++)

{

r3=m1[g1];

df[r34]=newStaff1[r3];

r34++;

}

label9= new JLabel("FFF");

for(int g1=0;g1<f;g1++)

{

String rrr=df[g1].getName();

hhh=hhh+"<br>"+rrr;

}

label9.setText("<html><br> Он правильно ответили на следующие темы(это - "+(f\*10)+"%)<br>"+hhh+"</html>");

Container contentPane = getContentPane ();

JPanel panel = new JPanel();

JPanel panel1 = new JPanel();

panel.add(label9);

replace2 = new JButton("OK");

panel1.add(replace2);

contentPane.add(panel,BorderLayout.CENTER);

contentPane.add(panel1,BorderLayout.SOUTH);

}

catch (IOException e)

{

System.out.print("Can not read file66666"+e);

}

replace2.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event2)

{

System.exit(0);

}

});

}

private JLabel label9;

private JButton replace2;

private String S2;

private String hhh=new String("");

private int r34=0;

private int r3=0;

private DataInputStream ids13;

private Device [] df;

private int [] m1;

private DataInputStream ids131;

private Device [] newStaff1=new Device[10];

static Device[] readData(BufferedReader in)

throws IOException

{

int n = Integer.parseInt(in.readLine());

Device [] e=new Device[n];

for(int i=0;i<n;i++)

{

e[i] = new Device();

e[i].readData(in);

}

return e;

}

}

class ButtonFrame11 extends JFrame

{

public ButtonFrame11()

{

Toolkit kit = Toolkit.getDefaultToolkit();

Dimension screenSize = kit.getScreenSize();

int screenHeight = screenSize.height;

int screenWidth = screenSize.width;

setSize(screenWidth/3,screenHeight/5);

setLocation(screenWidth/4,screenHeight/4);

Image img = kit.getImage("icon.gif");

setIconImage(img);

setTitle("CenteredFrame");

ButtonPanel11 panel = new ButtonPanel11();

add(panel);

}

}

class ButtonPanel11 extends JPanel

{

public ButtonPanel11()

{

JButton BeginButton = new JButton("Begin");

JButton ReButton = new JButton("Registration");

Password = new JPasswordField("",6);

login = new JTextField(15);

label1 = new JLabel("Please,input your name");

label2 = new JLabel("Please,input your password");

JPanel panel = new JPanel();

panel.setLayout(new GridLayout(2,2));

panel.add (label1);

panel.add (login);

panel.add (label2);

panel.add (Password);

add(panel,BorderLayout.CENTER);

JPanel panel1 = new JPanel();

panel1.add(BeginButton);

panel1.add (ReButton);

add(panel1,BorderLayout.SOUTH);

ColorAction11 BeginAction = new ColorAction11(2);

ColorAction11 ReAction = new ColorAction11(3);

BeginButton.addActionListener(BeginAction);

ReButton.addActionListener(ReAction);

}

private class ColorAction11 implements ActionListener

{

public ColorAction11(int c)

{

ccc=c;

t=0;

S1=new String("0");

S2=new String("1");

S3=new String("2");

S4=new String("3");

S5=new String("4");

S6=new String("5");

S7=new String("6");

S8=new String("7");

S9=new String("8");

S10=new String("9");

}

public void actionPerformed(ActionEvent event)

{

if(ccc==2)

{

S13 = login.getText();

pass = Password.getPassword();

S14=new String(pass,0,6);

try

{

FileInputStream is = new

FileInputStream(new

File("C:\\RGR\\"+S13+".txt"));

DataInputStream ids = new DataInputStream(is);

String S15=ids.readUTF();

if (S15.length() == S13.length())

{

if(S13.regionMatches(0, S15, 0, S15.length()))

{

String S16 = ids.readUTF();

if(S13.regionMatches(0, S11, 0, S11.length()))

{

t1=1;

}

if (S14.length() == S16.length())

{

if(S14.regionMatches(0, S16, 0, S16.length()))

{

if(S14.regionMatches(0, r, 0, r.length()))

{

t1=2;

}

FileOutputStream OS=new FileOutputStream (new File("C:\\USER\\NOW.txt"));

DataOutputStream ods = new DataOutputStream (OS);

ods.writeUTF(S13);

ods.close();

GregorianCalendar d = new GregorianCalendar();

int year=d.get(Calendar.YEAR);

int mouth=d.get(Calendar.MONTH);

int day=d.get(Calendar.DAY\_OF\_MONTH);

int hour=d.get(Calendar.HOUR);

int minute=d.get(Calendar.MINUTE);

int seconds=d.get(Calendar.SECOND);

String S111=""+year;

String S122=""+mouth;

String S133=""+day;

String S144=""+hour;

String S155=""+minute;

String S166=""+seconds;

FileOutputStream OS2=new FileOutputStream (new File("C:\\USER\\Time.txt"));

DataOutputStream ods2 = new DataOutputStream (OS2);

ods2.writeUTF(S111);

ods2.writeUTF(S122);

ods2.writeUTF(S133);

ods2.writeUTF(S144);

ods2.writeUTF(S155);

ods2.writeUTF(S166);

ods2.close();

if(t1==2)

{

FileOutputStream OS6=new FileOutputStream (new File("C:\\USER\\teacher.txt"));

DataOutputStream ods6 = new DataOutputStream (OS6);

ods6.writeUTF("1");

ods6.close();

}

else

{

FileOutputStream OS6=new FileOutputStream (new File("C:\\USER\\teacher.txt"));

DataOutputStream ods6 = new DataOutputStream (OS6);

ods6.writeUTF("0");

ods6.close();

}

ButtonFrame frame = new ButtonFrame ();

frame.setVisible(true);

}

}

}

}

else

{

System.out.print("You full!");

}

}

catch (IOException e)

{

System.out.print("Can not read file77777"+e);

}

}

if(ccc==3)

{

try{

t++;

S13 = login.getText();

System.out.println(S13);

pass = Password.getPassword();

System.out.println(pass);

S14=new String(pass,0,6);

File file = new File("C:\\RGR\\"+S13+".txt");

File file1 = new File("C:\\KURS\\"+S13+".txt");

File file2 = new File("C:\\Student\\"+S13+".txt");

File file3 = new File("C:\\Student1\\"+S13+".txt");

try

{

file.createNewFile();

file1.createNewFile();

file2.createNewFile();

file3.createNewFile();

}

catch(IOException e)

{

System.out.println("Can not create new file"+e);

}

FileOutputStream OS=new FileOutputStream (new File("C:\\RGR\\"+S13+".txt"));

DataOutputStream ods = new DataOutputStream (OS);

ods.writeUTF(S13);

ods.writeUTF(S14);

ods.close();

FileOutputStream OS1=new FileOutputStream (new File("C:\\KURS\\"+S13+".txt"));

DataOutputStream ods1 = new DataOutputStream (OS1);

ods1.writeUTF(S1);

ods1.writeUTF(S2);

ods1.writeUTF(S3);

ods1.writeUTF(S4);

ods1.writeUTF(S5);

ods1.writeUTF(S6);

ods1.writeUTF(S7);

ods1.writeUTF(S8);

ods1.writeUTF(S9);

ods1.writeUTF(S10);

ods1.close();

}

catch(IOException e)

{

System.out.println("Can not write file"+e);

}

}

}

private int ccc;

private char[] pass = new char[20];

private String S13;

private String S14;

private boolean dd;

private int t;

private int t1=0;

private String r="111111";

private String S1;

private String S2;

private String S3;

private String S4;

private String S5;

private String S6;

private String S7;

private String S8;

private String S9;

private String S10;

private String S11="lector";

}

private JPasswordField Password;

private JTextField login;

private JLabel label1;

private JLabel label2;

}

class CheckBoxFrame extends JFrame

{

public CheckBoxFrame()

{

setTitle("Test");

setSize(600,200);

Container contentPane = getContentPane ();

JPanel panel = new JPanel ();

label = new JLabel ("Внимания, вы проходите тестирование - не дурачится и внимательно вчитывайтесь в вопрос!");

contentPane.add(label,BorderLayout.NORTH);

replace1=new JButton("Finish");

replace = new JButton("Next");

panel.add(replace);

panel.add(replace1);

contentPane.add(panel,BorderLayout.SOUTH);

try

{

FileInputStream is45 = new

FileInputStream(new

File("C:\\USER\\NOW.txt"));

DataInputStream ids45 = new DataInputStream(is45);

S2=ids45.readUTF();

FileInputStream is = new

FileInputStream(new

File("C:\\USER\\Control.txt"));

ids = new DataInputStream(is);

}

catch(IOException e)

{

System.out.print("Can not read file888888"+e);

}

replace1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event2)

{

label.setText("Ваш результат - "+fff\*10+"% угаданых вопросов от общего количества вопросов");

}

});

replace.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent event2)

{

w=0;

c=0;

try

{

if(y<10)

{

S3=ids.readUTF();

label.setText("<html>Внимания, вы проходите тестирование - не дурачится и внимательно вчитывайтесь в вопрос!<br>"+S3);

}

}

catch(IOException e)

{

System.out.print("Can not read file5555"+e);

}

}

});

buttonPanel = new JPanel();

group = new ButtonGroup ();

addRadioButton("Дa",3);

addRadioButton("Частично",1);

addRadioButton("Нет",2);

addRadioButton("Другой ответ",0);

contentPane.add(buttonPanel,BorderLayout.CENTER);

try

{

FileInputStream is4 = new

FileInputStream(new

File("C:\\USER\\Answers.txt"));

ids4 = new DataInputStream(is4);

}

catch(IOException e)

{

System.out.print("Can not read file4444"+e);

}

}

public void addRadioButton(String name, final int size)

{

boolean selected = size==mode;

JRadioButton button = new JRadioButton(name,selected);

group.add(button);

buttonPanel.add(button);

try{

OS=new FileOutputStream (new File("C:\\Student1\\"+S2+".txt"));

ods = new DataOutputStream (OS);

}

catch (IOException e)

{

System.out.print("Can not read file3333"+e);

}

ActionListener listerner = new ActionListener()

{

public void actionPerformed(ActionEvent event1)

{

if(c==0)

{

if(y<10)

{

w++;

if(w==1)

{

try

{

String S31=ids4.readUTF();

r=Integer.parseInt(S31);

}

catch(IOException e)

{

System.out.print("Can not read file2222"+e);

}

if(r==size)

{

fff++;

try{

String X=""+y;

ods.writeUTF(X);

OS1=new FileOutputStream (new File("C:\\Student\\"+S2+".txt"));

ods1 = new DataOutputStream (OS1);

S3=""+fff;

ods1.writeUTF(S3);

}

catch (IOException e)

{

System.out.print("Can not read file111"+e);

}

}

}

}

}

y++;

}

};

button.addActionListener(listerner);

}

private JLabel label;

private JPanel buttonPanel;

private ButtonGroup group;

private static final int mode=0;

private int fff=0;

private int r=0;

private JButton replace;

private JButton replace1;

private int y=0;

private int c=1;

private String S3;

private String S2;

private DataInputStream ids4 ;

private DataInputStream ids;

private FileOutputStream OS;

private DataOutputStream ods;

private FileOutputStream OS1;

private DataOutputStream ods1;

private int w=0;

}

class JFrame1 extends JFrame

{

public JFrame1()

{

setSize(400,300);

ButtonPanel1 panel = new ButtonPanel1();

add(panel);

}

}

class ButtonPanel2 extends JPanel

{

public ButtonPanel2()

{

S133 =JOptionPane.showInputDialog("Введите логин искомого студента");

JButton Change = new JButton("Change");

m\_1=new JTextField(5);

m\_2=new JTextField(5);

m\_3=new JTextField(5);

m\_4=new JTextField(5);

m\_5=new JTextField(5);

m\_6=new JTextField(5);

m\_7=new JTextField(5);

m\_8=new JTextField(5);

m\_9=new JTextField(5);

m\_10=new JTextField(5);

JLabel l = new JLabel("Please, give me a logical speach, if you already want to change:");

JLabel l1 = new JLabel("Массив в С++: его значение, применение, свойства:");

JLabel l2 = new JLabel("Размер передаваемого массива");

JLabel l3 = new JLabel("Передача массивов как указателей");

JLabel l4 = new JLabel("Преимущества передачи массива как указателя");

JLabel l5 = new JLabel("Проблемы связаны с передачой массива...");

JLabel l6 = new JLabel("Некоторые особенности поведения компилятора");

JLabel l7 = new JLabel("Примеры правильного кода");

JLabel l8 = new JLabel("Массив обявленный как ссылка");

JLabel l9 = new JLabel("Использование абстрактного контейнера - это плохо?");

JLabel l10 = new JLabel("Коечто о шаблонах их работе и многом другом");

JPanel panel1 = new JPanel();

panel1.setLayout(new GridLayout(10,2));

add (l,BorderLayout.NORTH);

panel1.add (l1);

panel1.add (m\_1);

panel1.add (l2);

panel1.add (m\_2);

panel1.add (l3);

panel1.add (m\_3);

panel1.add (l4);

panel1.add (m\_4);

panel1.add (l5);

panel1.add (m\_5);

panel1. add (l6);

panel1.add (m\_6);

panel1.add (l7);

panel1.add (m\_7);

panel1.add (l8);

panel1.add (m\_8);

panel1.add (l9);

panel1.add (m\_9);

panel1.add (l10);

panel1. add (m\_10);

add(panel1,BorderLayout.CENTER);

add (Change,BorderLayout.SOUTH);

Change.addActionListener( new ActionListener(){public void actionPerformed(ActionEvent event)

{

try{

S2 = m\_1.getText();

S3 = m\_2.getText();

S4 = m\_3.getText();

S5 = m\_4.getText();

S6 = m\_5.getText();

S7 = m\_6.getText();

S8 = m\_7.getText();

S9 = m\_8.getText();

S10 = m\_9.getText();

S11 = m\_10.getText();

FileOutputStream OS=new FileOutputStream (new File("C:\\KURS\\"+S133+".txt"));

DataOutputStream ods = new DataOutputStream (OS);

ods.writeUTF(S2);

ods.writeUTF(S3);

ods.writeUTF(S4);

ods.writeUTF(S5);

ods.writeUTF(S6);

ods.writeUTF(S7);

ods.writeUTF(S8);

ods.writeUTF(S9);

ods.writeUTF(S10);

ods.writeUTF(S11);

ods.close();

}

catch(IOException e)

{

System.out.print("Can not read file99999"+e);

}

}

});

}

public ButtonPanel2(int e)

{

w=e;

m1[0]=0;

m1[1]=1;

m1[2]=2;

m1[3]=3;

m1[4]=4;

m1[5]=5;

m1[6]=6;

m1[7]=7;

m1[8]=8;

m1[9]=9;

}

private JTextField m\_1;

private JTextField m\_2;

private JTextField m\_3;

private JTextField m\_4;

private JTextField m\_5;

private JTextField m\_6;

private JTextField m\_7;

private JTextField m\_8;

private JTextField m\_9;

private JTextField m\_10;

private String S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11,S133;

protected int [] m1=new int[10];

int w=0;

public int[] getPrivios ()

{

return m1;

}

}

class JFrame2 extends JFrame

{

public JFrame2()

{

setSize(650,325);

ButtonPanel2 panel1 = new ButtonPanel2();

add(panel1);

}

}

class ButtonPanel1 extends JPanel

{

public ButtonPanel1()

{

JButton NextButton = new JButton("Next");

textArea = new JTextArea(12,35);

textArea.setEditable(false);

textArea.setLineWrap(true);

textArea.setWrapStyleWord(true);

scrollPane = new JScrollPane(textArea);

add(scrollPane,BorderLayout.CENTER);

add (NextButton);

S2 = new String();

try

{

FileInputStream is = new

FileInputStream(new

File("C:\\USER\\NOW.txt"));

DataInputStream ids = new DataInputStream(is);

S2=ids.readUTF();

FileInputStream is1 = new

FileInputStream(new

File("C:\\KURS\\"+S2+".txt"));

DataInputStream ids1 = new DataInputStream(is1);

String S30=ids1.readUTF();

String S31=ids1.readUTF();

String S32=ids1.readUTF();

String S33=ids1.readUTF();

String S34=ids1.readUTF();

String S35=ids1.readUTF();

String S36=ids1.readUTF();

String S37=ids1.readUTF();

String S38=ids1.readUTF();

String S39=ids1.readUTF();

d1[0]=Integer.parseInt(S30);

d1[1]=Integer.parseInt(S31);

d1[2]=Integer.parseInt(S32);

d1[3]=Integer.parseInt(S33);

d1[4]=Integer.parseInt(S34);

d1[5]=Integer.parseInt(S35);

d1[6]=Integer.parseInt(S36);

d1[7]=Integer.parseInt(S37);

d1[8]=Integer.parseInt(S38);

d1[9]=Integer.parseInt(S39);

}

catch (IOException e)

{

System.out.print("Can not read file0000"+e);

}

Color1Action NextAction = new Color1Action(d1);

NextButton.addActionListener(NextAction);

}

private class Color1Action implements ActionListener

{

public Color1Action(int[]o)

{

try{

BufferedReader in = new BufferedReader(new FileReader("device.dat"));

newStaff=readData(in);

in.close();

}

catch(IOException exception){

exception.printStackTrace();

}

for(int y=0;y<10;y++)

{

m[y]=o[y];

}

}

public void actionPerformed(ActionEvent event)

{

if(d<10)

{

i=m[d];

d++;

textArea.setText(newStaff[i].getName());

}

}

private int ccc1;

private String S1;

private int i=0;

private int d=0;

private int [] m= new int [10];

}

private JTextArea textArea;

private JScrollPane scrollPane;

private String S2;

private Device[] newStaff = new Device[10];

private int[] d1= new int[10];

static Device[] readData(BufferedReader in)

throws IOException

{

int n = Integer.parseInt(in.readLine());

Device [] e=new Device[n];

for(int i=0;i<n;i++)

{

e[i] = new Device();

e[i].readData(in);

}

return e;

}

}

class Device

{

private String name;

public Device() {}

public Device (String n) {

name=n;

}

public Device (Device e){

name=e.name;

}

public String getName () {

return name;

}

public void writeData(PrintWriter out) throws IOException

{

out.println(name+"|"+" ");

}

public void readData(BufferedReader in) throws IOException

{

String s=in.readLine();

StringTokenizer t= new StringTokenizer(s,"|");

name=t.nextToken();

}

}

**5. Структуры файлов:**

Структура файлов данной системы простая:

1. Она содержит единственный файл, содержащий код программы – Main.
2. Она использует 5 папок вспомогательных файлов, переменного количества файлов в каждой папке.
3. Папка RGR содержит файлы типа .txt – названия которых – это логины, а содержащаяся в них информация – пароли соответствующих пользователей.
4. Папка USER – содержит несколько файлов :

- NOW.txt – содержит имя пользователя, присутствующего в системе в данный момент.

- Time.txt – содержит время захода пользователя в систему;

- Control.txt – содержит вопросы тестирования;

- Answers.txt – содержит правильные ответы на вопросы тестирования;

- teacher.txt – содержит 0(если пользователь – учитель) или 1 (если пользователь – студент).

1. Папка KURS – содержит файлы последовательности вывода информации на экран для каждого пользователя в отдельности.
2. Папка Student – содержит номера разделов, на которые студент дал правильный ответ.
3. Папка Student1 – содержит количество правильных ответов студента

**6. Инструкция пользователя и контрольный пример:**

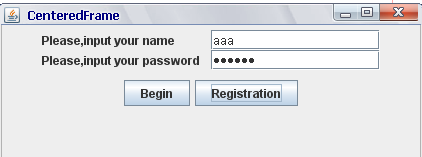
Все последующие пункты будут базироваться на следующих принципах:

1. Начало каждого пункта – теоретическая часть – это первый пункт инструкции, который дает информацию: что и почему мы это делаем.
2. Окончание – практическая часть – это первый пункт примера, который наглядно показывает действия описанные в теоретической части.

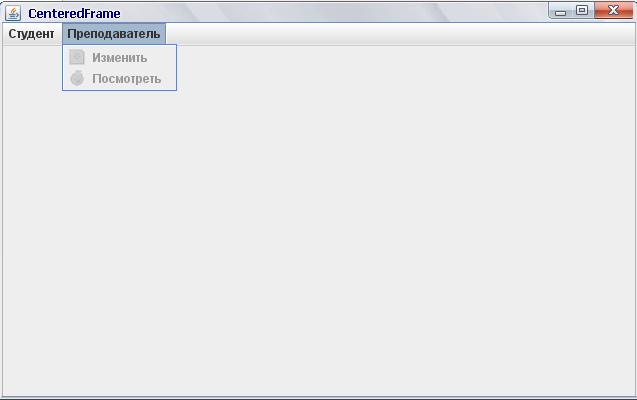
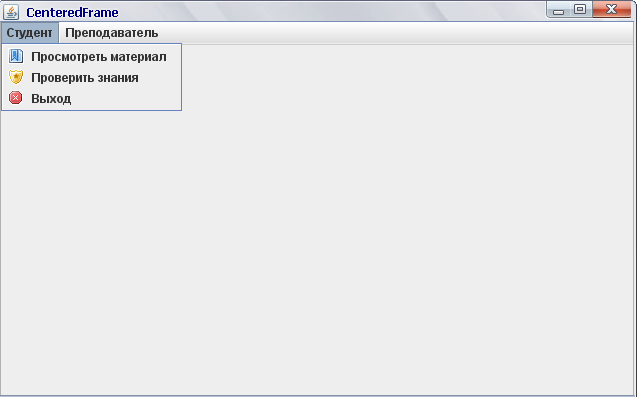
Сделано это было для удобства пользователя и повышения его восприятия описанной выше информации.

Пункты совместной инструкции и примера:

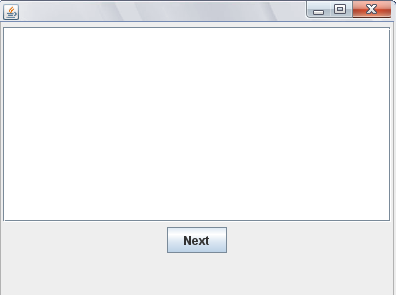
1. Сначала мы должны пройти регистрацию, и не просто пройти, а соблюсти некоторые правила. Если вы хотите зарегистрироваться как студент – у вас нет никаких ограничений, кроме шестизначного пароля, но если вы хотите зарегистрироваться как преподаватель, то вам придется ввести заданные значения полей логина и пароля, а именно – «lector» и «111111» соответственно. Это нужно, так как программа реализирует функции для студентов и преподавателей, а так как они не могут работать вместе, было решено их распараллелить в выборе доступных им функций. Для стабильной работы программы необходимо наличие хотя бы одного студента, поэтому мы сначала зарегистрируемся как студент:



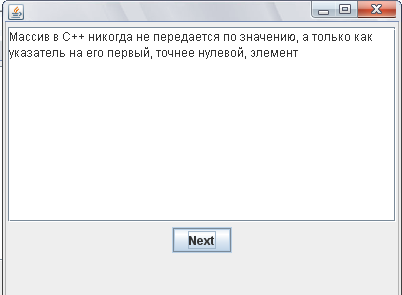
1. После нажатия кнопки Begin мы войдем в систему как студент ааа:



1. Как вы видите на рисунке наши действия как преподавателя в профиле студент невозможны, но возможны следующие действия: «Просмотреть материал» –

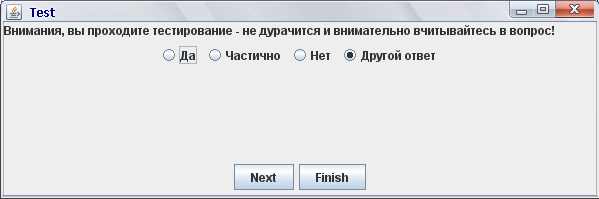


Как вы видите ничего нет пока мы не нажмем кнопку Next

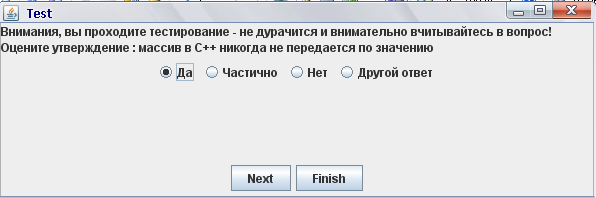


И таким образом мы можем просмотреть любое количество материала выложенное на освоение студентами.

1. «Проверить знание»:

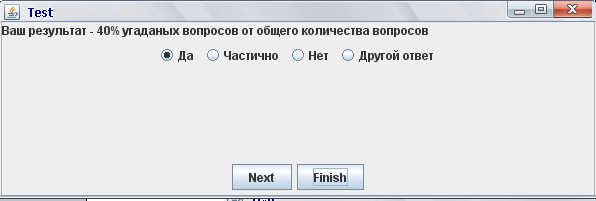


Чтобы правильно продолжить выполнение программы нужно нажать кнопку Next, не нажимая ничего другого:

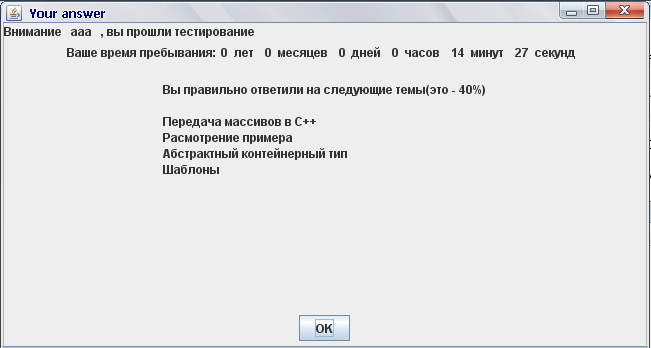


При выборе вариантов вы можете только один раз нажать любой предложенный вариант, остальные не будут засчитаны, кроме того вы должны нажать правильный ответ даже, если в нем при переходе, будет чёрная точка.

1. Наступит момент, когда нажатие кнопки Next не принесет каких-либо изменений в структуре окна, тогда нажмите кнопку Finish и вы получите результат:

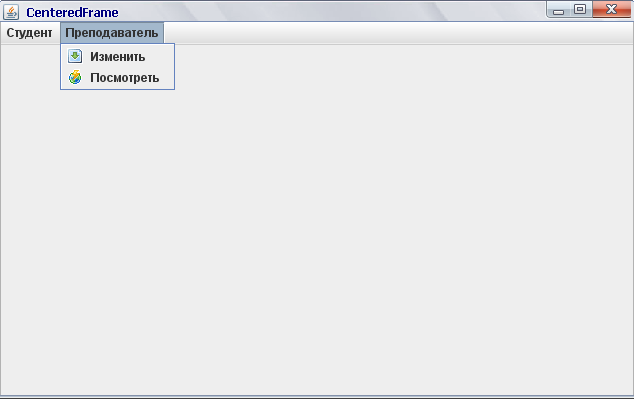
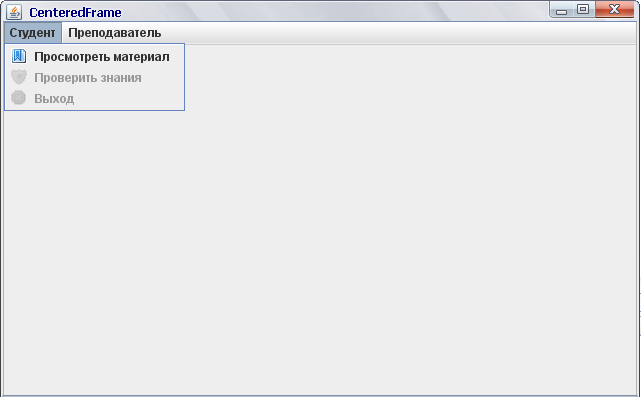


1. Только пройдя все пункты у вас возникает выбор: или повторить пункты 4 и 5 или воспользоваться функцией «Выход»:

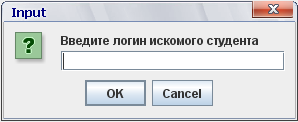


При нажатии кнопки ОК вы завершите программу.

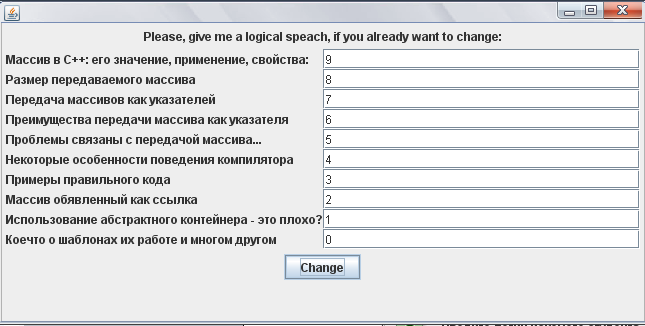
1. Теперь вы зайдите как преподаватель:



1. Вы видите, что часть функций студента прикрыта, но все остальные функции вам доступны: проверяем функцию «Изменить»:

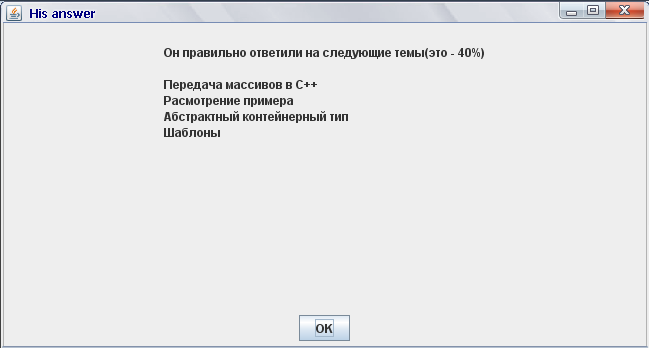


В этом окне вы должны ввести логин студента, которому вы хотели бы изменить порядок изложения материала:



Причем вы можете ввести любое целое число от 0 до 9.

1. Последняя функция – «Посмотреть»:



При нажатии на кнопку ОК вы выйдите из системы.

**Список использованной литературы:**

1. Методичні вказівки до курсової роботи «Обєктно-орієнтоване програмування» - Одеса, ОНПУ – 2007 р.;
2. Кей Хорстман, Гарри Корнел :«Том І. Основы» - Киев: «Вильямс» - 2007 год;
3. Кей Хорстман, Гарри Корнел :«Том 2. Тонкости програмирования» - Киев: «Вильямс» - 2007 год;