Міністерство освіти і науки України

Черкаський політехнічний технікум

Звіт з практики

з дисципліни «Об’єктно орієнтоване програмування»

Виконав студент ПС 4-1

Пісковий Денис Вадимович

Перевірив викладач

Коць О.І.

Черкаси 2016

Є клас „Група студентів” з полями

* назва групи,
* список студентів,
* список дисциплін,
* розклад занять,
* журнал відвідування,
* журнал оцінок.

та методами

* змінити назву групи,
* зарахувати студента в групу,
* відрахувати студента з групи,
* змінити розклад занять (аудиторію),
* змінити розклад занять (аудиторію, день, номер пари),
* змінити розклад занять (викладача),
* поставити відмітку про пропуск заняття (студенту з дисципліни),
* поставити оцінку в журналі оцінок (студенту з дисципліни).

Розробити клас-нащадок „Група випускників” з полями

* оцінка за переддипломну практику,
* оцінка за дипломне проектування,
* тема дипломного проекту,

та методами

* формувати виписку з диплому,
* оформити проходження переддипломної практики,
* захист дипломного проектування.

1. Лістинг програми:

**package** GroupStudents;

**public** **class** Studentgroup{

**static** String *nazva\_group* = "ПС 4-1";

**static** String *spisok\_studentiv* ="\n1. Пісковий Денис Вадимович \n"

+"2. Паршук Сергій Віталійович\n"

+"3. Підмогильний Олександр Сергійович";

**static** String *spisok\_disciplin* = "\n1. ІЗВП \n"

+"2. OOП\n"

+"3. КПЗ\n"

+"4. ОС\n"

+"5. Фізкультура";

**static** String *rozklad\_zanyat* = "\nПонеділок:\n"

+ "1. ІЗВП \n"

+ "2. ОС \n"

+ "3. КПЗ \n"

+ "\_-\_-\_-\_-\_-\_-\_-\n"

+ "Вівторок:\n"

+ "1. Фізкультура\n"

+ "2. ООП \n"

+ "3. ОС \n"

+ "4. ОС \n"

+ "\_-\_-\_-\_-\_-\_-\_-\n"

+ "Середа:\n"

+ "1. КПЗ \n"

+ "2. Фізкультура\n"

+ "3. ООП \n"

+ "\_-\_-\_-\_-\_-\_-\_-\n"

+ "Четвер: \n"

+ "1. ІЗВП \n"

"2. КПЗ \n"

+ "3. ООП \n"

+ "\_-\_-\_-\_-\_-\_-\_-\n"

+ "П'ятниця: \n"

+ "1. ООП \n"

+ "2. Фізкультура\n"

+ "3. КПЗ";

**public** **static** **void** main(String[] args) {

System.***out***.println("Назва групи "+ *nazva\_group*);

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println("Список студентів: "+ *spisok\_studentiv*);

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

System.***out***.println("Список дисциплін: "+ *spisok\_disciplin*);

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

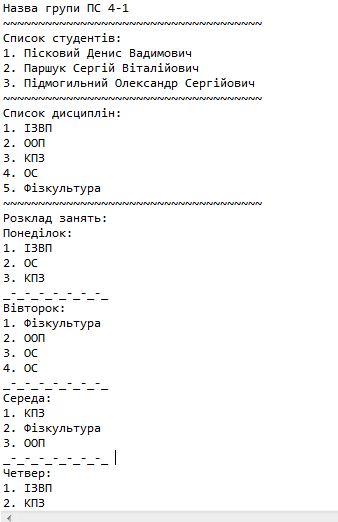
System.***out***.println("Розклад занять: "+ *rozklad\_zanyat*);

System.***out***.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

}

}

Результат:



**Варіант 6. Телефон**

Тема: Об'єктно-орієнтований аналіз і проектування програмного забезпечення. Програмне забезпечення вбудованого мікропроцесора телефону

Потрібно розробити засобами Rational Rose модель програмного забезпечення вбудованого мікропроцесора для апарату засновницької телефонної мережі.

Апарат підключений до лінії зв'язку, що веде до міні-АТС. У його завдання входить прийом і передача сигналів (в тому числі і голосових даних) міні-АТС. Апарат має кнопкову панель управління, екран для відображення списку номерів, дзвінок і трубку, в яку вбудовані мікрофон і гучномовець.

У початковому стані трубка телефону повішена, телефон не реагує на натискання кнопок. Телефон реагує тільки на сигнал «Виклик» від міні-АТС, при цьому включається дзвінок.

При знятті трубки на АТС подається сигнал «Трубка». При отриманні відповідного сигналу «Тон» від АТС телефон відтворює звуковий тон «Готовий» (довгий безперервний гудок) в трубку. При отриманні сигналу «Зайнято», в трубці відтворюється тон «Зайнято» (часті короткі гудки).

Користувач, чуючи в трубці тон «Готовий», набирає тризначний номер. Номер може бути набраний за допомогою кнопок з цифрами або натисканням на спеціальну кнопку «#». При натисканні на кнопку з цифрою відповідний їй сигнал «Цифра» передається АТС. Натискання на кнопки з цифрами після третього ігноруються. Під час набору номера введені цифри відображаються на екрані. Останній повністю набраний номер запам'ятовується в пам'яті апарату для того, щоб можна було його відтворити при натисканні на кнопку «#». При натисканні на цю кнопку номер з пам'яті апарату висвічується на екрані, і АТС передається послідовність з трьох сигналів «Цифра». У відповідь на набраний номер від АТС приходить або сигнал «Зайнято», або сигнал «Виклик». При отриманні сигналу «Виклик» телефон відтворює в трубку довгі гудки до того моменту, коли АТС здійснить комутацію і передасть сигнал «Дані».

Телефон відтворює дані, що передаються з сигналом, в трубку. Відповідь користувача сприймається мікрофоном трубки, перетворюється в сигнал «Дані» і передається АТС. Обмін даними переривається, якщо повішена трубка одного з телефонів, що беруть участь в обміні. Про те, що трубку повісив абонент, повідомляє сигнал «Зайнято», що посилається АТС. Після того, як трубка апарату була повішена, телефон посилає АТС сигнал «Кінець», і телефон переходить в початковий стан.

Лістинг програми:

package GroupStudents;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.text.\*;

public class CellPhone extends JFrame {

private final String WINDOW\_TITLE = "Phone";

private final int WINDOW\_WIDTH = 300;

private final int WINDOW\_HEIGHT = 422;

String input;

private JPanel textPanel = new JPanel();

private JPanel digitPanel = new JPanel();

private JPanel buttonPanel = new JPanel();

private JLabel displayLabel = new JLabel (" NOKIA ");

private JTextField displayTextField = new JTextField(20);

private JButton sendButton = new JButton("Зателефонувати");

private JButton clearButton = new JButton(" Очистити ");

private JButton endButton = new JButton("Завершити виклик");

private JButton redialButton = new JButton("Повторний виклик");

private JButton Button1 = new JButton("1");

private JButton Button2 = new JButton("2");

private JButton Button3 = new JButton("3");

private JButton Button4 = new JButton("4");

private JButton Button5 = new JButton("5");

private JButton Button6 = new JButton("6");

private JButton Button7 = new JButton("7");

private JButton Button8 = new JButton("8");

private JButton Button9 = new JButton("9");

private JButton Button0 = new JButton("0");

private JButton astrButton = new JButton("\*");

private JButton boundButton = new JButton("#");

private DecimalFormat df2 = new DecimalFormat("#,###.00");

public static void main(String[] args) {

new CellPhone();

}

public CellPhone()

{

setTitle(WINDOW\_TITLE);

setSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

buildTextPanel();

buildButtonPanel();

buildDigitPanel();

setLayout(new BorderLayout());

add(textPanel,BorderLayout.NORTH);

add(buttonPanel,BorderLayout.CENTER);

add(digitPanel,BorderLayout.SOUTH);

clearDisplay();

setVisible(true);

}

public void buildTextPanel () {

textPanel.setLayout(new BorderLayout());

textPanel.add(displayLabel,BorderLayout.NORTH);

textPanel.add(displayTextField,BorderLayout.CENTER);

textPanel.setBackground(Color.GREEN);

}

public void buildButtonPanel () {

buttonPanel.setLayout(new FlowLayout());

buttonPanel.add(sendButton);

buttonPanel.add(clearButton);

buttonPanel.add(endButton);

buttonPanel.add(redialButton);

sendButton.addActionListener(new SendButtonListener());

clearButton.addActionListener(new ClearButtonListener());

endButton.addActionListener(new EndButtonListener());

redialButton.addActionListener(new RedialButtonListener());

buttonPanel.setBackground(Color.RED);

}

public void buildDigitPanel () {

digitPanel.setLayout(new GridLayout(3,4));

digitPanel.add(Button1);

digitPanel.add(Button2);

digitPanel.add(Button3);

digitPanel.add(Button4);

digitPanel.add(Button5);

digitPanel.add(Button6);

digitPanel.add(Button7);

digitPanel.add(Button8);

digitPanel.add(Button9);

digitPanel.add(astrButton);

digitPanel.add(Button0);

digitPanel.add(boundButton);

InnerListener listener= new InnerListener();

Button1.addActionListener(listener);

Button2.addActionListener(listener);

Button3.addActionListener(listener);

Button4.addActionListener(listener);

Button5.addActionListener(listener);

Button6.addActionListener(listener);

Button7.addActionListener(listener);

Button8.addActionListener(listener);

Button9.addActionListener(listener);

Button0.addActionListener(listener);

astrButton.addActionListener(listener);

boundButton.addActionListener(listener);

}

private class SendButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

input= displayTextField.getText();

sendCalling();

}

}

private class ClearButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

clearDisplay();

}

}

private class EndButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

endCalling();

}

}

private class RedialButtonListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

input= displayTextField.getText();

redialCalling();

}

}

public void clearDisplay() {

displayTextField.setText(" ");

}

public void sendCalling() {

displayTextField.setText("Телефонуємо " + input+ input+ input+ "-" + input+ input+input+ "-" +input+ input+ input+ input);

}

public void endCalling() {

displayTextField.setText("Виклик завершено");

}

public void redialCalling() {

displayTextField.setText("Повторно викликаємо " + input+ input+ input+ "-" + input+ input+input+ "-" + input+ input+ input+ input );

}

private class InnerListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

if(e.getSource()==Button1) {

displayTextField.setText("1");

}

else if (e.getSource()==Button2) {

displayTextField.setText("2");

}

else if (e.getSource()==Button3) {

displayTextField.setText("3");

}

else if (e.getSource()==Button4) {

displayTextField.setText("4");

}

else if (e.getSource()==Button5) {

displayTextField.setText("5");

}

else if (e.getSource()==Button6) {

displayTextField.setText("6");

}

else if (e.getSource()==Button7) {

displayTextField.setText("7");

}

else if (e.getSource()==Button8) {

displayTextField.setText("8");

}

else if (e.getSource()==Button9) {

displayTextField.setText("9");

}

else if (e.getSource()==Button0) {

displayTextField.setText("0");

}

else if (e.getSource()==astrButton) {

displayTextField.setText("\*");

}

else if (e.getSource()==boundButton) {

displayTextField.setText("#");

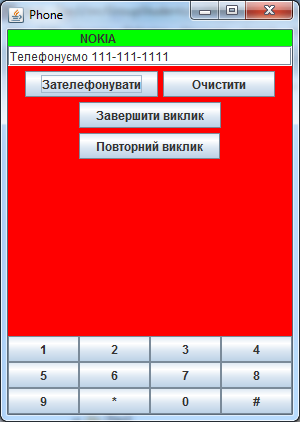
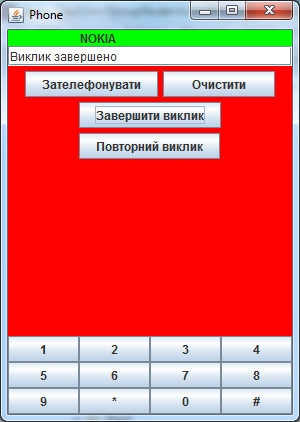
}

}

}

}

Результат:

Висновок: під час проходження практики я отримав нові навики роботи з Java та виконав своє завдання згідно варіанту.