Міністерств освіти і науки України  
Черкаський політехнічний технікум

Звіт з навчальної практики з предмету “Об’єктно-орієнтоване програмування”

Виконав студент групи ПС 4-1  
Шмиголь В.В.  
Перевірив викладач  
Коць О.І.

Черкаси 2016

**Варіант 10.**

**Завдання першого дня**

**Замок**

Тема: Об'єктно-орієнтований аналіз і проектування програмного забезпечення. Програмне забезпечення вбудованого процесора кодового замка

Потрібно розробити засобами Rational Rose модель програмного забезпечення вбудованого мікропроцесора для кодового замка, що регулює доступ в приміщення.

Кодовий замок складається з панелі з кнопками (цифри «0« ... «9», кнопка «Виклик», кнопка «Контроль»), цифрового дисплея, електромеханічного замка, дзвінка. Панель з кнопками встановлюється із зовнішнього боку дверей, замок встановлюється з внутрішньої сторони дверей, дзвінок встановлюється всередині приміщення, що охороняється.

У звичайному стані замок закритий. Доступ в приміщення здійснюється після набору коду доступу, що складається з чотирьох цифр. Під час набору коду введені цифри відображаються на дисплеї. Якщо код набраний правильно, то замок відкривається на деякий час, після чого двері знову закривається. Вміст дисплея очищається.

Кнопка «Виклик» використовується для подачі звукового сигналу всередині приміщення. Кнопка «Контроль» використовується для зміни кодів. Зміна коду доступу здійснюється наступним чином. При відкритих дверях потрібно набрати код контролю, що складається з чотирьох цифр, і новий код доступу. Для зміни коду контролю потрібно при відкритих дверях і, утримуючи кнопку «Виклик» набрати код контролю, після чого - новий код контролю.

**Лістинг програми:**

**package** day3;

**import** java.util.\*;

**public** **class** LockTester {

**public** **static** **class** Lock

{

**private** **boolean** nutural = **true**;

**private** **boolean** flag1, flag2, flag3;

**private** **int** one, two, three, five, four,six, seven,eight,nine, ten;

**private** **boolean** unlock = **true**;

**private** **int** position = 0;

**public** Lock (**int** one, **int** two, **int** three)

{

Scanner scan = **new** Scanner(System.***in***);

System.***out***.print("Введіть комбінацію з трьох чисел:");

**this**.one = one;

**this**.two = two;

**this**.three = three;

}

**public** **void** turnRight (**int** ticks) //

{

**if** (nutural)

{

position = (40 - ticks);

flag1 = **false**;

nutural = **false**;

**if** (position == one)

{

flag1 = **true**;

// System.out.println(position);

}

}

**else**

{

**if** (position - ticks > 0 )

{

position = position - ticks;

// System.out.println(position);

flag3 = **false**;

**if** (position == three)

{

flag3 = **true**;

}

}

**else**

{

position = (40 - (ticks - position));

flag3 = **false**;

// System.out.println(position);

**if** (position == three )

{

flag3 = **true**;

}

}

}

}

**public** **void** turnLeft (**int** ticks) // turns the dial left the given number of ticks

{

**if** (position + ticks < 40 )

{

position = (position + ticks);

// System.out.println(position);

flag2 = **false**;

**if** (position == two )

{

flag2 = **true**;

}

}

**else**

{

position = ( (position + ticks) - 40 );

// System.out.println(position);

flag2 = **false**;

**if** (position == two)

{

flag2 = **true**;

}

}

}

**public** **void** reset ()// âñòàíîâëþº ïåðåìèêà÷ íà 0

{

nutural = **true**;

flag1 = **false**;

flag2 = **false**;

flag3 = **false**;

}

**public** **void** open ()

{

**if** (!(flag1 && flag2 && flag3))

{

unlock = **false**;

System.***out***.println("Не правильна комбінація!");

}

**else**

{

System.***out***.println("Комбінація введена вірно.Замок відкрито!");

}

}

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

{

Lock lock = **new** Lock(35, 12, 37);

lock.turnRight(5);

lock.turnLeft(17);

lock.turnRight(15);

lock.open();

}

}

}

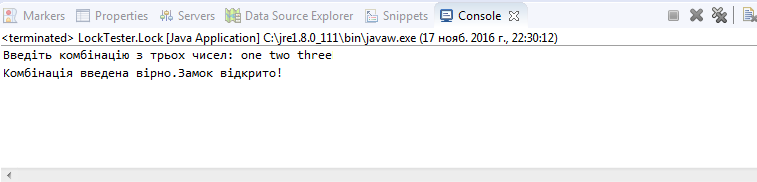
Результат роботи:

Рисунок 1. Виконання програми

**Завдання другого дня**

**Пацієнт**

Є клас „Пацієнт” з полями

* ПІБ
* вік
* стать
* флюорографія
* медичний огляд

та методами

* записати результат флюорографії,
* записати дату останньої флюорографії
* записати дату останнього медогляду
* записати висновок останнього медогляду

Розробити інтерфейс реєстрації результатів відвідування поліклініки з кнопками „Флюрографія”, „Медичний огляд”, полями введення даних про відвідувача поліклініки та текстовою областю з інформацію про кількість пацієнтів, що пройшли флюорографію та медогляд.

Лістинг програми:

**package** pacient;

**import** javax.swing.\*;

**import** java.awt.\*;

**public** **class** pacient {

**private** JTextField displayField;

**public** **void** setDisplayValue(String val) {

displayField.setText(val);

}

**public** String getDisplayValue() {

**return** displayField.getText();

}

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

JPanel p1;

JButton button0;

JPanel windowContent = **new** JPanel();

JTextField displayField;

button0 = **new** JButton("Внести данні");

p1 = **new** JPanel();

JLabel label1 = **new** JLabel("ПІБ");

JTextField field = **new** JTextField(10);

JLabel label = **new** JLabel("Вік");

JTextField field1 = **new** JTextField(10);

JLabel label2 = **new** JLabel("Стать");

JTextField field2 = **new** JTextField(10);

JLabel label3 = **new** JLabel("Флюорографія");

JTextField field3 = **new** JTextField(10);

JLabel label4 = **new** JLabel("Медичний огляд");

JTextField field4 = **new** JTextField(10);

JLabel label5 = **new** JLabel("Результат флюорографії");

JTextField field5 = **new** JTextField(10);

JLabel label6 = **new** JLabel("Дата останньої флюорографії");

JTextField field6 = **new** JTextField(10);

windowContent.add(label1);

windowContent.add(field1);

windowContent.add(label2);

windowContent.add(field2);

windowContent.add(label3);

windowContent.add(field3);

windowContent.add(label4);

windowContent.add(field4);

windowContent.add(label5);

windowContent.add(field5);

windowContent.add(label6);

windowContent.add(field6);

p1 = **new** JPanel();

GridLayout gl = **new** GridLayout(15, 30);

p1.setLayout(gl);

windowContent.add(button0);

JFrame frame = **new** JFrame("PAY");

frame.setContentPane(windowContent);

frame.setSize(1500, 400);

frame.setVisible(**true**);

}

}

Результат роботи:

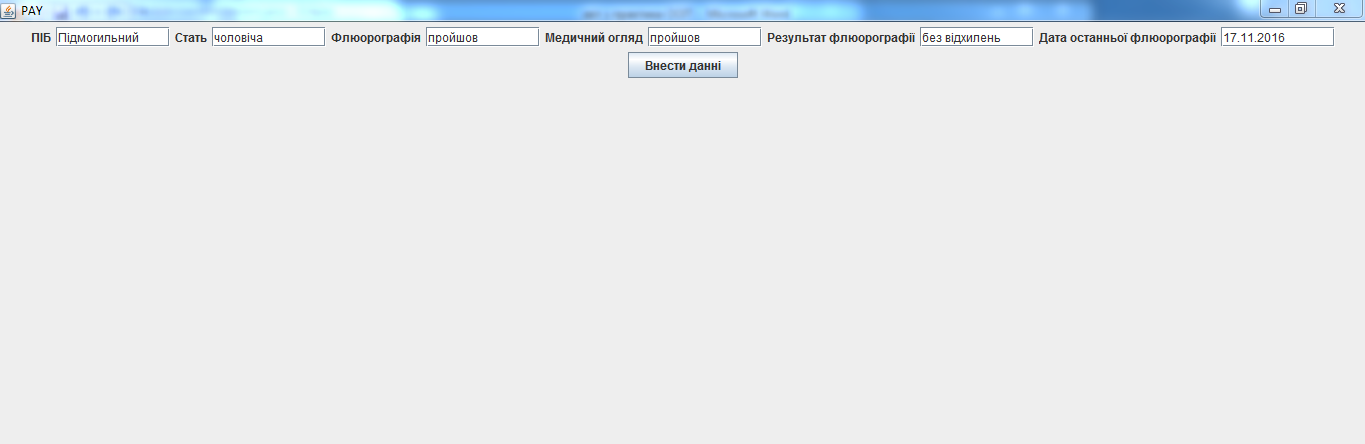


Рисунок 2. Результат виконання програми

Висновок: на даній практиці я використав всі отримані знання з об’єктно-орієнтованого програмування та застосував їх у розробленні програм за певними завданнями.