НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Інженерія програмного забезпечування»

*Звiт з лабораторної роботи №1*

*Тема: «Стекові об’єкти. Конструктор, деструктор, відображення, передача»*

Виконав:

ст. гр. КІТ-25а

Шиян А.Н.

Перевірив:

Молчанов Г.I.

Харків – 2017

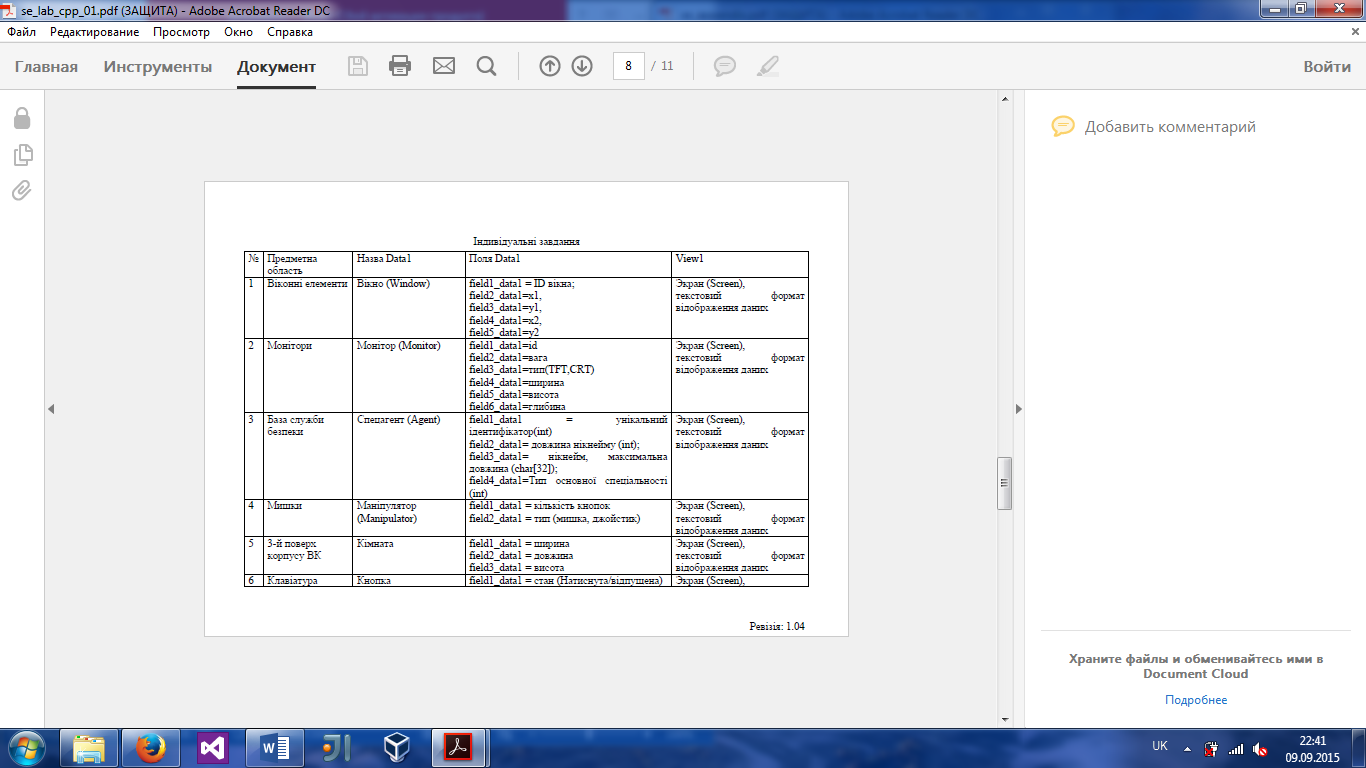
Мета:

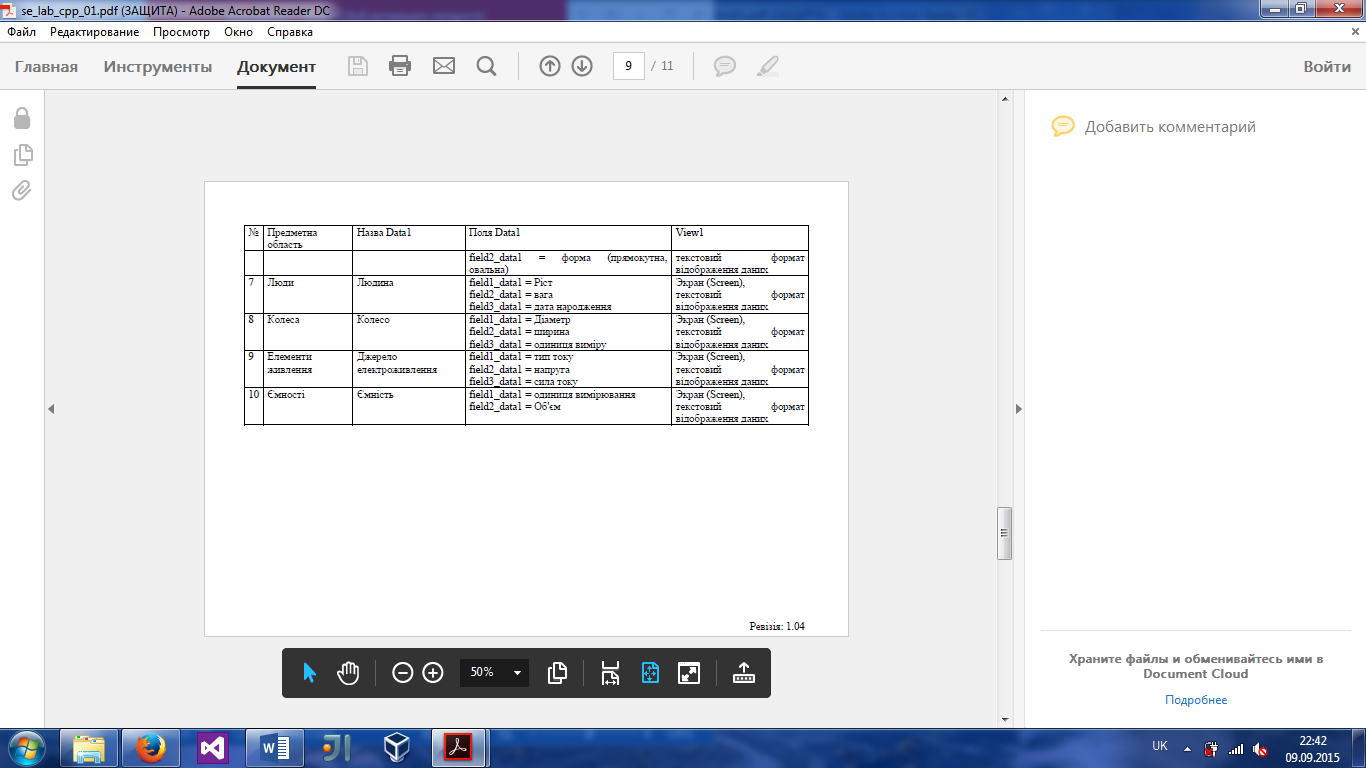
* навчитись створювати об’єкти.
* отримати розуміння створення об’єкта на стеку, а також передачу об’єкта по значенню.

1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Створити клас даних Button та клас відображення даних — Screen. Об’єкт відображення конструюється на стеку функції main() об`єктом даних, що заздалегідь створений на стеку. Передавати Button як значення.

Button має всі публічні поля та методи. Лише виконує відображення даних у форматі назва поля=значення; всі його методи та атрибути публічні.





2 РОЗРОБКА ПРОГРАМИ

Для реалізації програми було розроблено два класи: Button та Screen.

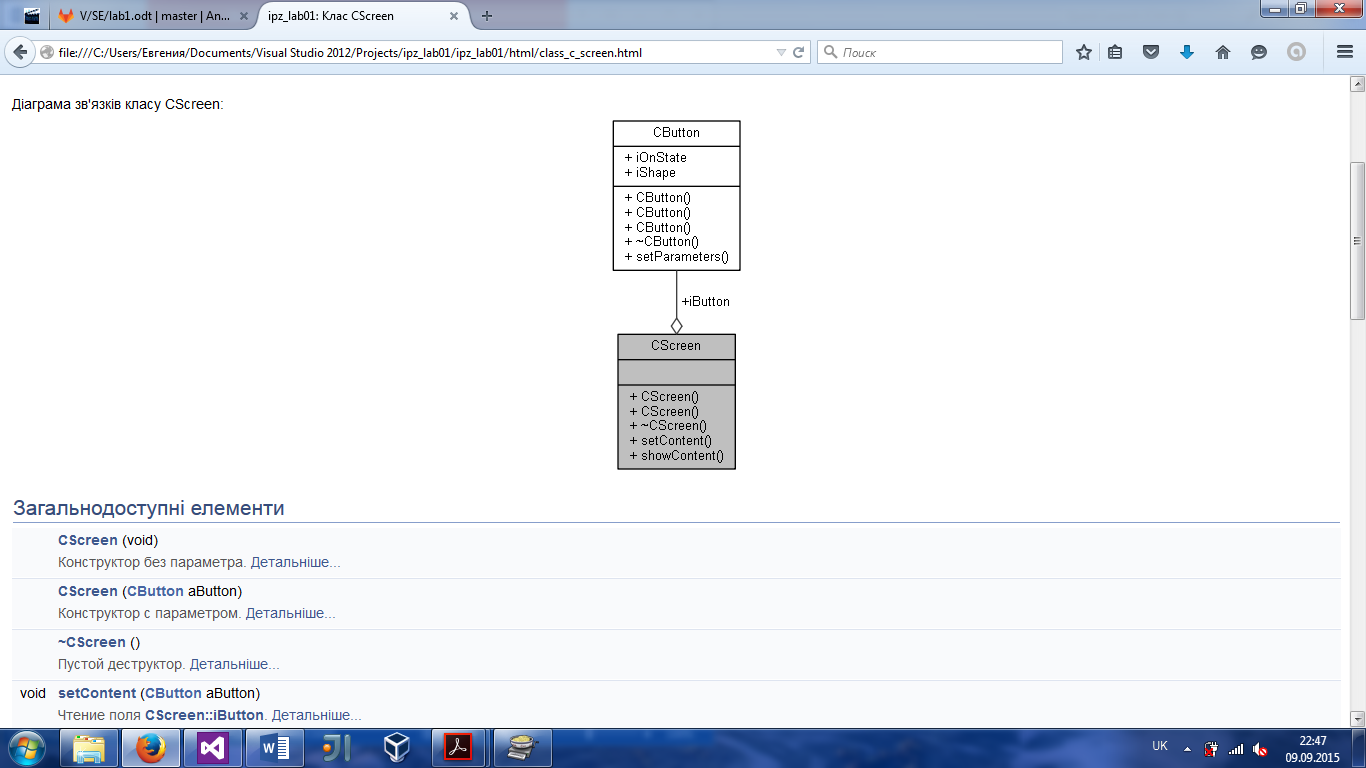
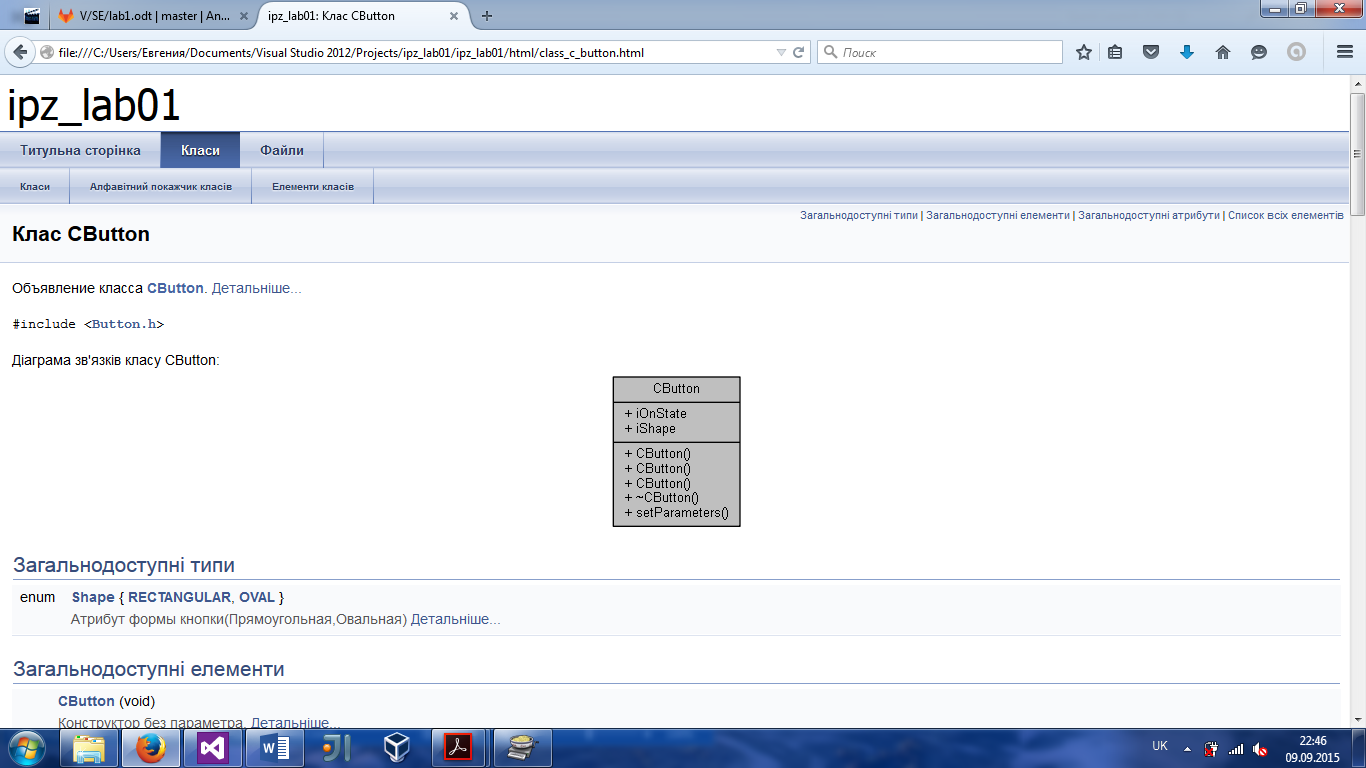
 2.1 Ієрархія та структура класів

Рисунок 2.1 — Ієрархія класів

2.2 Опис програми

На рис.2.2 наведена структура розробленого проекту

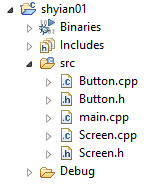


Рисунок 2.2 — Структура проекту

Призначення спроектованих класів наведено на рис. 2.3.

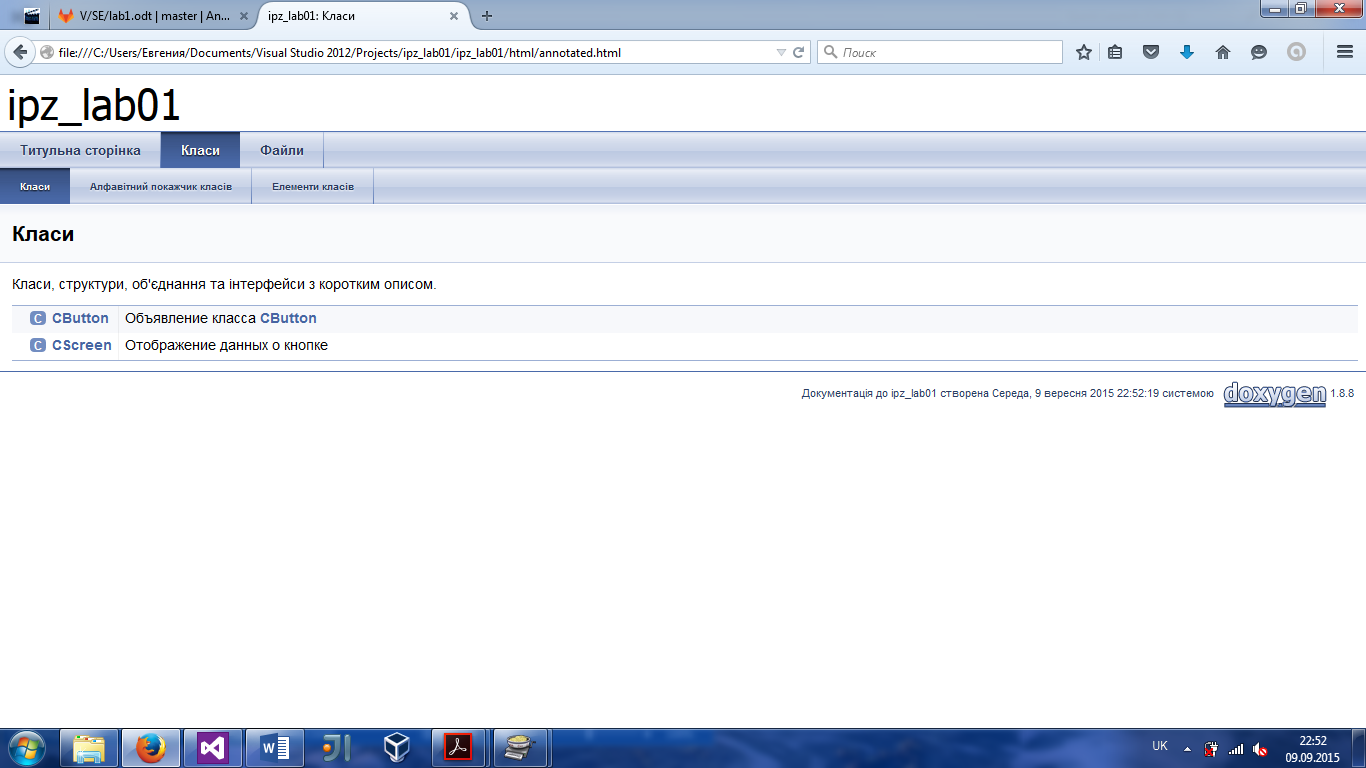


Рисунок 2.3 — Призначення класів, створене за допомогою Javadoc

2.3 Фрагменти програми

2.3.1 Файл Button.h

/\*\*

\* @file Button.h

\* Объявление класса Button

\* @author Shyian

\* @version 1.0.0

\* @date 11.11.2017

\*/

#ifndef BUTTON\_H

#define BUTTON\_H

/\*\*

\* Создание объекта класса CButton

\*/

class CButton

{

public:

/// Атрибут состояния кнопки(вкл/выкл)

bool iOnState;

///Атрибут формы кнопки(Прямоугольная,Овальная)

enum Shape { RECTANGULAR, OVAL } iShape;

/\*\*

\* Конструктор без параметра.

\*/

CButton(void);

/\*\*

\* Конструктор с двумя параметрами.

\* @param aOnState Указывается в качестве первого параметра метода CButton::SetParameters

\* @param aShape Указывается в качестве второго параметра метода CButton::SetParameters

\*/

CButton(bool aOnState, Shape aShape);

/\*\*

\* Конструктор копирования.

\* @param aButton Исходный объект с данными для копирования

\*/

CButton(const CButton& aButton);

/\*\*

\* Пустой деструктор.

\*/

~CButton();

/\*\*

\* Чтение полей CButton::iOnState и CButton::iShape

\* @param aOnState Текущее значение поля CButton::iOnState

\* @param aShape Текущее значение поля CButton::iShape

\*/

void setParameters(bool aOnState, Shape aShape);

};

#endif

2.3.2 Файл Button.cpp

/\*\*

\* @file Button.cpp

\* Реализация класса Button

\* @author Shyian

\* @version 1.0.0

\* @date 11.11.2017

\*/

#include "Button.h"

CButton::CButton()

{

iOnState = false;

iShape = RECTANGULAR;

}

CButton::CButton(bool aOnState, Shape aShape)

{

setParameters(aOnState, aShape);

}

CButton::CButton(const CButton& aButton)

{

setParameters(aButton.iOnState,aButton.iShape);

}

CButton::~CButton()

{

// OutputDebugString("Destructor is called! " );

}

void CButton::setParameters(bool aOnState, Shape aShape)

{

iOnState = aOnState;

iShape = aShape;

}

2.3.3 Файл Screen.h

/\*\*

\* @file Screen.h

\* Объявление класса Screen

\* @author Shyian

\* @version 1.0.0

\* @date 11.11.2017

\*/

#include "Button.h"

/\*\*

\* Отображение данных о кнопке

\*/

class CScreen

{

public :

///Атрибут объекта кнопки

CButton iButton;

/\*\*

\* Конструктор без параметра.

\*/

CScreen(void);

/\*\*

\* Конструктор с параметром.

\* @param aButton Указывается в качестве параметра метода CScreen::SetContent

\*/

CScreen(CButton aButton);

/\*\*

\* Пустой деструктор.

\*/

~CScreen();

/\*\*

\* Чтение поля CScreen::iButton

\* @param aButton Текущее значение поля CButton::iButton

\*/

void setContent(CButton aButton);

/\*\*

\* Отображение данных кнопки.

\*/

void showContent(void) ;

};

2.3.4 Файл Screen.cpp

/\*\*

\* @file CScreen.cpp

\* Реализация класса CScreen

\* @author Shyian

\* @version 1.0.0

\* @date 11.11.2017

\*/

#include "Screen.h"

#include "Button.h"

#include <iostream>

using namespace std;

CScreen::CScreen(void)

{

iButton.iOnState = false;

iButton.iShape = iButton.RECTANGULAR;

}

CScreen::CScreen(CButton aButton) {

setContent(aButton);

}

CScreen::~CScreen()

{

// OutputDebugString("Destructor is called! " );

}

void CScreen::setContent(CButton aButton)

{

iButton.iOnState = aButton.iOnState;

iButton.iShape = aButton.iShape;

}

void CScreen::showContent(void)

{

if(iButton.iShape == 1) {

cout << "Shape : Oval " <<endl;

}

if(iButton.iShape == 0) {

cout << "Shape : Rectangular" <<endl;

}

cout << "On State : " << iButton.iOnState <<endl;

}

2.3.5 Файл main.cpp

/\*\*

\* @file main.cpp

\* Реализация функции main().

\* @author Shyian

\* @version 1.0.0

\* @date 11.11.2017

\*/

#include "Button.h"

#include "Screen.h"

#include "conio.h"

/\*\*

\* Точка входа программы.

\* @return код завершения

\*/

int main(void)

{

CButton button;

button.setParameters(true,button.RECTANGULAR);

CScreen view1(button);

view1.showContent();

CButton\* secondButton = new CButton();

secondButton->setParameters(false, secondButton->RECTANGULAR);

CScreen view2(\*secondButton);

view2.showContent();

CButton\* thirdButton = new CButton();

thirdButton->setParameters(true,thirdButton->OVAL);

CScreen view3(\*thirdButton);

view3.showContent();

\_getch();

return 0;

}

3 РЕЗУЛЬТАТИ РОБОТИ

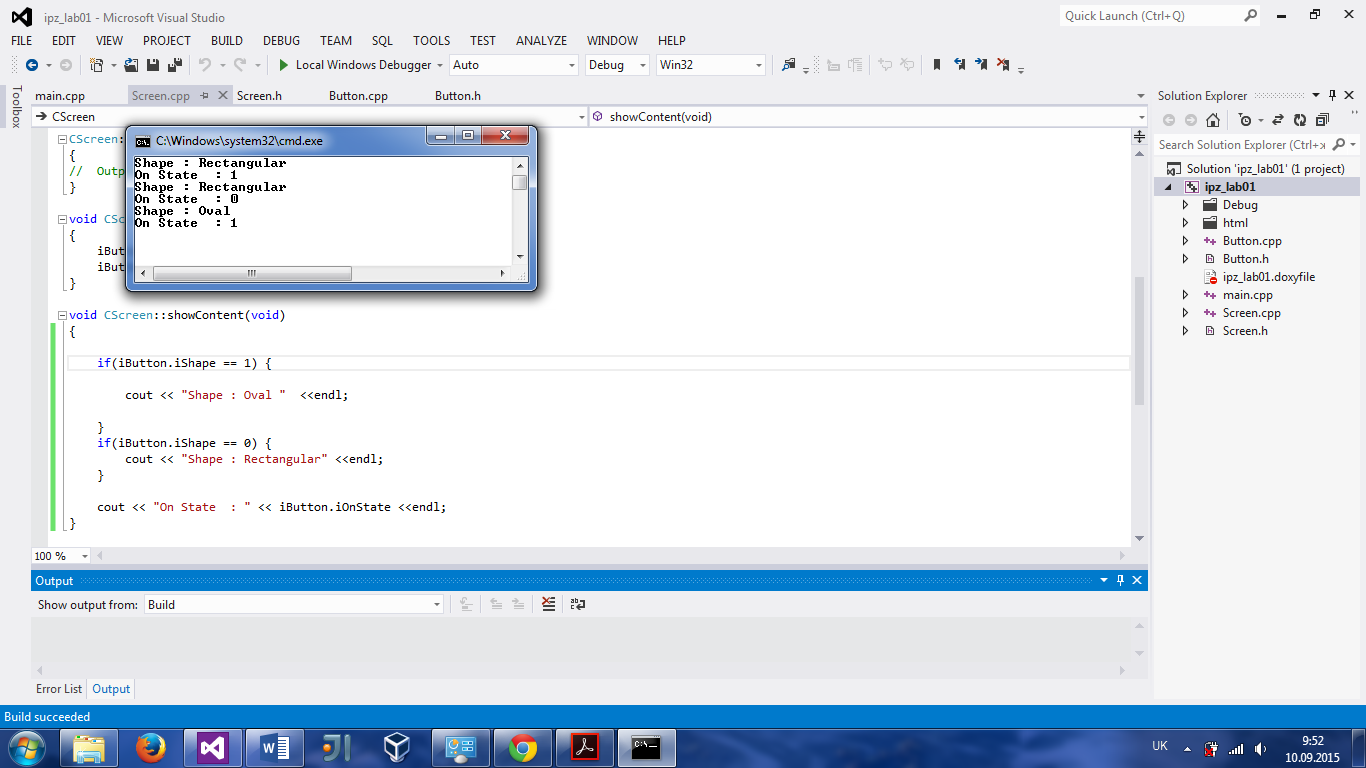


Рисунок 3.1 - Приклад роботи програми

ВИСНОВКИ

В розробленій програмі реалізовано класи Button та Screen. Button — клас для збереження інформації про кнопки, Screen — для відображення інформації про них.