Разработка библиотеки классов

Необходимо разработать библиотеку классов, воспользуйтесь файлом «Создание\_dll\_библиотеки.pdf». Данная библиотека будет подключаться к основному проекту и должна быть представлена в виде .dll файла. Чтобы система правильно интегрировалась вам необходимо обязательно следовать правилам именования библиотек, классов и методов в них. Классы и методы должны содержать модификатор public, чтобы внешние приложения могли получить к ним доступ. Необходимо решить какие виды расчетов будут важны для Вашей предметной области и реализовать этот расчет в виде библиотеки.

Создала библиотеку:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.Mime.MediaTypeNames;

namespace Class

{

public class Class1

{

public static int numCAPTCHA = 0;

public static string text = String.Empty;

public static Bitmap CreateImage(ref string text1, int Width, int Height)//!

{

Random rnd = new Random();

//Создадим изображение

Bitmap result = new Bitmap(Width, Height);

//Вычислим позицию текста

int Xpos = rnd.Next(0, Width - 50);

int Ypos = rnd.Next(15, Height - 15);

//Добавим различные цвета

Brush[] colors = { Brushes.Black,

Brushes.Red,

Brushes.RoyalBlue,

Brushes.Green };

//Укажем где рисовать

Graphics g = Graphics.FromImage((System.Drawing.Image)result);

//Пусть фон картинки будет серым

g.Clear(Color.White);

//Сгенерируем текст

text = String.Empty;

string ALF = "1234567890QWERTYUIOPASDFGHJKLZXCVBNM";

for (int i = 0; i < 5; ++i)

text += ALF[rnd.Next(ALF.Length)];

//Нарисуем сгенирируемый текст

g.DrawString(text,

new Font("Arial", 15),

colors[rnd.Next(colors.Length)],

new PointF(Xpos, Ypos));

//Добавим немного помех

/////Линии из углов

g.DrawLine(Pens.Black,

new Point(0, 0),

new Point(Width - 1, Height - 1));

g.DrawLine(Pens.Black,

new Point(0, Height - 1),

new Point(Width - 1, 0));

////Белые точки

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

for (int j = 0; j < Height; ++j)

if (rnd.Next() % 20 == 0)

result.SetPixel(i, j, Color.Gray);

text1 = text;

return result;

}

public static string CAPTCHA(Timer timer1, Button button4, Form form)

{

numCAPTCHA++;

MessageBox.Show("Символы введены неверно.", "Ошибка!",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

if (numCAPTCHA == 2)

{

MessageBox.Show("Слишком много неудачных попыток. Блокировка системы на 3 минуты.", "Ошибка!",

MessageBoxButtons.OK,

MessageBoxIcon.Information);

timer1.Interval = 1000;

button4.Enabled = false;

timer1.Start();

}

else if (numCAPTCHA > 2)

{

DialogResult result = MessageBox.Show("Слишком много неудачных попыток. Полная блокировка системы. Перезапустить систему?", "Ошибка!",

MessageBoxButtons.YesNo,

MessageBoxIcon.Information);

if (result == DialogResult.Yes)

{

System.Windows.Forms.Application.Restart();

form.TopMost = true;

}

form.TopMost = true;

form.Enabled = false;

}

return text;

}

public static string Load(int d)

{

string name = "Data Source=ADCLG1;Initial Catalog=CherEPractica;Integrated Security=True";

SqlConnection MyConnection = new SqlConnection(name);

MyConnection.Open();

string sql = $"SELECT Id\_type from Users where UserID=\'{d}\'";

SqlCommand cmd1 = new SqlCommand(sql, MyConnection);

int k1 = Convert.ToInt32(cmd1.ExecuteScalar().ToString());

MyConnection.Close();

if (k1 == 4 || k1 == 2)

{

return "Мастер или клиент";

}

else

{

return "оператор";

}

}

}

}

Подключила к основному проекту:





Использование библиотеки:

….

private string text = String.Empty;//!

// private int numCAPTCHA = 0;//!

string name = "Data Source=ADCLG1;Initial Catalog=CherEPractica;Integrated Security=True";

//string name = "Data Source=LAPTOP-E9A534AD;Initial Catalog=CherEPractica;Integrated Security=True";

public Form1()

{

InitializeComponent();

MaximumSize = new System.Drawing.Size(404, 310);

MinimumSize = new System.Drawing.Size(404, 310);

pictureBox1.Image = Class1.CreateImage(ref text, pictureBox1.Width, pictureBox1.Height);//!

this.TopMost = true;//!

}

…

private void button4\_Click(object sender, EventArgs e)

{

if (textBox3.Text == this.text)

{

//numCAPTCHA = 0;

button1.Enabled = true;

MaximumSize = new System.Drawing.Size(404, 310);

MinimumSize = new System.Drawing.Size(404, 310);

}

else

{

text=Class1.CAPTCHA(timer1, button4, this);

}

}

…

Тестирование приложения

Подготовьте Test-Case (хотя бы 1 тест на 1 функцию) **минимум 5 штук**, проверяющие один наиболее важный на ваш взгляд процесс в системе. Важно, чтобы тестовые данные предусматривали различные ситуации. При оформлении используйте представленный ниже шаблон.

Аннотация теста:

|  |  |
| --- | --- |
| **Название проекта** | Rab\_5 |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Червонцева Елизавета |
| **Дата(ы) теста** | 05.11.2024 |

Расшифровка тестовых информационных полей:

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| **Название проекта** | Название тестируемого проекта |
| **Рабочая версия** | Версия проекта/программного обеспечения (первый тест считается 1.0). |
| **Имя тестирующего** | Имя того, кто проводил тесты |
| **Дата(ы) теста** | Дата(ы) проведения тестов – это один или несколько дней. Если тесты проводились в более протяженный период времени, нужно отметить отдельную дату для каждого теста. |
| **Тестовый пример #** | Уникальный ID для каждого тестового примера. Следуйте некоторым конвенциям, чтобы указать типы тестов. Например,‘TC\_UI\_1′ означает‘user interface test case #1′ ( ТС\_ПИ\_1: тестовый случай пользовательского интерфейса#1) |
| **Приоритет тестирования**  *(Низкий/Средний/Высокий)* | Насколько важен каждый тест. Приоритет тестирования для бизнес-правил и функциональных тестовых случаев может быть средним или высоким, в то время как незначительные случаи пользовательского интерфейса могут иметь низкий приоритет. |
| **Заголовок/название теста** | Название тестового случая. Например, Подтвердите страницу авторизации с действительным именем пользователя и паролем. |
| **Краткое изложение теста** | Описание того, что должен достичь тест. |
| **Этапы теста** | Перечислите все этапы теста подробно. Запишите этапы теста в том порядке, в котором они должны быть реализованы. Предоставьте как можно больше подробностей и разъяснений. Пронумерованный список – хорошая идея. |
| **Тестовые данные** | Перечислите/опишите все тестовые данные, используемые для данного тестового случая. Так, фактические используемые входные данные можно отслеживать по результатам тестирования. Например, Имя пользователя и пароль для подтверждения входа. |
| **Ожидаемый результат** | Каким должен быть вывод системы после выполнения теста? Подробно опишите ожидаемый результат, включая все сообщения/ошибки, которые должны отображаться на экране. |
| **Фактический результат** | Каким должен быть фактический результат после выполнения теста? Опишите любое релевантное поведение системы после выполнения теста. |
| **Предварительное условие** | Любые предварительные условия, которые должны быть выполнены до выполнения теста. Перечислите все предварительные условия для выполнения этого тестового случая. |
| **Постусловие** | Каким должно быть состояние системы после выполнения теста? |
| **Статус**  *(Зачет/Незачет)* | Если фактический результат не соответствует ожидаемому результату, отметьте тест как неудачный. В ином случае обновление пройдено. |
| **Примечания/комментарии** | Используйте эту область для любых дополнительных заметок/комментариев/вопросов. Эта область предназначена для поддержки вышеуказанных полей (например, если есть некоторые особые условия, которые не могут быть описаны в любом из вышеуказанных полей, или если есть вопросы, связанные с ожидаемыми или фактическими результатами). |

Тестовый пример #1:

|  |  |
| --- | --- |
| **Тестовый пример #** | UserTest1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Войдите в систему с действительными логином и паролем заказчика |
| **Краткое изложение теста** | Войти в приложение, ввести в поле логина логин из тестовых данных, ввести пароль в поле пароля из тестовых данных, нажать кнопку войти. |
| **Этапы теста** | 1. Войти в приложение 2. Ввести действующие логин и пароль заказчика 3. Нажать кнопку Войти |
| **Тестовые данные** | Логин: login4  Пароль: pass4 |
| **Ожидаемый результат** | Открывается главное меню заказчика. |
| **Фактический результат** | Открылось главное меню заказчика. |
| **Статус** | Зачет |
| **Предварительное условие** | Убедиться, что пользователь с данными логином и паролем существует. |
| **Постусловие** | Доступен функционал заказчика |
| **Примечания/комментарии** |  |

Тестовый пример #2:

|  |  |
| --- | --- |
| **Тестовый пример #** | UserTest2 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Оформление заявки. |
| **Краткое изложение теста** | Войти в систему, ввести в поле логина логин из тестовых данных, ввести пароль в поле пароля из тестовых данных, нажать кнопку войти. Нажать на кнопку Заявок, после этого на кнопку Добавление Заявок. В открывшемся окне заполнить тип модели, модель и проблему. Нажать на кнопку Добавить заявку. |
| **Этапы теста** | 1. Войти в приложение 2. Ввести действующие логин и пароль заказчика 3. Нажать кнопку Войти 4. Нажать кнопку Заявки 5. Нажать кнопку Добавление Заявки 6. Заполнить поле тип модели, модель и проблема 7. Нажать кнопку Добавить заявку |
| **Тестовые данные** | Логин: login4  Пароль: pass4  Тип модели: Фен  Модель: Ладомир ТА112 белый  Проблема: Перестал работать |
| **Ожидаемый результат** | В поле таблицы появляется новая строка с введенными данными заказчика. |
| **Фактический результат** | В поле таблицы появляется новая строка с введенными данными заказчика. |
| **Статус** | Зачет |
| **Предварительное условие** | Войти в систему. Убедиться что существует тип модели Фен, модель Ладомир ТА112 белый и проблема Перестал работать. |
| **Постусловие** | Доступна форма заявок с функционалом заказчика. Созданная заявка доступна для других пользователей. |
| **Примечания/комментарии** |  |

Тестовый пример #3:

|  |  |
| --- | --- |
| **Тестовый пример #** | UserTest3 |
| **Приоритет тестирования** | низкий |
| **Заголовок/название теста** | Просмотр истории входа в приложение под введенным логином |
| **Краткое изложение теста** | Войти в приложение, ввести в поле логина логин из тестовых данных, ввести пароль в поле пароля из тестовых данных, нажать кнопку войти. Нажать кнопку История входа и найти среди строк таблицы информацию о входе. |
| **Этапы теста** | 1. Войти в приложение 2. Ввести действующие логин и пароль оператора 3. Нажать кнопку Войти 4. Нажать кнопку История входа 5. Найти в таблице информацию о входе с введенным логином |
| **Тестовые данные** | Логин: perinaAD  Пароль: 250519 |
| **Ожидаемый результат** | Таблица загрузилась и в ней отображается информация о входе под логином из тестовых данных. |
| **Фактический результат** | Таблица загрузилась и в ней отображается информация о входе под логином из тестовых данных. |
| **Статус** | Зачет |
| **Предварительное условие** | Убедиться что существует пользователь с данными логином и паролем существует. |
| **Постусловие** | В таблице отображается информация о входе с данным логином и паролем. |
| **Примечания/комментарии** |  |

Тестовый пример #4:

|  |  |
| --- | --- |
| **Тестовый пример #** | UserTest4 |
| **Приоритет тестирования** | низкий |
| **Заголовок/название теста** | Неверная авторизация в систему 1 раз с вводом CAPCHA. |
| **Краткое изложение теста** | Войти в приложение, ввести в поле логина логин из тестовых данных, ввести пароль в поле пароля из тестовых данных, нажать кнопку войти. Ввести данные с CAPCHA и нажать кнопку Подтвердить. |
| **Этапы теста** | 1. Войти в приложение 2. Ввести неверные логин и пароль 3. Нажать кнопку Войти 4. Ввести данные с картинки CAPCHA 5. Нажать кнопку Подтвердить |
| **Тестовые данные** | Логин: авав  Пароль: вав |
| **Ожидаемый результат** | После верного ввода капчи доступна кнопка Войти |
| **Фактический результат** | Доступна кнопка Войти |
| **Статус** | Зачет |
| **Предварительное условие** | Убедиться что вводимые данные не принадлежат какому-либо пользователю. |
| **Постусловие** | Доступна форма для авторизации. |
| **Примечания/комментарии** |  |

Тестовый пример #5:

|  |  |
| --- | --- |
| **Тестовый пример #** | UserTest5 |
| **Приоритет тестирования** | низкий |
| **Заголовок/название теста** | Проверить работу фильтра в заявках |
| **Краткое изложение теста** | Войти в приложение, ввести в поле логина логин из тестовых данных, ввести пароль в поле пароля из тестовых данных, нажать кнопку войти. Нажать на кнопку Заявок. Ввести данные из тестовых данных в поле для фильтрации и нажать кнопку Найти. |
| **Этапы теста** | 1. Войти в приложение 2. Ввести неверные логин и пароль 3. Нажать кнопку Войти 4. Нажать кнопку Заявки 5. Вставить текст в поле фильтра 6. Нажать кнопку Найти |
| **Тестовые данные** | Логин: perinaAD  Пароль: 250519  Данные для фильтрации: Готов к выдаче |
| **Ожидаемый результат** | Отображение в таблице 3 заявок с содержанием Статуса Готов к выдаче |
| **Фактический результат** | Отображение в таблице 3 заявок с содержанием Статуса Готов к выдаче |
| **Статус** | Зачет |
| **Предварительное условие** | Убедиться что существует пользователь с данными логином и паролем. Убедиться, что в таблице есть поле с содержанием «Готов к выдаче» |
| **Постусловие** | Доступен функционал Заявок оператора и отфильтрованные данные в таблице Заявки по данным из фильтра. |
| **Примечания/комментарии** |  |

**Модульные тесты**

В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Обратите внимание, что имена тестов должны отражать их суть, т.е. вместо TestMethod1() тест следует назвать, например, GetQuantityForProduct\_NonExistentProductType() для тестирования случая передачи несуществующего типа продукции.

Реализуйте unit-тесты (хотя бы 1 тест на 1 функцию) **минимум 5 штук** для библиотеки. Важно, чтобы тестовые данные предусматривали различные ситуации.

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using Class;

using static System.Net.Mime.MediaTypeNames;

using System.Threading;

using System.Windows.Forms;

using System.Drawing;

namespace UnitTestProjectClass

{

[TestClass]

public class UnitTest1

{

public static string text = "HHHH";

System.Windows.Forms.Timer timer1;

// public static int numCAPTCHA = 1;

Form form;

Button button4;

[TestMethod]

public void Return\_1Text()

{

Class1.numCAPTCHA = 0;

text = Class1.CAPTCHA(timer1, button4, form);

Assert.AreEqual(

text,

"Символы введены неверно."

);

}

[TestMethod]

public void Return\_Text\_numCAPTCHA\_2()

{

Class1.numCAPTCHA = 1;

text = Class1.CAPTCHA(timer1, button4, form);

Assert.AreEqual(

text,

"Слишком много неудачных попыток. Блокировка системы на 3 минуты."

);

}

[TestMethod]

public void Return\_Text\_numCAPTCHA\_3()

{

Class1.numCAPTCHA = 3;

text = Class1.CAPTCHA(timer1, button4, form);

Assert.AreEqual(

text,

"Слишком много неудачных попыток. Полная блокировка системы. Перезапустить систему?"

);

}

[TestMethod]

public void Rol42()

{

text = Class1.Load(4);

Assert.IsNotNull(

text, "Мастер или клиент"

);

}

[TestMethod]

public void Rol3()

{

text = Class1.Load(8);

Assert.IsNotNull(

text, "оператор"

);

}

}

}

