МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Лабораторная работа №2**

по дисциплине: «Объектно-ориентированные технологии

программирования и стандарты проектирования»

на тему: «Разработка диаграмм прецедентов и классов»

Вариант №1 «Журнал регистрации корреспонденции»

|  |  |
| --- | --- |
| Выполнил | Студент группы 21-ИТ-1  Чиникайло А.П. |
|  |  |
| Проверил | Преподаватель  Хирьянов И.Д. |

Полоцк 2023

**Цель работы:** Разработать тестовое приложение в Windows Form на основе диаграммы прецедентов (use case diagram) и классов из лабораторной работы №1 согласно ранее выбранному варианту.

**Ход работы**

**Задание:**

1. Создать проект.
2. Создать классы согласно диаграмме классов. Включить атрибуты и спецификации методов в точности с диаграммой классов.
3. Добавить в классы атрибут name (строковый) и инициализировать его в конструкторе строкой «Имя класса №объекта».
4. Реализовать методы классов в упрощенном варианте: при вызове метода выдавать сообщение о том, что выполняет метод и с каким объектом.
5. Проанализировать требования варианта задания и диаграмму прецедентов из Лабораторной работы No1. В случае возникновения различий доработать приложение.
6. Скорректировать диаграммы прецедентов и классов по разработанному приложению.
7. Подготовить отчет к лабораторной работе.

**Результат работы**

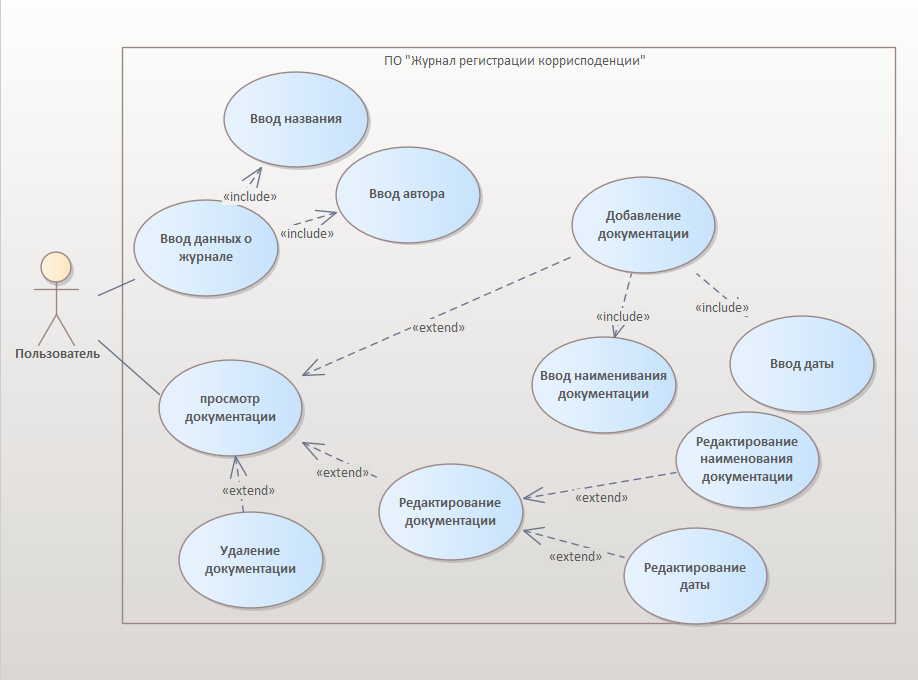
****

Рисунок 1. Исходная диаграмма прецендентов

****

Рисунок 2. Исходная диаграмма классов

Листинг 1.1 - реализация класса From1:

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace WF\_OOTPiSP\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

magazines = new List<Magazine>();

}

List<Magazine> magazines;

public int selectListBox;

private void btn\_skipToAddition\_Click(object sender, EventArgs e)

{

lab\_author.Visible = true;

lab\_inName.Visible = true;

btn\_AddMagazine.Visible = true;

txb\_inAuthor.Visible = true;

txb\_inName.Visible = true;

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

int number = listBox1.SelectedIndex;

}

public Magazine GetMagazineAt(int index)

{

return magazines[index];

}

private void btn\_AddMagazine\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txb\_inAuthor.Text) || string.IsNullOrEmpty(txb\_inName.Text)) {

MessageBox.Show("введите все данные", "error", MessageBoxButtons.OK);

return;

}

magazines.Add(new Magazine(txb\_inName.Text, txb\_inAuthor.Text));

string str = magazines[magazines.Count - 1].name;

listBox1.Items.Add(magazines[magazines.Count - 1].name);

txb\_inAuthor.Text = txb\_inName.Text = string.Empty;

}

private void btn\_ShowMagazine\_Click(object sender, EventArgs e)

{

if(listBox1.SelectedIndex == -1)

{

return;

}

selectListBox = listBox1.SelectedIndex;

Magazine magazine = GetMagazineAt(selectListBox);

using(Form2 form = new Form2(magazine))

{

form.ShowDialog();

}

}

}

}

Листинг 1.2 - реализация класса From2:

using System;

using System.Windows.Forms;

namespace WF\_OOTPiSP\_2

{

public partial class Form2 : Form

{

public Form2(Magazine magazine)

{

InitializeComponent();

this.magazine = magazine;

lab\_nameMagazine.Text = magazine.name;

}

Magazine magazine;

private void btn\_Add\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(txb\_InDate.Text))

{

MessageBox.Show("введите дату");

return;

}

if (string.IsNullOrEmpty(txb\_InName.Text))

{

MessageBox.Show("введите наименование");

return;

}

if(!DateTime.TryParse(txb\_InDate.Text, out DateTime dt))

{

MessageBox.Show("неверный ввод даты");

return;

}

magazine.Add(new Correspondence("new", Type.resolution));

MessageBox.Show($"вызван метод Add у magazine. name: {magazine.name} и author: {magazine.author}");

}

private void btn\_delete\_Click(object sender, EventArgs e)

{

magazine.Delete(0);

MessageBox.Show($"вызван метод Delete у magazine. name: {magazine.name} и author: {magazine.author}");

}

private void btn\_edit\_Click(object sender, EventArgs e)

{

magazine.Edit(new Correspondence("edit", Type.resolution));

MessageBox.Show($"вызван метод Edit у magazine. name: {magazine.name} и author: {magazine.author}");

}

private void lab\_exit\_Click(object sender, EventArgs e)

{

this.Hide();

}

}

}

Листинг 1.3 - реализация класса Magazine:

using System;

using System.Collections.Generic;

using System.Windows.Forms;

namespace WF\_OOTPiSP\_2

{

public class Magazine

{

public Magazine(string name, string author)

{

correspondence = new List<Correspondence>();

this.name = name;

this.author = author;

nameClass = $"{GetType().Name} {numberClass++}";

}

List<Correspondence> correspondence;

public string name { get; }

public string author { get; }

public string nameClass;

private static int numberClass = 1;

public void Add(Correspondence correspondence)

{

this.correspondence.Add(correspondence);

}

public void Delete(int index)

{

if(index > 0 && index < correspondence.Count)

correspondence.RemoveAt(index);

}

public void Edit(Correspondence correspondence)

{

//edit

}

public void InputMagazine(string name, string author)

{

if (string.IsNullOrEmpty(name))

{

name = name;

author = author;

}

}

public void Show()

{

}

}

}

Листинг 1.4 - реализация класса Correspondence:

using System;

using System.Collections.Generic;

namespace WF\_OOTPiSP\_2

{

public class Correspondence

{

public Correspondence(string text, Type type)

{

this.text = text;

this.type = type;

nameClass = $"{GetType().Name} {numberClass++}";

}

public string text { get; }

public Type type { get; }

public string nameClass;

private static int numberClass = 1;

}

}

Листинг 1.5 - реализация enum Type:

public enum Type

{

letter,

order,

resolution,

complaints

}

Листинг 1.6 - реализация класса Program:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WF\_OOTPiSP\_2

{

internal static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

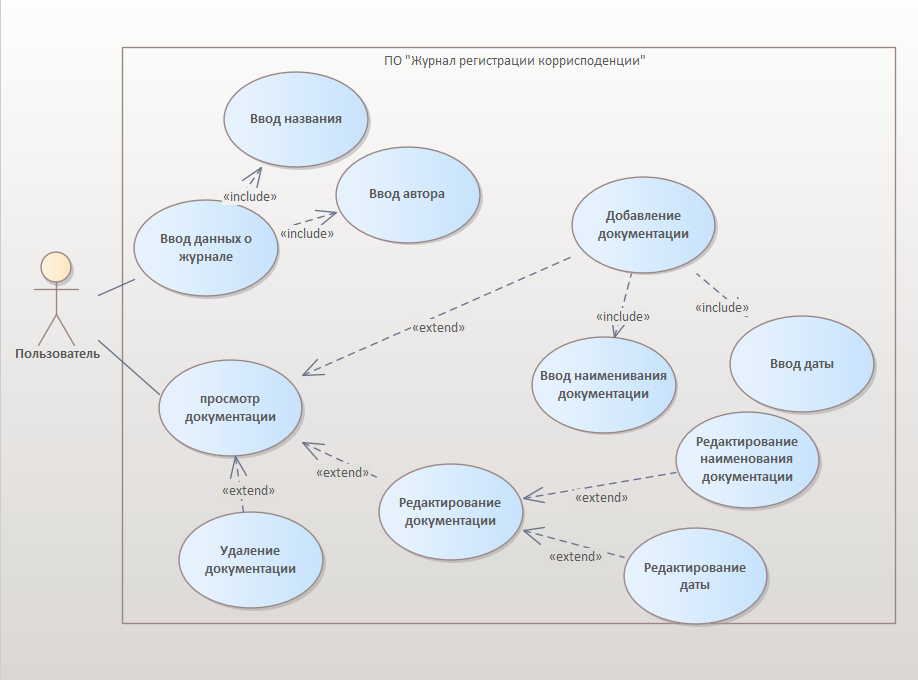
****

Рисунок 3. Скорректированная диаграмма прецендентов

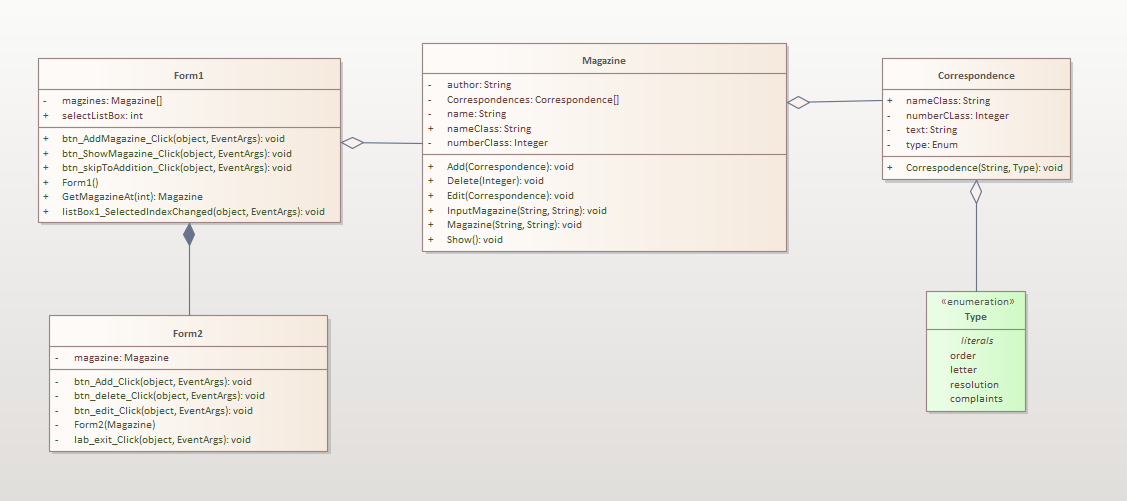


Рисунок 4. Скорректированная диаграмма классов

**Вывод:** В ходе данный лабораторной работе было разработано тестовое приложение в Windows Form на основе диаграммы прецедентов (use case diagram) и классов.