

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА
ПОЯСНЮВАЛЬНА ЗАПИСКА
з дисципліни “ Об’єктно -орієнтоване програмування”
на тему: «Довідник начальника в’язниці»

Керівник, доцент

Студент гр. ПЗП-18-2

Мар’їн С. О.

Шемрікович А. Д.

Комісія:

доц.	_____	Кравець Н. С.
доц.	_____	Мар’їн С.О.
ст. викл.	_____	Черепанова Ю. Ю.

Харків 2019

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ**

Кафедра: *Програмної інженерії*

Дисципліна: *Об'єктно-орієнтоване програмування*

Спеціальність: *121 Інженерія програмного забезпечення*

Освітня програма: *Програмна інженерія*

Курс 1. Група ПЗП-18-2. Семестр 2.

ЗАВДАННЯ
на курсовий проект студента
Шемрікович Ангеліни Дмитрівни
(Прізвище, Ім'я, По батькові)

1 Тема проекту: довідник начальника в'язниці

2 Термін здачі студентом закінченого проекту: **"15" - червня - 2019 р.**

3 Вихідні дані до проекту:

База ув'язнених: перегляд анкетних даних ув'язнених, статті терміну, дати взяття під варту, місця в тюремній ієрархії, камери, відомості про родичів, особливостей характеру. Формування статистичних зведень про склад, вибір за довільною ознакою.

4 Зміст розрахунково-пояснювальної записки:

Вступ, програмна специфікація, проектна специфікація, інструкція користувача, висновки, список літератури, додаток А.

5 Перелік графічного матеріалу:

Схема об'єктної моделі, алгоритми, приклади екранних форм, архітектура проекту

КАЛЕНДАРНИЙ ПЛАН

<i>№</i>	<i>Назва етапу</i>	<i>Термін виконання</i>
1	Видача теми, узгодження і затвердження теми	19-02-2019 р.
2	Формулювання вимог до програми	20-02-2018 – 26-02-2019 р.
4	Розробка підсистеми зберігання даних ув'язнених.	30-02-2019 – 4-03-2019 р.
5	Розробка функцій звільнення ув'язнених та виведення статистичних даних.	5-03-2019 – 10-03-2019 р.
6	Розробка функцій зберігання та завантаження даних	10-03-2019 – 12-03-2019 р.
7	Тестування і доопрацювання розробленої програмної системи.	15-03-2019 – 11-05-2019 р.
8	Оформлення пояснювальної записки, додатків, графічного матеріалу	30-05-2019 – 10-06-2019 р.
9	Захист	5-06-2019 – 15-06-2019 р.

Студент гр. ПЗП- 18-2 _____

Шемрікович А. Д .

Керівник _____

доц. кафедри ПІ Мар'їн С. О.

« 19 » лютого _____ 2019 р.

РЕФЕРАТ

Пояснювальна записка: 37 с., 21 рис., 1 додаток, 5 джерел.

МОВА ПРОГРАМУВАННЯ C#, КЛАС, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПРОГРАМА, ПЛАТФОРМА .NET, ДОВІДНИК НАЧАЛЬНИКА В'ЯЗНИЦІ, ДЕСКТОПНИЙ ДОДАТОК, WINFORMS

Метою роботи є розробка програми «Довідник начальника в'язниці» на засадах об'єктно-орієнтованого програмування.

Методи розробки базуються на використанні середовища розробки Microsoft Visual Studio 2017, Windows Forms, платформи .NET Framework 4.5, мови програмування C#.

У результаті розроблено програму «Довідник начальника в'язниці», яка здатна полегшити робочий процес начальника в'язниці та уповноважених осіб. Користувачеві доступний великий спектр функцій упорядкування.

ЗМІСТ

ВСТУП.....	6
1 СПЕЦИФІКАЦІЯ ПРОГРАМИ.....	7
1.1 Опис функцій програми	7
1.2. Інтерфейс користувача	7
2 ПРОЕКТНА СПЕЦИФІКАЦІЯ	11
2.1 Об'єктна модель програми	11
2.2. Реалізація функцій програми.....	13
2.3. Формат даних	14
3 КЕРІВНИЦТВО КОРИСТУВАЧА.....	15
3.1. Установка програми	15
3.2 Робота с програмою	15
ВИСНОВКИ	19
ПЕРЕЛІК ПОСИЛАНЬ	20
ДОДАТОК А	21

ВСТУП

У курсовій роботі розглядається розробка програми, в якій буде представлений список в'язнів із необхідними даними, як-от номер камери, строк ув'язнення, стаття кримінального кодексу тощо. Додаток заснований на принципах проектно-орієнтованого програмування.

Темою курсової роботи є клієнтська програма, що дозволяє вести базу, а також взаємодіяти з нею шляхом додавання, видалення, редагування даних у базі, а також здійснювати пошук серед елементів бази та впорядковувати їх за обраними ознаками. У програмі віддано перевагу текстовим документам.

Працюючи з великою кількістю анкетних даних складно не загубити важливу інформацію. Запропонована програма допомагає користувачу, що обіймає посаду начальника в'язниці в будь-який час знайти потрібну інформацію про ув'язнених. Розроблена програма дозволяє це швидко зробити, тому вона стане відмінним помічником у роботі. У програмі наявні вся необхідна інформація про в'язнів, повне ім'я, стаття Кримінального Кодексу України, за якою засуджено особу, номер камери, строк ув'язнення, дата взяття під варту, дата народження, місце прописки, особливості характеру, відомості про родичів, місце, що особа обіймає у тюремній ієрархії. Програма має повноцінний функціонал роботи з анкетними даними про людей: можна видаляти колекції, створювати їх, додавати нових людей і дані про них, видаляти їх і редагувати.

Програма, розроблена в курсовій роботі, полегшує пошук інформації про ув'язнених дає можливість повноцінно працювати з вже внесеними даними.

Використання об'єктно-орієнтованого підходу при створенні програми спрощує реалізацію функціоналу, дозволяє побудувати чітку і лаконічну архітектуру, що в подальшому дає змогу легко вдосконалювати додаток шляхом мінімальних утручань в вихідний код [1]

1 СПЕЦИФІКАЦІЯ ПРОГРАМИ

1.1 Опис функцій програми

Програма «Довідник начальника в'язниці» надає користувачеві можливість структуровано зберігати й переглядати інформацію про ув'язнених та додавати нових, редагувати свої колекції.

Серед основних функцій програмного засобу слід виділити наступні:

- перегляд списку людей;
- додавання нової людини в базу каталогу;
- редагування інформації про людину;
- видалення інформації про людину;
- пошук людини за різними критеріями;
- упорядкування списку внесених до бази людей за різними ознаками;

1.2. Інтерфейс користувача

При запуску програми буде відображена головне вікно зі списком, який буде заповнений інформацією про всіх персон, які занесені в програму, та меню зверху (див. рис. 1.1).

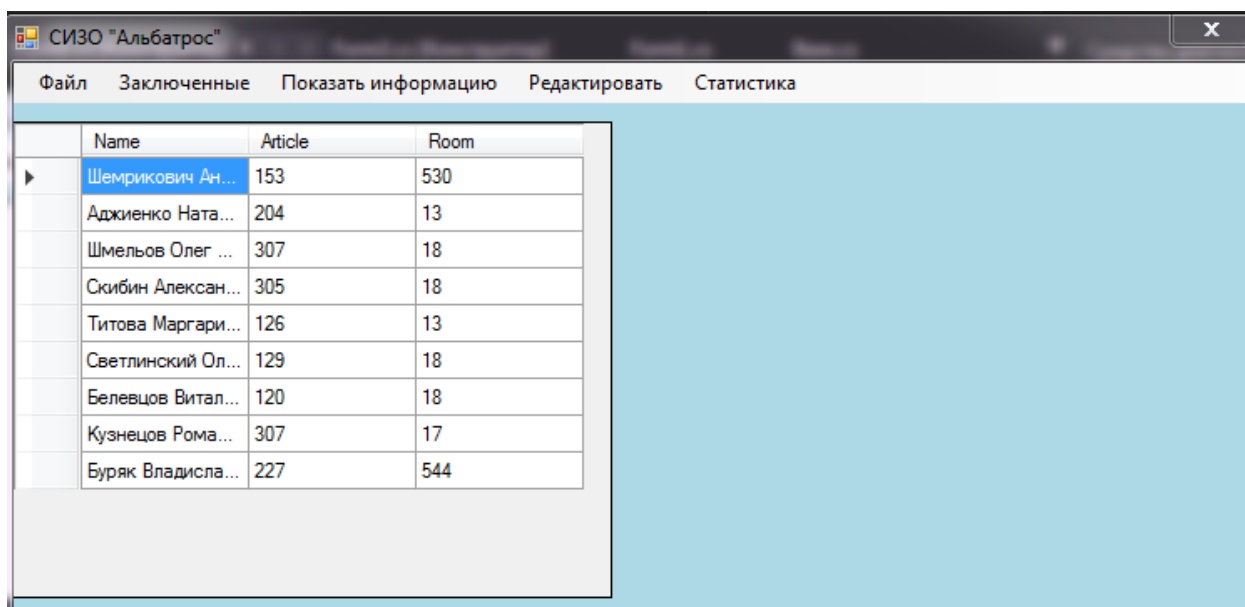


Рисунок 1.1. Головне вікно

Програма надає змогу подивитися детальну інформацію про ув'язненого, виділивши потрібний рядок та натиснувши у меню кнопку «Показать информацию» (див. рис. 1.2.).

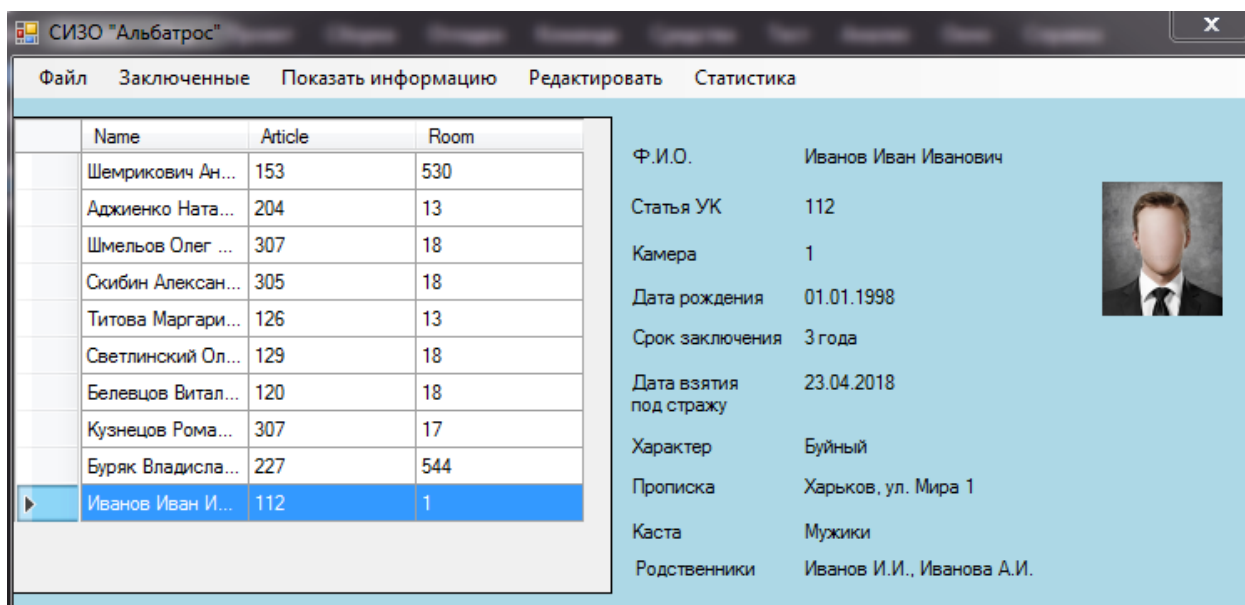


Рисунок 1.2. Детальна інформація про особу

Користувач може подивитися статистичні дані в'язниці, такі як кількість ув'язнених за довільною статтею або ж кількість ув'язнених у довільній кімнаті, для цього йому потрібно натиснути на кнопку «Статистика», відкриється дочірнє вікно з порожніми полями для введення (див. рис. 1.3).

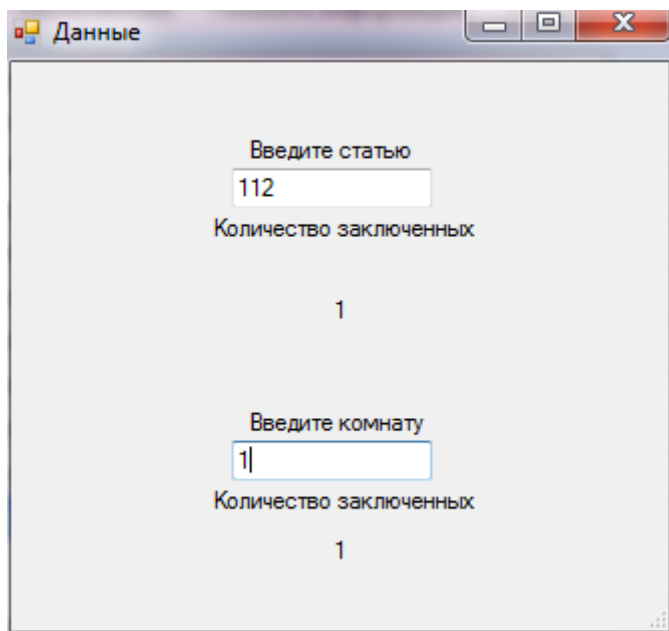


Рисунок 1.3. Демонстрація виведення статистичних даних

Для додавання нового запису треба натиснути кнопку «Заключенные» та обрати пункт «Добавить». При цьому з'являється дочірнє вікно з порожніми полями додавання. Після заповнення потрібно натиснути на кнопку «Загрузить фотографию» та обрати потрібний файл формату .jpg, .png, або .gif на своєму комп'ютері. Для того, щоб додати особу до списку, треба натиснути кнопку «ОК», а для відміни – «Закрыть» (див. рис. 1.4).

Регистрация

Ф.И.О

Статья УК Украины

Камера

Дата рождения

Срок заключения

Срок взятия под стражу

Загрузить фото

Характер

Место прописки

Тюремная каста

Родственники

OK

Закреть

Рисунок 1.4. Додавання запису

У списку можна редагувати будь-який запис, для цього потрібно натиснути на кнопку «Редактирование» обрати пункт «Начать редактирование», таблиця з даними розшириться та з'являться додаткові колонки інформації. Для редагування треба два рази натиснути на клітинку та вписати в нього оновлені дані. Для завершення редагування потрібно натиснути на кнопку «Редактирование» та обрати пункт «Закончить редактирование» (див. рис. 1.5.).

	Name	Age	Character
✎	Иван Петров	01.02.1989	а

Рисунок 1.5. Процес редагування

Для того, щоб видалити запис з таблиці, треба виділити потрібний рядок, обрати кнопку «Заключенные» та пункт «Удалить». Після цього особа буде видалена з бази.

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ

2.1 Об'єктна модель програми

Розроблена програма являє собою набір класів, які взаємодіють між собою (див. рис. 2.1).

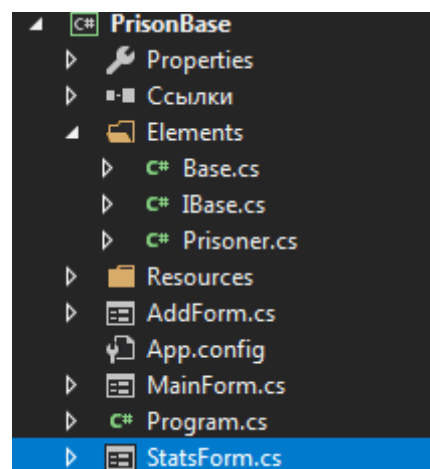


Рисунок 2.1. Архітектура проекту

У даній програмі реалізовані класи Prisoner, PrisonBase, IBase, об'єктна модель яких зображена на рисунку 2.2.

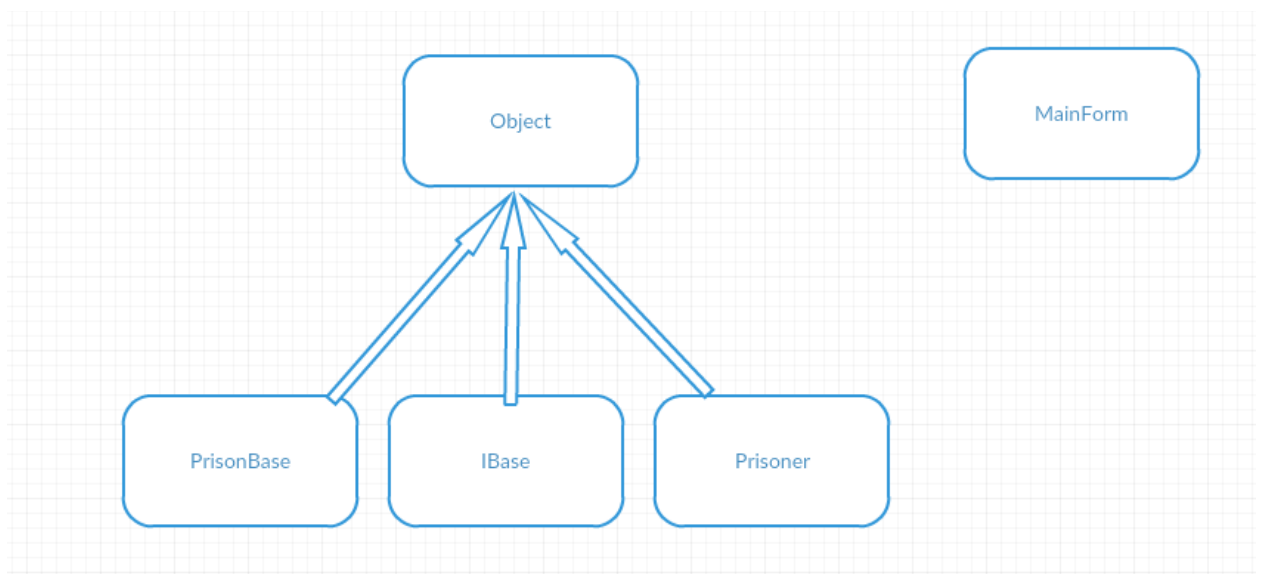


Рисунок 2.2. Об'єктна модель класів

На рисунку 2.3 показано, як об'єкти класів взаємодіють між собою (Controls являє собою сукупність всіх візуальних елементів користувацького інтерфейсу).

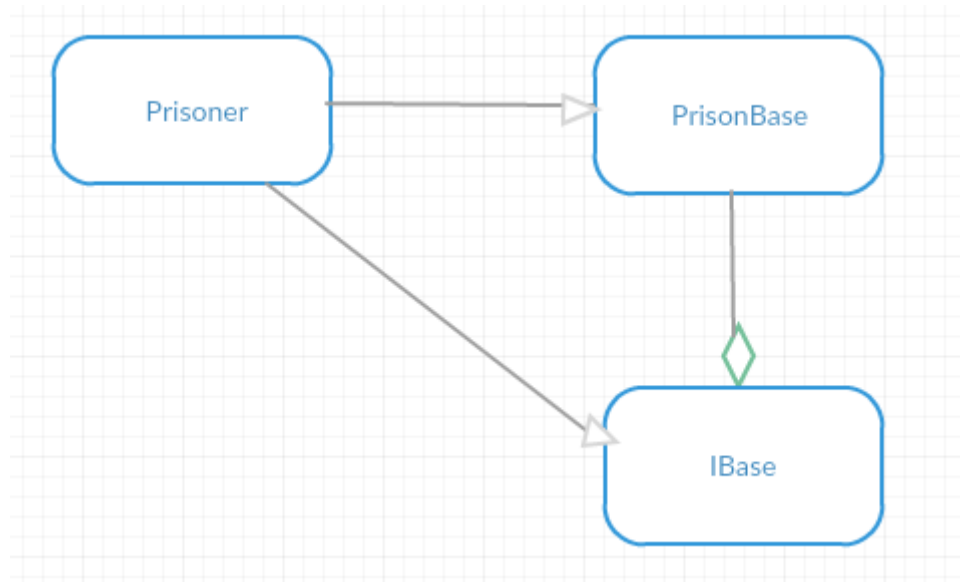


Рисунок 2.3 Ілюстрація взаємодії класів в програмі

Клас «PrisonBase» (див. рис. 2.4.) є основним класом для завантаження інформації в файли, заповнення таблиці інформацією.

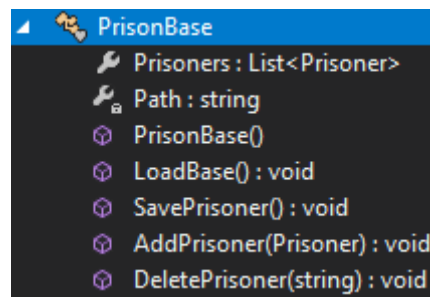


Рисунок 2.4. Клас PrisonBase та його методи

Через метод «LoadBase» програма здійснює завантаження даних з файлів програми. Метод «AddPrisoner» додає нову людину до колекції. Метод «DeletePrisoner» видаляє людину з колекції. Метод «SavePrisoner» записує до текстового файлу дані про ув'язнених.

Клас «Prisoner» представлено на рисунку 2.5.

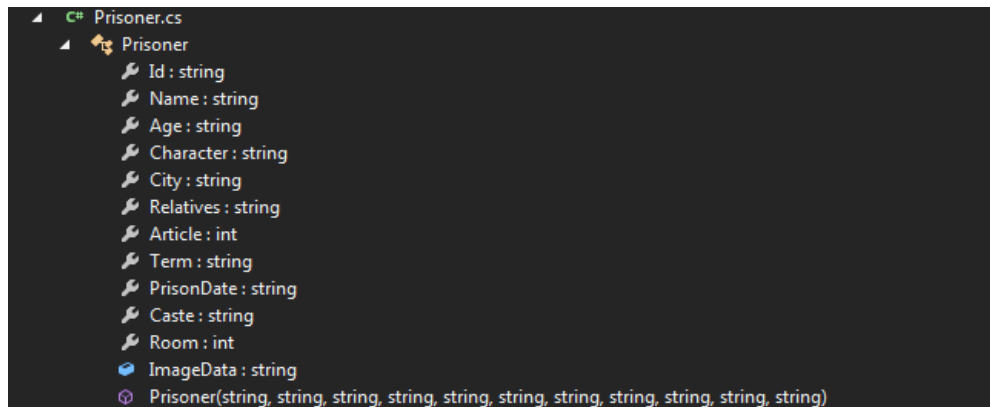


Рисунок 2.5. Клас Prisoner

Клас містить властивості та конструктор, що отримує та привласнює дані цим властивостям.

Інтерфейс «IBase» представлено на рисунку 2.6.

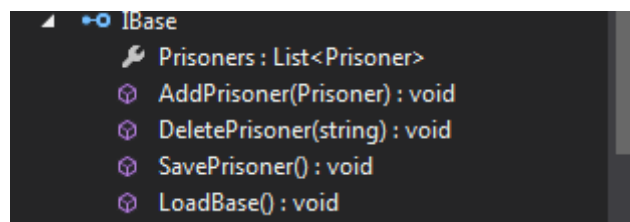


Рисунок 2.6. Інтерфейс IBase

Інтерфейс містить всі методи класу «PrisonerBase» та колекцію типу Prisoner.

2.2. Реалізація функцій програми

Клас «Prisoner» - основний клас програми. Клас містить ряд властивостей, необхідних для роботи з елементом даного класу.

Клас «PrisonBase» містить метод «LoadPrisoner», який реалізує завантаження інформації про людину шляхом зчитування з файлу серіалізованих даних у форматі txt та передачі їх у конструктор класу

«Prisoner». Метод «AddPrisoner» існує для додавання нової людини в колекцію. Метод «DeletePrisoner» видаляє людину з колекції за її ключем. Метод «SavePrisoner» реалізує запис інформації про об'єкти у файл у форматі txt.

Інтерфейс IBase містить у собі методи «LoadPrisoner», «SavePrisoner», «AddPrisoner», «DeletePrisoner» та колекцію типу Prisoner.

2.3. Формат даних

Структура зберігання даних в програмі здійснюється за допомогою серіалізації. Дані зберігаються в форматі txt (див. рис. 2.7).

```
1
Шемрикович Ангелина дмитриевна
153
530
09.06.2001
Буйный
Харьков, ул.целиноградская 36
Шемрикович Д.Н, Шемрикович И.И.
4 года
14.07.2018
Шныри
\PrisonBase\Prison\photo_2018-11-18_21-46-05.jpg
```

Рисунок 2.7. Демонстрація зберігання інформації про людину в файлі

Програма надає змогу не тільки зберігати дані в окремому файлі, а і легко змінювати їх у файлі вручну.

3 КЕРІВНИЦТВО КОРИСТУВАЧА

3.1. Установка програми

Установка програми відбувається таким чином:

- запускаємо «Мій комп'ютер»;
- заходимо носій, з програмним продуктом;
- копіюємо папку «Записна книжка» на комп'ютер;
- запускаємо «.exe» файл для початку роботи.

3.2 Робота с програмою

Запускаємо програму. Головне вікно представлено на рисунку 3.1.

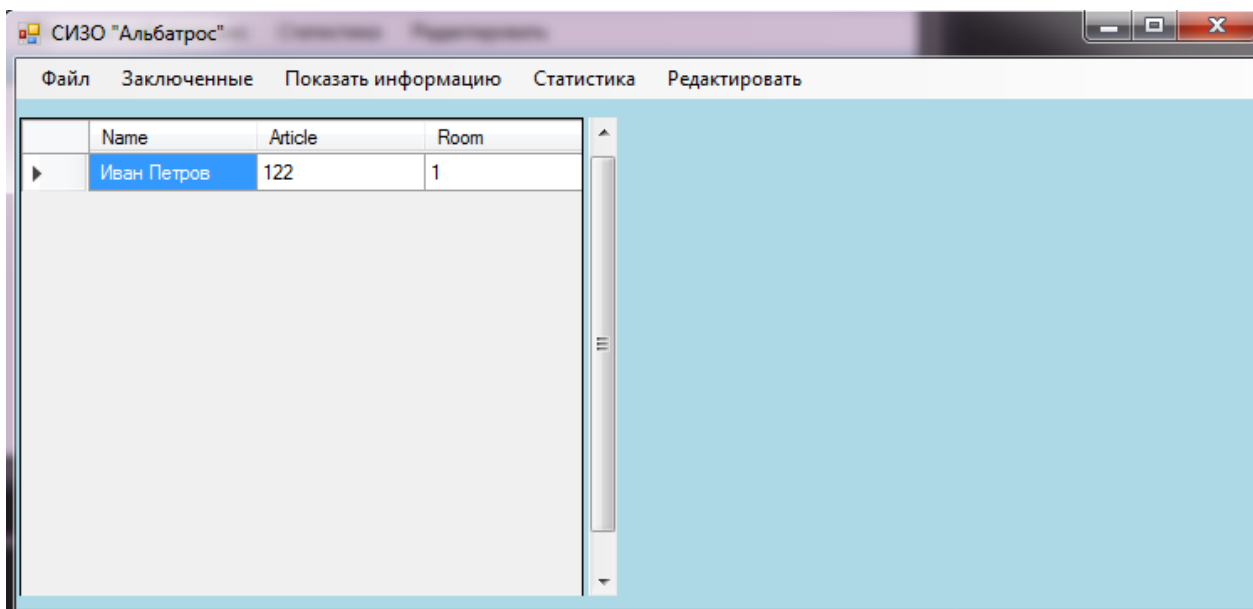


Рисунок 3.1. Головне вікно

Список всіх людей завантажується в таблицю відразу ж після запуску програми.

Щоб показати детальну інформацію про особу, треба виділити рядок за допомогою стрілки, що знаходиться зліва від таблиці та натиснути кнопку

«Показать информацию» (див. рис. 3.2)

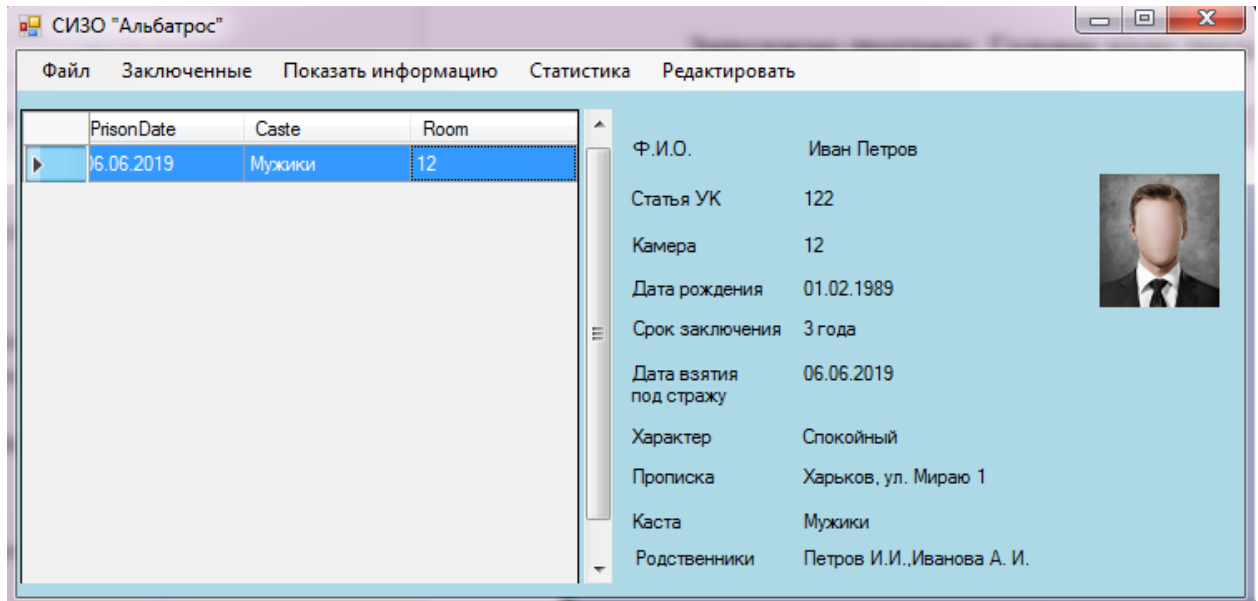


Рисунок 3.2. Демонстрація функції виведення детальної інформації про ув'язненого

Додавання запису:

- натиснути на кнопку «Заключенные» обрати пункт «Добавить»;
- відкриється вікно створення запису, де користувач повинен ввести ім'я, статтю Кримінального кодексу України, за якою особа відбуває покарання, номер камери, дату народження, строк ув'язнення, дату взяття під варту, характер, місце прописки, місце у тюремній ієрархії, відомості про родичів (див. рис. 3.3);
- підтвердити, натиснувши кнопку «ОК».

Рисунок 3.3. Додавання нового запису

Редагування запису:

- виділити потрібний рядок стрілочкою, що знаходиться зліва від таблиці, і натиснути кнопку «Редактирование», обрати пункт «Начать редактирование»;
- Коректувати дані у таблиці;
- Натиснути кнопку «Редактирование», обрати пункт «Закончить редактирование».

Видалення запису:

- виділити потрібний рядок стрілочкою, що знаходиться зліва від таблиці;
- натиснути кнопку «Заключенные», обрати пункт «Удалить».

Виведення статистичних даних:

- Натиснути на кнопку «Статистика»;

Відкриється вікно (див. рис. 3.4), де користувач повинен ввести довільний номер статті кримінального кодексу, або довільний номер кімнати, програма виведе кількість затриманих за вказаною статтею або номером кімнати.

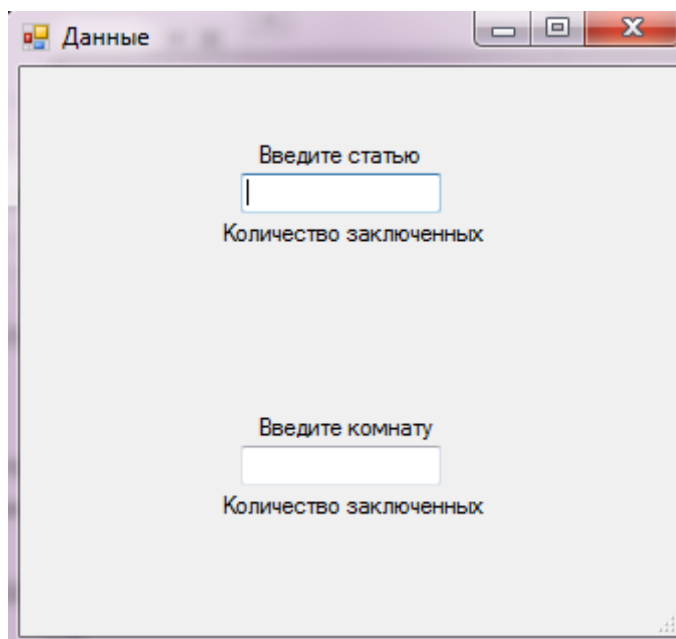


Рисунок 3.4. Виведення статистичних даних

Для завершення роботи програми потрібно натиснути на хрестик.

ВИСНОВКИ

У ході виконання курсової роботи була розроблена програма «Довідник начальника в'язниці», яка дозволяє вести упорядкований довідник з ув'язненими, зберігати інформацію про них, як-от статті Кримінального кодексу України, за якими засуджені особи, номер камери, строк заримання тощо. У програмі реалізовані різні методи: додавання, редагування, видалення записів, збереження даних в окремому файлі, виведення статистичних даних

У курсовій роботі реалізовано десктопний додаток, який може стати в нагоді людині, що обіймає посаду начальника в'язниці або ж замісника начальника в'язниці.

Результатом курсової роботи стала програма «Довідник начальника в'язниці». Вона була розроблена на мові C# за допомогою системи Windows Forms. У ході роботи були опановані аспекти створення додатків у цілому і створення додатків у середовищі .NET [2]. Також були вивчені окремі фази створення програмного забезпечення й компонування їх в єдине ціле [5]. У ході реалізації курсового проекту були досліджені та використані стандарти оформлення програмного забезпечення [4].

ПЕРЕЛІК ПОСИЛАНЬ

1. Бондарєв В. М. Об'єктно-орієнтоване програмування на С#. Навчальний посібник [Текст] / В. М. Бондарєв. – Харків: СМІТ, 2009р. – 221 с.
2. Макконнелл С. Досконалий код [Текст] / С. Макконнелл – СПб.: Питер, 2007. – 896 с.
3. Троелсен Е. С# і платформа .NET. Бібліотека програміста [Текст] / Е. Троелсен – СПб.: Питер, 2005. – 796с.
4. Фролов А. В. Візуальне проектування додатків С # / А. В. Фролов., Г. В. Фролов. – М.: КУДИЦ-ОБРАЗ, 2003. – 512 с.
5. Шілдт Г. Повний довідник по С # [Текст] / Г. Шілдт. – М.: Видавничий дім «Вільямс», 2004. – 752 с.

ДОДАТОК А

```

namespace PrisonBase
{
    public class Prisoner
    {
        public string Id { set; get; }
        public string Name { set; get; }
        public string Age { set; get; }
        public string Character { set; get; }
        public string City { set; get; }
        public string Relatives { set; get; }
        public string Article { set; get; }
        public string Term { set; get; }
        public string PrisonDate { set; get; }
        public string Caste { set; get; }
        public string Room { set; get; }
        public string ImageData;

        public Prisoner(string name, string article,
string room, string age,
                        string character, string city,
string relatives,
                        string term, string prisondate,
string caste, string imageData)
        {
            Name = name;
            Article = article;
            Room = room;
            Age = age;
            Character = character;
            City = city;
            Relatives = relatives;
            Term = term;
            PrisonDate = prisondate;
            Caste = caste;
            ImageData = imageData;
        }
    }
}

class PrisonBase: IBase
{

```

```

        public List<Prisoner> Prisoners {protected set;
get; }
        string path { set; get; }
        public PrisonBase(string path )
        {
            this.path = path;
            if(File.Exists(path))
            {
                LoadBase();
            }
            else
            {
                Prisoners = new List<Prisoner>();
            }
        }
        public void LoadBase()
        {
            Prisoners = new List<Prisoner>();
            using (TextReader reader = new
StreamReader(File.OpenRead(path)))
            {
                string s = null;
                while ((s = reader.ReadLine()) != null)
                {
                    var name = reader.ReadLine();
                    var article = reader.ReadLine();
                    var room = reader.ReadLine();
                    var age = reader.ReadLine();
                    var character = reader.ReadLine();
                    var city = reader.ReadLine();
                    var relatives = reader.ReadLine();
                    var term = reader.ReadLine();
                    var prisonDate = reader.ReadLine();
                    var caste = reader.ReadLine();
                    var imageData = reader.ReadLine();
                    Prisoners.Add(new
Prisoner(name,article,room, age, character, city,
relatives,term,prisonDate, caste, imageData));
                }
            }
        }
        public void SavePrisoner()
        {

```

```

        using (TextWriter writer = new
StreamWriter(path))
        {
            foreach (Prisoner prisoner in
Prisoners)
            {
                writer.WriteLine(prisoner.Id);
                writer.WriteLine(prisoner.Name);
                writer.WriteLine(prisoner.Article);
                writer.WriteLine(prisoner.Room);
                writer.WriteLine(prisoner.Age);

writer.WriteLine(prisoner.Character);
                writer.WriteLine(prisoner.City);

writer.WriteLine(prisoner.Relatives);
                writer.WriteLine(prisoner.Term);

writer.WriteLine(prisoner.PrisonDate);
                writer.WriteLine(prisoner.Caste);

writer.WriteLine(prisoner.ImageData);

            }
        }
    }
    public Prisoner AddPrisoner (Prisoner prisoner)
    {
        prisoner.Id = Prisoners.Count() == 0 ? "1"
: (Prisoners.Max(b => Convert.ToInt32(b.Id)) +
1).ToString();
        Prisoners.Add(prisoner);
        return prisoner;
    }
    public void DeletePrisoner (string id)
    {
        int id_p = 0;

        for(int i = 0; i < Prisoners.Count; i++)
        {
            if( Prisoners[i].Id == id )
            {
                id_p = i;

```

```

        break;
    }
}

Prisoners.RemoveAt(id_p);
}
}

public interface IBase
{
    List<Prisoner> Prisoners { get; }
    Prisoner AddPrisoner(Prisoner prisoner);
    void DeletePrisoner(string id);
    void SavePrisoner();
    void LoadBase();
}

public partial class MainForm : Form
{
    public static IBase PrisonBase;

    public MainForm()
    {
        PrisonBase = new
Elements.PrisonBase("base.txt");
        InitializeComponent();
    }
    private void Form1_Load(object sender,
EventArgs e)
    {
        if (PrisonBase.Prisoners.Count != 0)
        {
            Bind();
            InvisibleRows();
        }
    }
    private void Bind()
    {
        dataGridView1.DataSource = new int[] { };
        var data = PrisonBase.Prisoners;
        dataGridView1.DataSource = data;
    }

    void InvisibleRows()

```



```

        {
            dataGridView1.Columns["Id"].Visible =
false;
            dataGridView1.Columns["Age"].Visible =
false;
            dataGridView1.Columns["Character"].Visible
= false;
            dataGridView1.Columns["City"].Visible =
false;
            dataGridView1.Columns["City"].Visible =
false;
            dataGridView1.Columns["Relatives"].Visible
= false;
            dataGridView1.Columns["Term"].Visible =
false;
            dataGridView1.Columns["PrisonDate"].Visible
= false;
            dataGridView1.Columns["Caste"].Visible =
false;
        }

        private void
dataGridView1_CellValueChanged(object sender,
DataGridViewCellEventArgs e)
        {
            PrisonBase.SavePrisoner();
            if (dataGridView1.SelectedRows != null)
            {
                panell1.Visible = true;
                Show();
            }
        }

        public void Show()
        {
            foreach (DataGridViewRow row in
dataGridView1.SelectedRows)
            {
                int ind = row.Index;
                NameLabel.Text =
PrisonBase.Prisoners[ind].Name;
                ArticleLabel.Text =
Convert.ToString(PrisonBase.Prisoners[ind].Article);
            }
        }
    }
}

```

```

        RoomLabel.Text =
Convert.ToString(PrisonBase.Prisoners[ind].Room);
        AgeLabel.Text =
PrisonBase.Prisoners[ind].Age;
        TermLabel.Text =
PrisonBase.Prisoners[ind].Term;
        PrisonDateLabel.Text =
PrisonBase.Prisoners[ind].PrisonDate;
        CharacterLabel.Text =
PrisonBase.Prisoners[ind].Character;
        CityLabel.Text =
PrisonBase.Prisoners[ind].City;
        CasteLabel.Text =
PrisonBase.Prisoners[ind].Caste;
        RelativesLabel.Text =
PrisonBase.Prisoners[ind].Relatives;
        pictureBox1.Image = new
Bitmap(PrisonBase.Prisoners[ind].ImageData);

    }
}

private void AddToolStripMenuItem_Click(object
sender, EventArgs e)
{
    AddForm form = new AddForm();
    form.ShowDialog();
    Bind();
    InvisibleRows();
}

private void
DeleteToolStripMenuItem_Click(object sender, EventArgs
e)
{
    foreach (DataGridViewRow row in
dataGridView1.SelectedRows)
    {
        int ind = row.Index;
        string ind_ = Convert.ToString(ind);
        PrisonBase.DeletePrisoner(ind_);
        Bind();
        InvisibleRows();
        panell.Visible = false;
    }
}

```

```

        }
    }

    private void
ShowInfoToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        if (dataGridView1.SelectedRows != null)
        {
            panell1.Visible = true;
            Show();
        }
    }

    private void
StatsToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        StatsForm form = new StatsForm();
        form.ShowDialog();
    }

    private void SaveToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        if (openFileDialog1.ShowDialog() ==
DialogResult.OK)
        {
            PrisonBase.SavePrisoner();
        }
    }

    private void
StartEditToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        dataGridView1.Columns["Age"].Visible =
true;
        dataGridView1.Columns["Character"].Visible
= true;
        dataGridView1.Columns["City"].Visible =
true;
        dataGridView1.Columns["City"].Visible =
true;
    }

```

```

        dataGridView1.Columns["Relatives"].Visible
= true;
        dataGridView1.Columns["Term"].Visible =
true;
        dataGridView1.Columns["PrisonDate"].Visible
= true;
        dataGridView1.Columns["Caste"].Visible =
true;
    }

    private void
StopEditToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        InvisibleRows();
    }
}

public partial class StatsForm : Form
{
    public StatsForm()
    {
        InitializeComponent();
    }

    private void Form3_Load(object sender,
EventArgs e)
    {
        PrisonerLabel.Visible = false;
        RoomLabel.Visible = false;
    }

    private void textBox1_TextChanged(object
sender, EventArgs e)
    {
        if(textBox1.Text != "")
        {
            string point = textBox1.Text;
            int count = 0;
            for (int i = 0; i <
MainForm.PrisonBase.Prisoners.Count; i++)
            {
                if
(MainForm.PrisonBase.Prisoners[i].Article == point)

```

```

        {
            count++;
        }
    }
    PrisonerLabel.Visible = true;
    PrisonerLabel.Text =
Convert.ToString(count);
    }
}

private void textBox2_TextChanged(object
sender, EventArgs e)
{
    if (textBox2.Text != "")
    {
        string point = textBox2.Text;
        int count = 0;
        for (int i = 0; i <
MainForm.PrisonBase.Prisoners.Count; i++)
        {
            if
(MainForm.PrisonBase.Prisoners[i].Room == point)
            {
                count++;
            }
        }
        RoomLabel.Visible = true;
        RoomLabel.Text =
Convert.ToString(count);
    }
}

public partial class AddForm : Form
{
    public string im;
    public Prisoner Prisoner { private set; get; }

    public AddForm()
    {
        InitializeComponent()
    }
}

```

```

        private void OkButton_Click(object sender,
EventArgs e)
        {
            string name = NameBox.Text;
            string article = "1";
            string room = RoomBox.Text;
            string age = AgeBox.Text;
            string character = CharacterBox.Text;
            string city = CityBox.Text;
            string relatives = RelativesBox.Text;
            string term = TermBox.Text;
            string prisondate = PrisonDateBox.Text;
            string caste = CasteBox.Text;
            string imageData = im;
            if (string.IsNullOrEmpty(NameBox.Text)
|| string.IsNullOrEmpty(ArticleBox.Text) ||
                string.IsNullOrEmpty(RoomBox.Text)
|| string.IsNullOrEmpty(AgeBox.Text) ||

string.IsNullOrEmpty(CharacterBox.Text) ||
string.IsNullOrEmpty(CityBox.Text)||

string.IsNullOrEmpty(RelativesBox.Text) ||
string.IsNullOrEmpty(RelativesBox.Text)||
                string.IsNullOrEmpty(TermBox.Text)
|| string.IsNullOrEmpty(PrisonDateBox.Text)||

string.IsNullOrEmpty(CasteBox.Text) ||
string.IsNullOrEmpty(im))
            {
                DialogResult = DialogResult.None;
                Label.Text = "Заполните все поля";
                return;
            }

            if (IsDigitsOnly(ArticleBox.Text) == false)
            {
                DialogResult = DialogResult.None;
                Label.Text = "Введите существующий
номер статьи";
                return;
            }

```

```

else
{
    if (Convert.ToInt32(ArticleBox.Text) >
0 && Convert.ToInt32(ArticleBox.Text) <= 447)
    {
        article = ArticleBox.Text;
    }
    else
    {
        DialogResult = DialogResult.None;
        Label.Text = "Введите существующую
статью";

        return;
    }
}

if (IsDigitsOnly(RoomBox.Text) == false)
{
    DialogResult = DialogResult.None;
    Label.Text = "Введите существующий
номер камеры";
    return;
}
else
{
    if (Convert.ToInt32(RoomBox.Text) > 0
&& Convert.ToInt32(RoomBox.Text) <= 530)
    {
        article = ArticleBox.Text;
    }
    else
    {
        DialogResult = DialogResult.None;
        Label.Text = "Введите существующий
номер камеры";

        return;
    }
}

```

```

        Prisoner = new Prisoner(name, article, room, age,
                                character, city, relatives,
                                term, prisondate, caste,
imageData);
        MainForm.PrisonBase.AddPrisoner(Prisoner);
        Close();

    }

    bool IsDigitsOnly(string str)
    {
        foreach (char c in str)
        {
            if (c < '0' || c > '9')
                return false;
        }

        return true;
    }

    private void UploadButton_Click(object sender,
EventArgs e)
    {
        OpenFileDialog ofd = new OpenFileDialog();
        ofd.Filter = "Image
Files (*.BMP;*.JPG;*.PNG;) | *.BMP;*.JPG;*.PNG; | All files
(*.*) | *.*";
        if (ofd.ShowDialog() == DialogResult.OK)
        {
            try
            {
                pictureBox1.Image = new
Bitmap(ofd.FileName);
                im = ofd.FileName;
            }
            catch
            {
                MessageBox.Show("Не удалось открыть
данный файл", "Ошибка", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }
    }
}

```



```
        private void CanselButton_Click(object sender,  
EventArgs e)  
        {  
            Close();  
        }  
    }
```