Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра Програмної інженерії

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліні “ Об’єктно-орієнтоване програмування”

ДОВІДНИК НАЧАЛЬНИКА В’ЯЗНИЦІ

Керівник, професор Бондарєв В.М.

Студент гр. ПЗПІ-19-4 Кабаннік Ю.К.

Комісія:

Проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бондарєв В.М.

Доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Побіженко І. О.

Ст. викл. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Черепанова Ю.Ю.

Харків 2020

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ**

Кафедра: ***Програмної інженерії***

Дисципліна: ***Об’єктно-орієнтоване програмування***

Спеціальність: ***121***  ***Інженерія програмного забезпечення***

Освітня програма: ***Програмна інженерія***

Курс 1 .Група *ПЗПІ-19-4* . Семестр *2* .

***ЗАВДАННЯ***

***на курсовий проект студента***

*Кабаннік Юрій Костянтинович*

(Прізвище, Ім'я, По батькові)

1 Тема проекту: Довідник начальника в’язниці

2 Термін здачі студентом закінченого проекту: ***“19” - червня - 2020 р***.

3 Вихідні дані до проекту:

*Специфікація програми, методичні вказівки до виконання курсової роботи*

4 Зміст розрахунково-пояснювальної записки:

*Вступ,* *специфікація програми, проектна специфікація, інструкція користувача, висновки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5 Перелік графічного матеріалу:

*Схема об’єктної моделі, алгоритми, приклади екранних форм, діаграма класів* \_\_\_\_ *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |
| --- | --- | --- |
| *№* | *Назва етапу* | *Термін виконання* |
| 1 | Видача теми, узгодження і затвердження теми | 21-02-2020 р. |
| 2 | Формулювання вимог до програми | 22-02-2020 – 31-03-2020 р. |
| 3 | Розробка підсистеми зберігання та пошуку в’язнів | 01-04-2020 – 12-04-2020 р. |
| 4 | Розробка підсистеми взаємодії зі збереженими даними | 13-04-2020 – 20-04-2020 р. |
| 5 | Розробка графічного інтерфейсу | 21-04-2020 – 30-04-2020 р. |
| 6 | Розробка функцій зберігання та завантаження даних | 01-05-2020 – 04-05-2020 р. |
| 7 | Тестування і доопрацювання розробленої програмної системи | 05-05-2020 – 15-05-2020 р. |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу | 15-05-2020 – 05-06-2020 р. |
| 9 | Захист | 01-06-2020 – 19-06-2020 р. |

Студент Кабаннік Ю.К.

Керівник Бондарев В.М.

« 21 »\_лютого\_\_\_\_\_\_\_\_\_\_ 2020 р.

**РЕФЕРАТ**

Пояснювальна записка до курсової роботи: 49 с., 15 рис., 5 додатків, 5 джерел.

В’ЯЗЕНЬ, В’ЯЗНИЦЯ, КАМЕРА, КЛАС, МОВА ПРОГРАМУВАННЯ C#, ОБ’ЕКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ПЛАТФОРМА .NET, ПРОГРАМА, СТАТТЯ.

Метою роботи є розробка програми «Довідник начальника в’язниці» на засадах об'єктно-орієнтованого програмування для збереження інформації про склад в’язниці.

Методи розробки базуються на використанні середи розробки Microsoft Visual Studio 2019, Windows Forms, платформи .NET Framework 4.5, мови програмування C#.

У результаті отримана програма під назвою «Довідник начальника в’язниці», яка дозволяє звичайному користувачу переглядати інформацію про в’язнів, шукати в’язнів за за певним параметром: ПІБ, ідентифікаційний код, номер статті, номер камери, дивитися статистику про склад в’язниці. Авторизований користувач (начальник) має доступ до усіх можливостей звичайного користувача, а також може додавати, видаляти та змінювати інформацію про в’язнів.

**ЗМІСТ**

ВСТУП………………………………………………………………………7

1 СПЕЦИФІКАЦІЯ ПРОГРАМИ…………………………………….8

1.1 Вимоги персоналу…………………………………………….8

1.2 Вимоги начальника…………………………………………...8

1.3 Функціональні вимоги ………………………………………8

1.4 Нефункціональні вимоги……………………………………..9

1.5 Опис інтерфейсу користувача………………………………..9

2 ПРОЕКТНА СПЕЦИФІКАЦІЯ……………………………………16

2.1 Діаграма класів………………………………………………16

2.2 Об’єктна структура програми………………………………16

2.2.1 Абстрактний клас Person……………………………...17

2.2.2 Клас Prisone…………………………………………….18

2.2.3 Клас Relative…………………………………………...18

2.2.4 Клас State………………………………………………19

2.2.5 Клас Length…………………………………………….20

2.2.6 Клас Room……………………………………………...20

2.2.7 Клас Prison……………………………………………..20

2.2.8 Клас Stats……………………………………………….22

2.2.9 Клас Dao………………………………………………..22

2.2.10 Статичний клас Global……………………………….23

3 ІНСТРУКЦІЯ КОРИСТУВАЧА…………………………………..24

3.1 Запуск та вікно авторизації…………………………………24

3.2 Елементи меню………………………………………………24

3.3 Головне вікно персоналу: пошук в’язнів та перегляд інформації про в’язня……………………………………………………..24

3.4 Головне вікно начальника: додавання, видалення та зміна інформації про в’язнів ……………………………………………………25

3.5 Головне вікно начальника: додавання та видалення родичів та рис характеру…………………………………………………………………………26

3.6 Вікно статистики…………………………………………………...27

3.7 Збереження даних та вихід з програми…………………………...27

ВИСНОВКИ……………………………………………………………………...28

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ…………………………………………….29

ДОДАТОК А Методи класу Prison……………………………………………..30

ДОДАТОК Б Методи класу Dao………………………………………………..34

ДОДАТОК В Посилання на GitHub…………………………………………….36

**ВСТУП**

В’язниця – це дуже важлива державна структура, коректна робота якої є обов’язковою. Збереження інформації про в’язнів у паперовому вигляді – є не зручним: займає багато простору, не надійне, складне для пошуку та обробки. Ці проблеми повністю вирішує програма “Довідник начальника в’язниці”.

Розроблена програма дозволить у зручному форматі переглядати інформацію про в’язнів, робити пошук за параметрами: ПІП, ідентифікаційний код, номер статті, номер камери та дивитися статистику про склад в’язниці. Повний доступ до функцій програми отримує авторизований користувач - начальник в’язниці. Він може робити все, вище зазначене, а також додавати, видаляти та змінювати інформацію про в’язнів. Доступ до облікового запису начальника здійснюється через пароль.

Окрім розробки вище описаної програмної системи метою даної курсової роботи є освоїти мову програмування С#, навчитися об’єктно-орієнтованому програмуванню, розвити навички роботи в Windows Forms.

Задачі роботи:

* Розробка системи зчитування та запису даних;
* Створення логіки роботи системи;
* Робота з користувачем;
* Розробка логіки додавання, пошуку та видалення інформації;
* Тестування готового продукту;
* Створення стійкості програми.

1. **СПЕЦИФІКАЦІЯ ПРОГРАМИ**

1.1 Вимоги персоналу

Серед вимог персоналу слід зазначити наступні:

* Перегляд повної інформації про в’язня;
* Пошук в’язня за певним параметром: ПІП, ідентифікаційний код, номер статті, номер камери;
* Перегляд статистичних даних про склад в’язниці: підрахунок кількості в’язнів за віком та номером статті, пошук моди та середнього віку.

1.2 Вимоги начальника

Серед вимог начальника слід зазначити наступні:

* Перегляд повної інформації про в’язня;
* Пошук в’язня за певним параметром: ПІП, ідентифікаційний код, номер статті, номер камери;
* Перегляд статистичних даних про склад в’язниці: підрахунок кількості в’язнів за віком та номером статті, пошук моди та середнього віку;
* Додавання нових в’язнів;
* Видалення в’язнів;
* Зміна інформації про в’язнів.

1.3 Функціональні вимоги

Серед функціональних вимог слід зазначити наступні:

* Система має зчитувати з файла список в’язнів та список камер, записувати зміни у ньому за необхідності;
* Система має зчитувати з файла статті з крмимінального кодексу України;
* Система має звіряти введений пароль з паролем, зчитаним з файлу;
* Система повинна підраховувати вік, кількість вільних місць у камері, текст статті, дату визволення згідно з введеними даними.

1.4 Нефункціональні вимоги

Серед нефункціональних вимог слід зазначити наступні:

* Легкість та простота використання;
* Легкість переміщення;
* Цілісність;
* Ефективність та стійкість до збоїв;
* Взаємодія системи і зовнішнього світу;
* Обмеження дизайну і реалізації.

Серед обмежень дизайну і реалізації слід зазначити наступні:

* + Використання платформи .NET Framework 4.5;
  + Використання мови програмування C#;
  + Використання Windows Forms.

1.5 Опис інфтерфейсу користувача

Після запуску програми відкривається вікно авторизації (рис. 1.1), у якому є вожливість війти як гість або ввести пароль та війти як начальник.

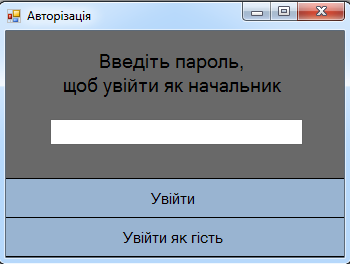


Рисунок 1.1 – Вікно авторизації

У цьому вікні є такі елементи:

* Поле введення паролю;
* Кнопка “Увійти”;
* Кнопка “Увійти як гість”.

Якщо натиснути кнопку “Увійти як гість”, то відкриється головне вікно персоналу (рис. 1.2). Якщо натиснути кнопку “Увійти” та введен дійсний пароль, то відкриється головне вікно начальника (рис. 1.3).

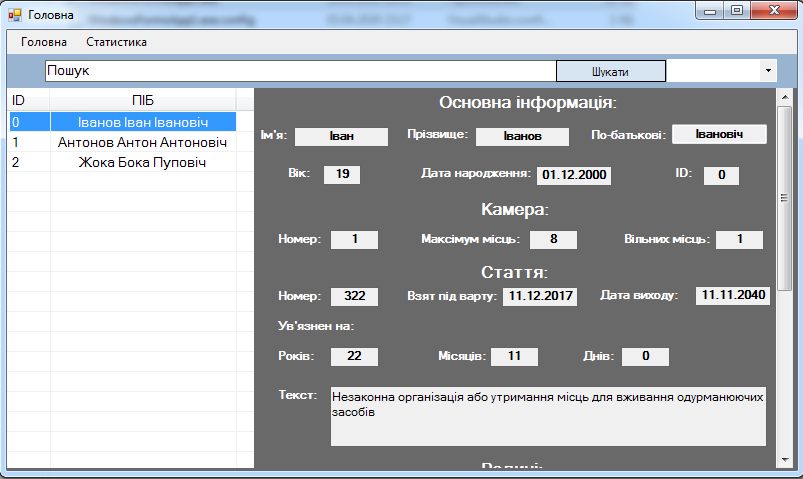


Рисунок 1.2 - Головне вікно (персонал)

У цьому вікні є такі елементи:

* Кнопка “Головна”;
* Кнопка “Статистика”;
* Поле вводу параметра для пошуку, кнопка “Шукати” та випадаюче меню вибору параметру;
* Таблиця з в’язнями;
* Панель з усією інформацією, серед якої є дві таблиці зі списком родичів та списком рис характеру.

Кнопка “Головна” відкриває голвне вікно поточного користувача (рис 1.2 або рис 1.3), кнопка “Статистика” відкриває вікно статистики (рис. 1.4).

Кнопка “Шукати” оновлює список в’язнів.

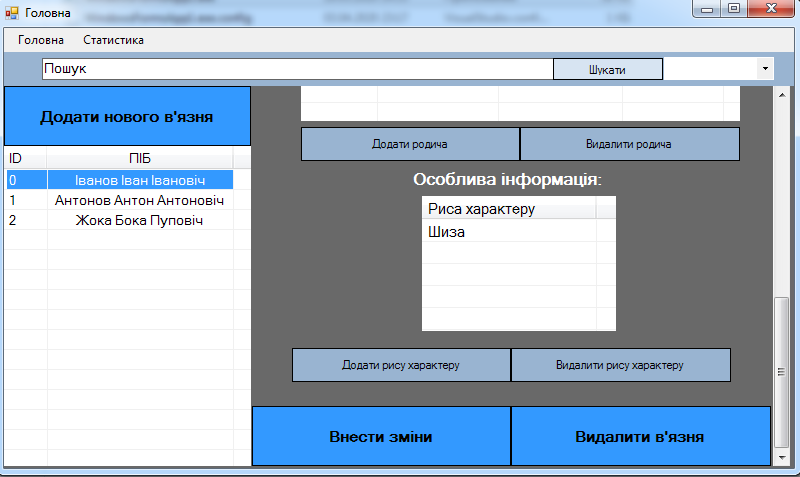


Рисунок 1.3 – Головне вікно (начальник)

У цьому вікні є всі елементи головного вікна персоналу (рис. 1.2), до яких додається ряд кнопок:

* Кнопка “Додати нового в’язня”;
* Кнопка “Внести зміни”;
* Кнопка “Видалити в’язня”;
* Кнопка “Додати родича”;
* Кнопка “Видалити родича”;
* Кнопка “Додати рису характеру”;
* Кнопка “Видалити рису характеру”.

Кнопка “Додати нового в’язня” відкриває вікно додавання нового в’язня (рис 1.5), кнопка “Додати родича” відкриває вікно додавання нового родича (рис 1.6), кнопка “Додати рису характеру” відкриває вікно додавання нової риси характеру (рис 1.7).

Кнопка “Внести зміни” оновлює список в’язнів та панель з інформацією про в’язня, кнопка “Видалити родича” оновлює список родичів, кнопка “Видалити рису характеру” оновлює список рис характеру.

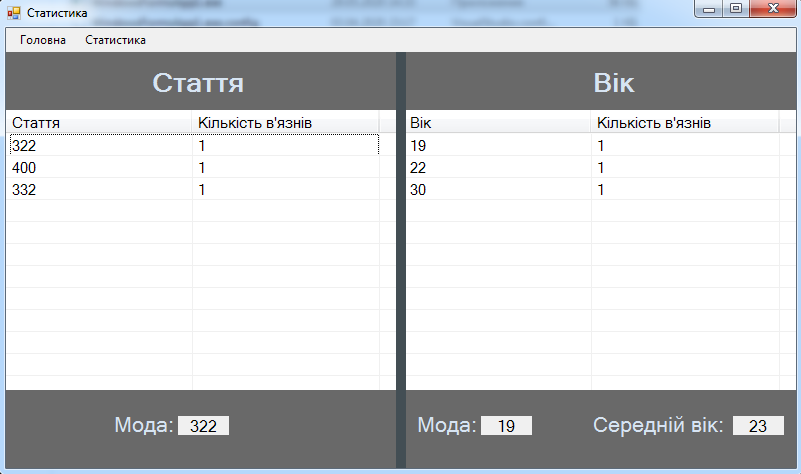


Рисунок 1.4 – Вікно статистики

У цьому вікні є такі елементи:

* Кнопка “Головна”;
* Кнопка “Статистика”;
* Панель зі статистични даними, серед яких є два списку.

Кнопка “Головна” відкриває голвне вікно поточного користувача (рис 1.2 або рис 1.3), кнопка “Статистика” відкриває вікно статистики (рис. 1.4).

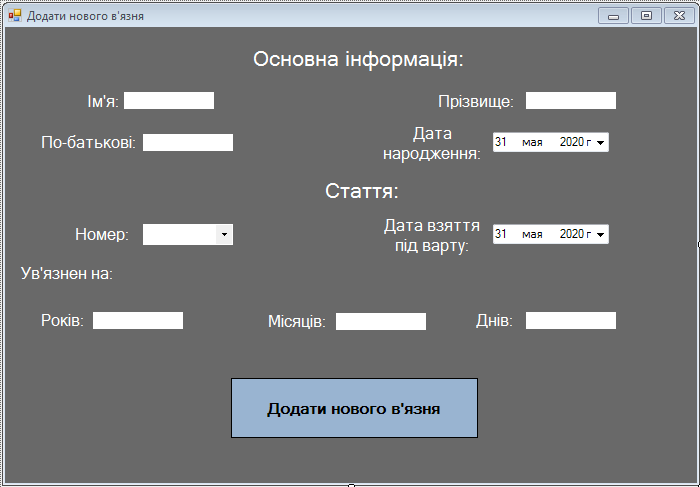


Рисунок 1.5 – Вікно додавання нового в’язня

У цьому вікні є такі елементи:

* Поля для введеня даних;
* Кнопка “Додати нового в’язня”.

Кнопка “Додати нового в’язня” закриває вікно додавання нового в’язня (рис. 1.5), якщо усі поля коректно введені.

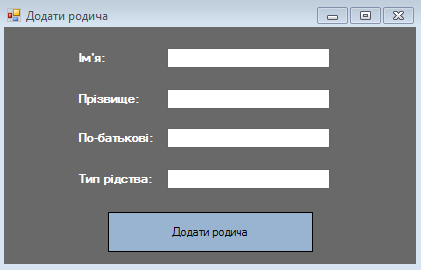


Рисунок 1.6 – Вікно додавання нового родича

У цьому вікні є такі елементи:

* Поля введеня даних;
* Кнопка “Додати родича”.

Кнопка “Додати родича” закриває вікно додавання нового родича (рис. 1.6), якщо усі поля коректно введені.

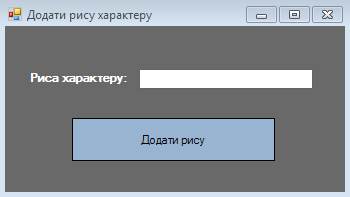


Рисунок 1.7 – Вікно додавання риси характеру

У цьому вікні є такі елементи:

* Поле введеня даних;
* Кнопка “Додати рису”

Кнопка “Додати рису” закриває вікно додавання риси характеру (рис 1.7), якщо поле коректно введено.

1. **ПРОЕКТНА СПЕЦИФІКАЦІЯ**

2.1 Діаграма класів

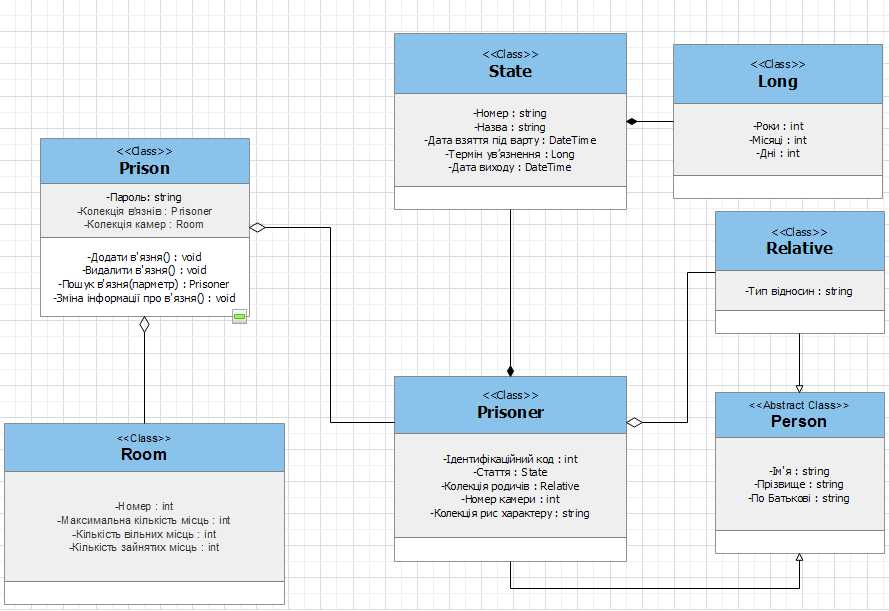


Рисунок 2.1 – Діаграма класів

2.2 Об’єктна структура програми

Програма складається з десяти класів та вісьми форм. Сім форм були розгянуті у попередньому розділі. Одна потрібна для наслідування.

На рисунку 2.2 наведено архітектуру проекту.

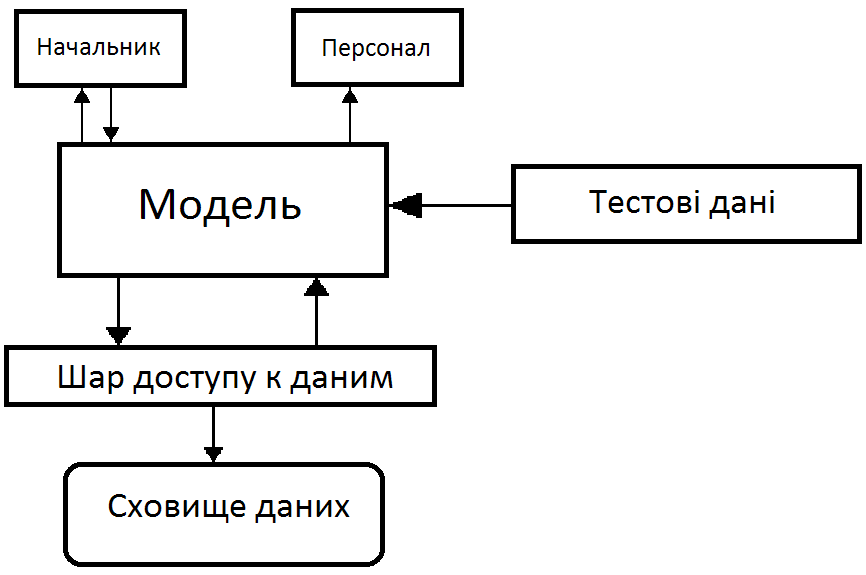


Рисунок 2.2 – Архітектура проекту

2.2.1 Абстрактний клас Person

Абстрактний клас Person описує людину. Він має три відкриті властивості:

//Ім'я

public string Name { get; set; }

//Прізвище

public string Surname { get; set; }

//По-батькові

public string Patronymic { get; set; }

2.2.2 Клас Prisoner

Клас Prisoner описує в’язня і є спадкоємцем абстрактного класу Person. Крім вище зазначених властивостей має сім відкритих властивостей:

//ID

public int ID { get; set; }

//Дата народження

public DateTime BirthDate { get; set; }

//Вік

public int Age => (int)((DateTime.Today - BirthDate).TotalDays / 365.2425);

//Список родичів

public List<Relative> Relatives;

//Список рис характеру

public List<string> Character;

//Стаття

public State State;

//Номер камери

public int RoomNumber;

2.2.3 Клас Relative

Клас Relative описує родича і є спадкоємцем абстрактного класу Person. Крім вище зазначених властивостей має одну відкриту властивість:

//Тип рідства

public string Relation { get; set; }

2.2.4 Клас State

Клас State описує статтю в’язня і срок у в’язниці. Він має п’ять відкритих властивостей:

//Номер статті

public string Number { get; set; }

//Текст статті

public string Text => Global.Codex[Number];

//Дата взяття під варту

public DateTime GuardDate { get; set; }

//Довжина сроку

public Length Length;

//Дата визволення

public DateTime ReleaseDate

{

get

{

DateTime releaseDate = new DateTime(GuardDate.Year, GuardDate.Month, GuardDate.Day);

releaseDate = releaseDate.AddYears(Length.Years);

releaseDate = releaseDate.AddMonths(Length.Months);

releaseDate = releaseDate.AddDays(Length.Days);

return releaseDate;

}

}

2.2.5 Клас Length

Клас Length описує довжину сроку у в’язниці. Він має три відкриті властивості:

//Кількість років

public int Years { get; set; }

//Кількість місяців

public int Months { get; set; }

//Кількість днів

public int Days { get; set; }

2.2.6 Клас Room

Клас Room описує камеру у в’язниці. Він має три відкриті властивості:

//Номер камери

public int Number;

//Максимальна кількість в'язнів у камері

public int Max;

//Кількість вільних місць у камері

public int Free;

2.2.7 Клас Prison

Клас Prison описую в’язницю. Він має чотири відкриті властивості:

//Пароль для авторизації

public string Password { get; set; }

//Список в'язнів

public List<Prisoner> Prisoners;

//Список камер

public List<Room> Rooms;

//Максимальна кількість в'язнів

public int MaxPrisoners => Rooms.Aggregate(0, (a, room) => a + room.Max);

Клас Prison має десять методів (додаток A):

* public void Add(string name, string surname, string patronymic, DateTime birth, string stateNumber, DateTime guardDate, int years, int months, int days) – додає нового в’язня у колекцію в’язнів;
* public void Remove(int id) – видаляє в’язня з колекції в’язнів;
* public List<Prisoner> FindByID(string id) – шукає в’язня за ID;
* public List<Prisoner> FindByStateNumber(string number) – шукає в’язнів за номером статті;
* public List<Prisoner> FindByRoomNumber(string number) – шукає в’язнів за номером камери;
* public List<Prisoner> FindByName(string name) – шукає в’язнів за ПІП;
* public void ChangeRoom(Prisoner prisoner, int n) – змінює камеру у в’язня;
* public void Load() – завантажує дані;
* public void Save() – сберегає дані;
* public Stats DoStatistics() – підраховує статистику.

2.2.8 Клас Stats

Клас Stats описує підраховану статистику. Він має п’ять відкритих властивостей:

//Словник для статті

public Dictionary<string, int> State = new Dictionary<string, int>();

//Словник для віку

public Dictionary<int, int> Age = new Dictionary<int, int>();

//Мода статтей

public string MaxState;

//Мода віків

public int MaxAge;

//Середній вік

public int AverageAge = 0;

2.2.9 Клас Dao

Клас Dao служить для доступу до даних. Він має три закриті константи:

//Шлях до файлу з в'язницею

private const string pathPrison = "Prison.xml";

//Шлях до файлу з паролем

private const string pathPassword = "Password.txt";

//Шдях до кримінального кодексу України

private const string pathCodex = "Codex.txt";

Клас Dao має два методи (додаток Б):

* public void Load() – завантажує дані;
* public void Save() – сберегає дані.

2.2.10 Статичний клас Global

Статичний клас Global – є глобальним станом програми. Він має чотири статичні властивості:

//Об'єкт в'язниці

public static Prison Prison = new Prison();

//Словник кримінального кодексу України

public static Dictionary<string, string> Codex = new Dictionary<string, string>();

//Головна форма

public static Main Main;

//Форма статистики

public static Statistics Statistics = new Statistics();

1. **ІНСТРУКЦІЯ КОРИСТУВАЧА**

3.1 Запуск та вікно авторизації

Після запуску програми відкривається вікно авторизації (рис. 1.1).

Якщо ввести дійсний пароль та натиснути кнопку “Увійти”, відкриється головне вікно начальника (рис 1.3). Від початку пароль дорівнює “1”, але його можна змінити у файлі “Password.txt”.

Якщо без введення паролю натиснути кнопку “Увійти як гість”, відкриється вікно персоналу (рис. 1.2).

3.2 Елементи меню

І в меню персоналу, і в меню начальника є однакові елементи, які переключають вікна.

Кнопка “Головна” відкриває головне вікно поточного користувача (рис 1.2 або рис 1.3).

Кнопка “Статистика” відкриває вікно статистики (рис. 1.4).

3.3 Головне вікно персоналу: пошук в’язнів та перегляд інформації про в’язня

Для пошуку в’язнів спочатку слід обрати з випадаючего списку параметр пошуку: ID, ПІП, номер статті, номер камери. Далі після введеня у поле аргументу для пошуку і натискання кнопки “Пошук”, у списку в’язнів будуть всі в’язні, які задовальняють умовам пошуку. Якщо аргумент для пошуку не введений, програма покаже усіх в’язнів, які є у в’язниці.

Для того щоб подивитися інформацію про в’язня, потрібно натиснути на нього у списку в’язнів. Інформація про в’язня поділена на пункти:

* Основна інфориація: ім’я, прізвище, по-батькові, вік, дата народження, ID;
* Камера: номер, максімум місць, вільних місць;
* Стаття: номер, дата взяття під варту, дата виходу, термін ув’язнення (роки, місяці, дні), текст статті;
* Родичі – список родичів, про яких відомо: ім’я, прізвище, по-батькові, тип рідства;
* Особлива інформація – список рис характеру.

3.4 Головне вікно начальника: додавання, видалення та зміна інформації про в’язнів

У головному вікні начальника є такі ж самі елменти, як і у головному вікні персоналу. До них додається ряд кнопок, які потрібні для маніполювання списком в’знів.

Для додавання нового в’язня потрібно натиснути кнопку “Додати нового в’язня”, після чого відкривається вікно додавання нового в’язня (рис. 1.5). У цій формі подрібно ввести ім’я, прізвище, по-батькові, дати народження, номера статті, дати взяття під варту та терміну ув’язнення. Якщо всі поля заповнені коректно, у список додається новий в’язень. Програма автоматично заповнює поля: вік, ID, номер камери, кількість вільних місць у камері, максимальна кількість місць у камері, дату виходу, текст статті згідно з кримінальним кодексом України. Слід відзначити, що додається перша камера, у якій є вільні місця. Далі форма закривається, а уголовному вікні оновлюється список в’язнів.

Для видалення в’язня потрібно спочатку обрати його зі списку в’язнів, далі натиснувши на кнопку “Видалити в’язня”, треба підвердити це у вспливаючему вікні. Після цього в’язень буде видален, а список в’язнів оновиться.

Для зміни інформації про в’язня треба обрати його зі списку в’язнів. Можна змінити такі поля: ім’я, прізвище, по-батькові, дату народження, номер камери, номер статті, дату взяття під варту, термін ув’язнення. Після внесення даних у поля потрібно натиснути кнопку “Внести зміни”. Інформація про в’язня буде оновлена і відображена на страниці. Інші поля будуть автоматично змінені згідно зі введеними даними.

3.5 Головне вікно начальника: додавання та видалення родичів та рис характеру

Для додавання нового родича потрібно спочатку обрати в’язня зі списку в’язнів. Далі слід натиснути на кнопку “Додати родича”, після чого відкриється вікно додавання нового родича (рис. 1.6). У цьому вікні потрібно заповнити такі поля: ім’я, прізвище, по-батькові, тип рідства. Якщо всі поля заповнені коректно, після натискання на кнопку “Додати родича” вікно закриється, а список родичів буде оновлен.

Для видалення родича потрібно спочатку обрати в’язня зі списку в’язнів, далі обравши в’язня слід обрати родича, якого потрібно видалити, зі списку родичів. Натиснувши кнопку “Видалити родича” та підтвердивши це у вспливаючему вікні, родич буде видален, а список родичів оновиться.

Для додавання риси характеру потрібно спочатку обрати в’язня зі списку в’язнів. Далі слід натиснути на кнопку “Додати рису характеру”, після чого відкриється вікно додавання риси характеру (рис. 1.7). У цьому вікні потрібно заповнити поле з рисою характеру. Якщо поле заповнено коректно, після натискання кнопки “Додати рису”, вікно закриється, а список рис характеру буде оновлен.

Для видалення риси характеру потрібно спочатку обрати в’язня зі списку в’язнів, далі обравши в’язня слід обрати рису характеру, яку потрібно видалити, зі списку рис характеру. Натиснувши кнопку “Видалити рису характеру” та підтвердивши це у вспливаючему вікні, риса характеру буде видалена, а список родичів оновиться.

3.6 Вікно статистики

Вікно статистики (рис. 1.4) поділено на два розділи: стаття та вік. У кожному розділі є список, який відображує парметр та кількість в’язнів з цим параметром. Мода показує параметр, який зустрічається найбільше. У розділі з віком також відображен середній вік.

3.7 Збереження даних та вихід з програми

Для виходу з програми потрібно натиснути системну кнопку “Закрити”. Під час закриття програми всі внесені зміни будуть автоматично збережені.

**ВИСНОВКИ**

У ході виконання курсової роботи були освоєні методи створення програмних систем на засадах об’єктно-орієнтованого програмування, набулися навички роботи з програмною системою, створення в’язків між об’єктами.

У результаті отримана програма під назвою “Довідник начальника в’язниці”, яка дозволяє у зручному форматі переглядати інформацію про в’язнів, робити пошук за параметрами: ПІП, ідентифікаційний код, номер статті, номер камери та дивитися статистику про склад в’язниці. Повний доступ до функцій програми отримує авторизований користувач - начальник в’язниці. Він може робити все, вище зазначене, а також додавати, видаляти та змінювати інформацію про в’язнів.

Програма написана на мові програмування C# на платформі .NET Framework 4.5 з використанням технологій Windows Forms у середі розробки Microsoft Visual Studio 2019.

Усі дані програми зберігаються локально у файлах XML та TXT. Програма може бути поліпшена застосуванням справжньої бази даних.

Галузь застосування – в’язниці та інші місця позбавлення волі.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

1. Полное руководство по языку программирования С# 7.0 и платформе .NET 4.7 [Електронний ресурс] – Режим доступу до ресурсу: https://metanit.com/sharp/tutorial/.
2. Бондарев В. М. Объектно-ориентированное программирование на С# [Текст] : учеб. Пособие / В. М. Бондарев. – Харьков : СМИТ, 2009. –224 с. – Библиогр. : ISBN 978-966-2028-34-8.
3. Microsoft Developer Network. Библиотека MSDN / MSDN – сеть разработчиков Microsoft [Електронний ресурс] – Режим доступу до ресурсу: https://msdn.microsoft.com/ru-ru/.
4. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд. [Текст]: пер. с англ./ Э. Троелсен.- М.: Вильямс, 2013 - 1312 с. – ISBN 978-5-8459-1814-7, 978-1-43-024233-8.
5. Либерти Дж. Программирование на С# [Текст] : пер. с англ. / Дж. Либерти. – 2-е изд. – СПб. ; М. : Символ-Плюс, 2003. – 688 с.

ДОДАТОК А

Методи класу Prison

//Додавання нового в'язня

public void Add(string name, string surname, string patronymic, DateTime birth, string stateNumber, DateTime guardDate, int years, int months, int days)

{

if (Prisoners.Count < MaxPrisoners)

{

var prisoner = new Prisoner(name, surname, patronymic, Prisoners.Count);

prisoner.BirthDate = birth;

prisoner.State = new State(stateNumber, guardDate, years, months, days);

var tempRoom = Rooms.First(r => r.Free != 0);

prisoner.RoomNumber = tempRoom.Number;

tempRoom.Free--;

Prisoners.Add(prisoner);

}

else

{

throw new ArgumentException();

}

}

//Видалення в'язня

public void Remove(int id)

{

var tempRoom = Rooms.Find(r => r.Number == Prisoners[id].RoomNumber);

tempRoom.Free++;

Prisoners.RemoveAt(id);

for (int i = 0; i < Prisoners.Count; i++)

{

Prisoners[i].ID = i;

}

}

//Пошук в'язня за ID

public List<Prisoner> FindByID(string id)

{

int idInt;

bool success = Int32.TryParse(id, out idInt);

if (success)

return Prisoners.FindAll(p => p.ID == idInt);

else

throw new ArgumentException();

}

//Пошук в'язня за номером статті

public List<Prisoner> FindByStateNumber(string number)

{

return Prisoners.FindAll(p => p.State.Number == number);

}

//Пошук в'язня за номером камери

public List<Prisoner> FindByRoomNumber(string number)

{

int numberInt;

bool success = Int32.TryParse(number, out numberInt);

if (success)

return Prisoners.FindAll(p => p.RoomNumber == numberInt);

else

throw new ArgumentException();

}

//Пошук в'язня за ПІП

public List<Prisoner> FindByName(string name)

{

return Prisoners.FindAll(p =>

{

string SFP = p.Surname + " " + p.Name + " " + p.Patronymic;

return SFP == name;

});

}

//Зміна камери

public void ChangeRoom(Prisoner prisoner, int n)

{

var currentRoom = Rooms.Find(r => r.Number == prisoner.RoomNumber);

currentRoom.Free++;

var tempRoom = Rooms.Find(r => r.Number == n);

prisoner.RoomNumber = tempRoom.Number;

tempRoom.Free--;

}

//Завантаження даних

public void Load()

{

new Dao().Load();

}

//Сбереження даних

public void Save()

{

new Dao().Save();

}

//Підрахунок статистики

public Stats DoStatistics()

{

Stats stats = new Stats();

foreach (var prisoner in Prisoners)

{

string state = prisoner.State.Number;

int age = prisoner.Age;

if (stats.State.ContainsKey(state))

stats.State[state]++;

else

stats.State.Add(state, 1);

if (stats.Age.ContainsKey(age))

stats.Age[age]++;

else

stats.Age.Add(age, 1);

stats.AverageAge += age;

}

stats.AverageAge /= Prisoners.Count;

stats.MaxState = stats.State.First(x => x.Value == stats.State.Values.Max()).Key;

stats.MaxAge = stats.Age.First(x => x.Value == stats.Age.Values.Max()).Key;

foreach (var age in stats.Age.Keys)

{

if (stats.Age[age] > stats.Age[stats.MaxAge]) stats.MaxAge = age;

}

return stats;

}

ДОДАТОК Б

Методи класу Timetable

*//Завантаження даних*

public void Load()  
{  
 XmlSerializer xml = new XmlSerializer(typeof(Prison));  
   
 using (FileStream fs = new FileStream(pathPrison, FileMode.OpenOrCreate))  
 {  
 Global.Prison = (Prison)xml.Deserialize(fs);  
 }  
   
 TextReader readerPassword = new StreamReader(pathPassword, Encoding.Default);  
 Global.Prison.Password = readerPassword.ReadToEnd();  
 readerPassword.Close();  
   
 TextReader readerCodex = new StreamReader(pathCodex, Encoding.Default);  
 string[] arr = readerCodex.ReadToEnd().Split('\n');  
 readerCodex.Close();  
   
 foreach (var item in arr)  
 {  
 string[] elements = item.Split(':');  
 Global.Codex.Add(elements[0], elements[1]);  
 }  
}  
  
*//Збереження даних*public void Save()  
**{** XmlSerializer xml = new XmlSerializer(typeof(Prison));  
   
 using(FileStream fs = new FileStream(pathPrison, FileMode.Create))  
 {  
 xml.Serialize(fs, Global.Prison);  
 }  
}

ДОДАТОК В

Посилання на GitHub

<https://github.com/yura2100/OOP-Kursas-Prison>