Архітектура

Програма буде створюватися на мові С# з використанням середи розробки Microsoft Visual Studio 2019 на платформі .NET Framework 4.7.2 з використанням інтерфейсу програмування Windows Forms.

Вибір мови обґрунтован об’єктно-орієнтованим підходом до написання програми. На цей час мова C# є однією з найвідоміших, зручних і повноцінних серед об’єктно-орієнтованих мов програмування.

Під час проектування розглядалося 2 платформи, з якими мова С# має найкращу взаємодію, а саме .Net Core та .NETT Framework. Наш вибір пав на останню, оскільки на ній більше керуючих елементів і більш високого рівня розвитку платформи. На жаль, це означає що додаток буде працювати лише на машинах с операційною системою Windows.

Для реалізації користувальницького інтерфейсу було вирішено використати графічний інтерфейс, тому що він простий для використання й інтуїтивно зрозумілий. Для реалізації інтерфейсу будемо використати інтерфейс Windows Forms, який є частиною платформи .NET Framework. З його допомогою в нас буде можливість створити повноцінний інтерфейс для користування додатком.

**Структура проекту**

Не має ніякого сенсу ділити проект, описаний в специфікації, на дві програми, оскільки функції користувачів дуже схожі. Персонал може лише переглядати вже записані дані, а начальник може їх змінювати та додавати нові. Крім того, для тестування програми на етапі програмування доречно створити де-кілько тестових даних для перевірки правильності роботи програми.

На цій підставі можемо скласти наступну структуру проекту.

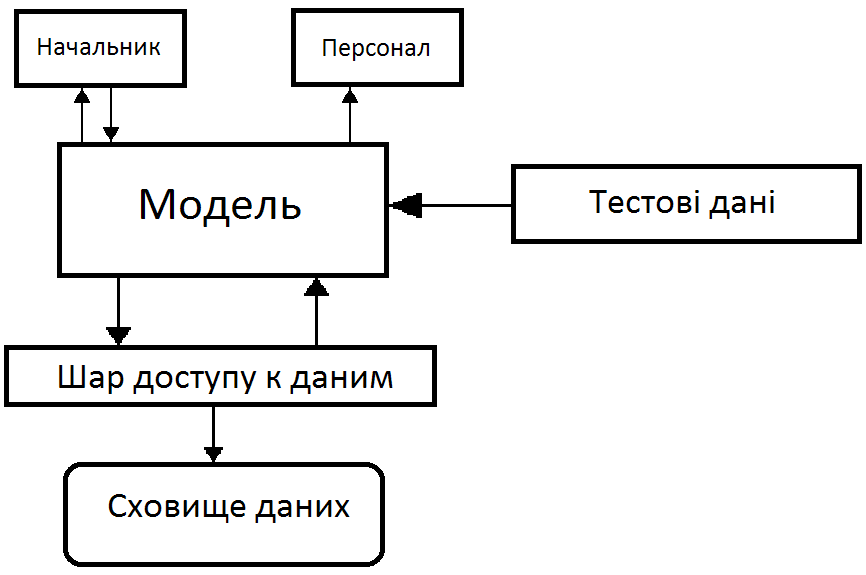


Рисунок 1 – Схема структури проекту

**Об’єктна модель**

**Опис**

У **базі** **в’язниці** містяться **в’язні**.

**Начальник** додає, видаляє та змінює інформацію у **списку в’язнів**.

**Персонал** може лише дивитися інформацію про **в’язнів**.

Про в’язня відомо: **ідентифікаційний код, ПІП, дата народження, вік, стаття, інформація про родичів, інформація про камеру, особливі риси характеру.**

Про статтю в’язня відомо**: номер статті, назва статті, дата взяття під варту, термін ув’язнення, дата виходу.**

Про родичів в’язня відомо: **ПІП, вид відносин з ув’язненним.**

Про камеру відомо: **номер, максимальна кількість місць, кількість вільних місць, кількість зайнятих місць.**

В’язниця – це колекція в’язнів та колекція всіх камер.

Інформація про родичів – це колекція родичів в’язня.

Особливі риси характеру – це колекція рис характеру.

**Методи:**

На даному етапі до головних методів ми можемо віднести:

1. Додавання нового в’язня
2. Видалення в’язня
3. Пошук в’язня за параметром (ПІП, ідентифікаційний код, номер статті, номер камери)
4. Зміна інформації про в’язня

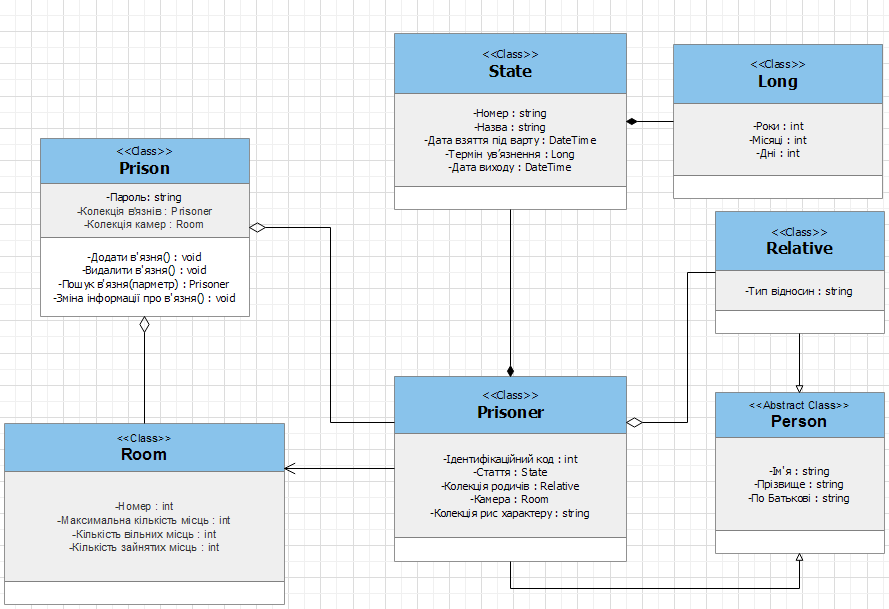


Рисунок 2 – Діаграма класів

**Сховище та доступ до даних**

Всі дані, помічені словом “база” будуть представлені у виді окремих XML файлів у каталозі проекту. Дані будуть перекладатися в XML файл за допомогою серіалізації з використанням бібліотеки Xml. Захист даних у рамках проекту не розглядається.

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