**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Природничо-гуманітарний фаховий коледж**

**ДВНЗ «Ужгородський національний університет»**

***Реєстраційний номер***

**Дата реєстрації**

Магей Юрій Юрійович

студент ІІI курсу

групи КН-31

денної форми навчання

**Розробка програми для обліку замовлень «Галерея дверей» Курсовий проект з дисципліни**

«Об’єктно-орієнтоване програмування»

**Допущено до захисту Науковий керівник:**

Дата « » 2021 р.викл. Лукавська А.В.

Підпис

**Робота захищена « » 2021 р.**

**з оцінкою « »**

**Підписи членів комісії**

Ужгород 2021

.

ЗМІСТ

ВСТУП 3

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

2

КП. 233129МБД.21.000.ПЗ

Розроб.

*Магей Ю.Ю.*

Перевір.

Лукавська А.В.

Реценз.

Н. Контр.

Кравчук М.І.

Затверд.

Александра Н.В.

Розробка програми для обліку замовлень

«Галерея дверей»

Літ.

Аркушів

91

ПГК ДВНЗ «УжНУ» КН-31

[1. Теоретична частина 4](#_Toc74289082)

[1.1 Статистика використання програм, та її призначення 4](#_Toc74289083)

[1.2 Огляд аналогів 5](#_Toc74289084)

[1.3 Опис предметної області та постановка задачі 6](#_Toc74289085)

[1.4 Обґрунтування вибору методу розв’язку задачі 6](#_Toc74289086)

[1.5 Огляд засобів розробки 7](#_Toc74289087)

[2. ПРАКТИЧНА ЧАСТИНА 9](#_Toc74289088)

[2.1 Проектна частина 9](#_Toc74289089)

[2.1.1 Інформаційна структура моделі 9](#_Toc74289090)

[2.1.2 Опис алгоритму 9](#_Toc74289091)

[2.1.3 Функціональна часттина програми 10](#_Toc74289092)

[2.1.4 Фізична структура проекту 13](#_Toc74289093)

[2.1.5 Опис структури даних та використаних змінних 13](#_Toc74289094)

[2.1.6 Тестування програми 15](#_Toc74289095)

[2.2 Експлуатаційна частина 19](#_Toc74289096)

[2.2.1 Вимоги до технічних засобів 19](#_Toc74289097)

[2.2.2 Склад інсталяційного пакету та інструкція 20](#_Toc74289098)

[2.2.3 Опис інтерфейсу 20](#_Toc74289099)

[2.2.4 Ієрархічна структура продукту 20](#_Toc74289100)

[2.2.5 Керівництво користувача та приклади використання 22](#_Toc74289101)

[програмного продукту 22](#_Toc74289102)

[ВИСНОВОК 23](#_Toc74289103)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 24](#_Toc74289104)

ДОДАТОК 1 ЛІСТИНГ 25

ДОДАТОК 2 ДОКУМЕНТАЦІЯ 49

Завданням даного курсового проекту стала розробка програмного продукту, який дасть змогу маніпулювати з записами в базі даних.

Для реалізації цієї задачі, буде використано мову програмування Java, її бібліотеку Java Swing для створення графічного інтерфейсу користувача, а також систему керування базами даних (далі — СКБД) MySQL.

Метою виконання курсового проекту є систематизація та закріплення отриманих теоретичних знань, поглиблене вивчення можливостей мови програмування Java, закріплення отриманих та набуття нових практичних навичок у постановці задачі, алгоритмізації, програмуванні, обробці інформації, використанні сучасних інформаційних технологій для організації інтерфейсу, обробці і збереженні результатів обчислень, аналізі та виявленні можливих шляхів підвищення ефективності отримання результатів.

Предметом дослідження виступає мова програмування Java та середовище розробки NetBeans IDE

**1. Теоретична частина**

**1.1 Статистика використання програм, та її призначення**

Двері — щитова конструкція та її облаштунки (коробка), яка перекриває чи звільняє отвір у стіні, що його відкривають, щоб входити — виходити.

Двері призначені, щоб перекривати (положення «зачинено») та звільняти (положення «відчинено») отвір (вхід-вихід) в стіні будівлі та проходи між внутрішніми приміщеннями. В положенні «зачинено» двері захищають внутрішній простір від вітру, пилюки, комах, диких чи домашніх тварин, а також сторонніх осіб.

Ранні будівлі людей не були міцними, тому першими дверима були завіси: шматки шкір впольованих тварин, зроблені з трави чи очерету циновки, пізніше — повстяні полотна та килими (у кочовиків в юртах). Відносно міцними були щити з тонких гілок, вплетених в середину рами рамки з більш товстих паль. Такі конструкції перекриття проходів легко відсовувались і часто зовсім не мали зв'язку зі стінами. Їх встановлювали на ніч, чи виходячи на довгий час.

З часом, одну сторону рамної конструкції почали прив'язувати до вертикальних паль проходу жмутиком трави, шворками чи ремінцями. Так її легко було повернути до стіни, тим самим відкривши прохід. Саме звідси з'явивсь термін «петлі». На ніч такі двері так саме прив'язували з другого боку.

Міцні двері з деревини досить пізній винахід людства, що став потрібним в місцях щільного міжродового заселення чи важких кліматичних умов. Африканські скотарі масаї і сьогодні мають традиційні домівки з каркасом сплетеним з тонких гілок та обмазаних коров'ячим гноєм. Вони мають відкритий вхід (без дверей), бо вхід до житлового простору здійснюється через невеличку прибудову, вхід в яку під кутом 90 градусів до входу в будівлю. Тварина просто не може там розвернутися

Сучасні щитові двері розрізняють на зовнішні і внутрішні, подвійні й одинарні. За числом дверних стулок двері бувають одностулкові, півторачні і двостулкові. За способом відчиняння можна виокремити: двері, що відчиняються в один або обидва боки; двері обертові (турнікети); двері складані, відкотні і піднімні (шторні). Двері виробляють з дерева, металу, скла. Число дверей і їхні розміри визначають для кожного приміщення залежно від потрібної пропускальності та розмірів устаткування.

У виробничих і складських будинках часто обладнують ворота для проїзду транспортних засобів, забезпечення механізації вантажно-розвантажувальних робіт тощо.

## 1.2 Огляд аналогів

Переглянувши великий список інтернет-ресурсів в пошуках подібного додатку або вебдодатку, знайшов два аналоги першим з них став веб додаток галерея дверей (див. рисунок 1.1)

Перевагою вебдодатку є більший функціонал

Недоліком став складний інтерфейс

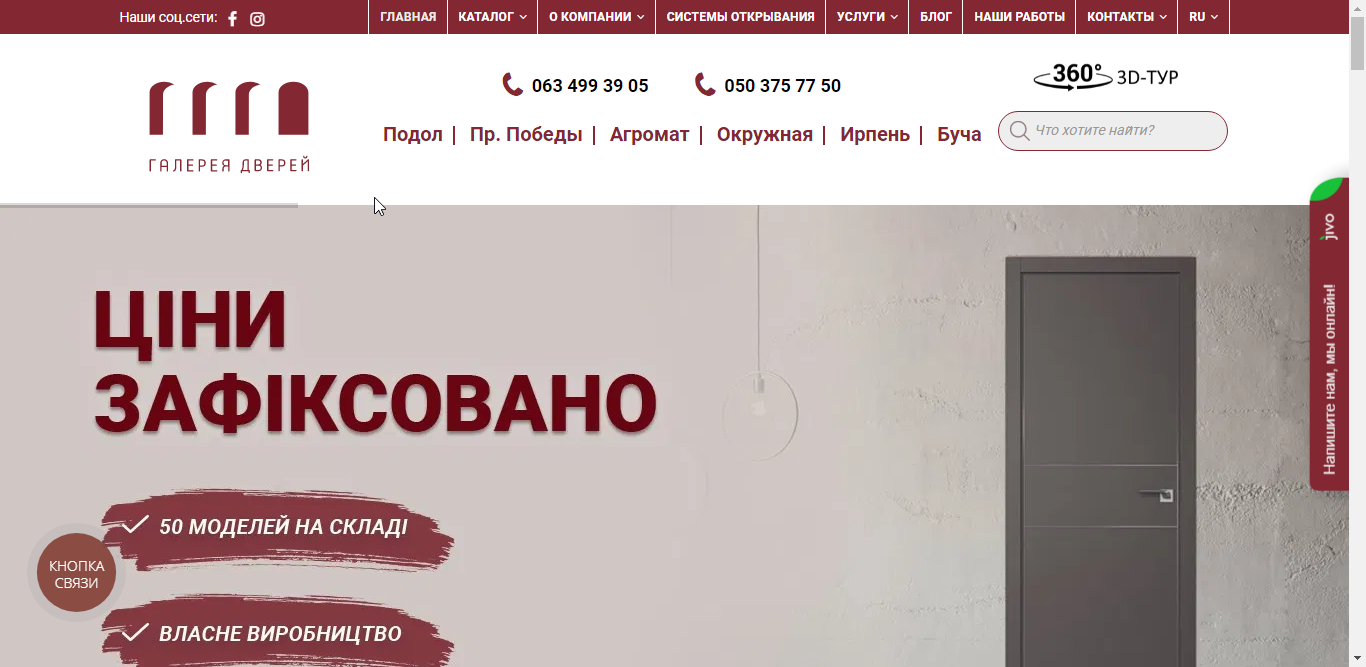


Рисунок 1.1 Галерея дверей

Другим аналогом став стандартний додаток блокнот

Його плюсом являється приємний інтерфейс.

Мінусом являється нагромадженість інтерфейсу.

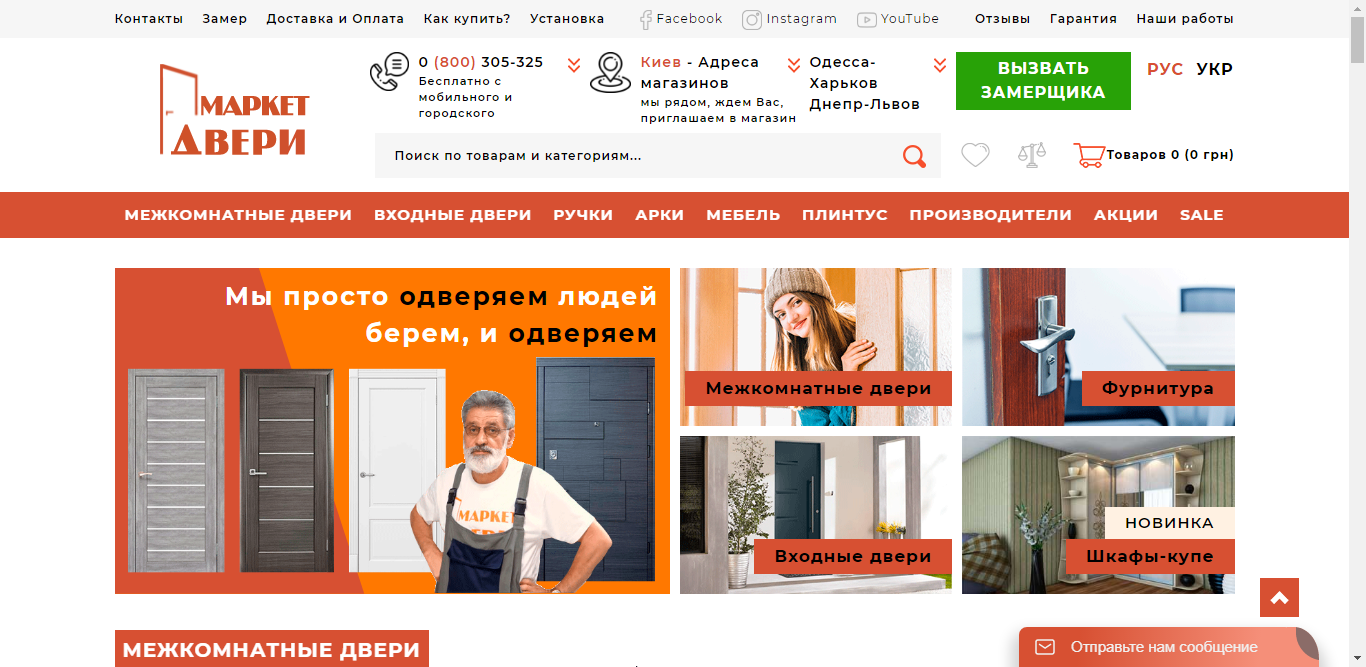


Рисунок 1.2 Головне вікно Блокноту

## 1.3 Опис предметної області та постановка задачі

Предметною областю даного курсового проекту є дверна галерея Так як в даній області вже існує багато програмних рішень, зокрема корпоративних, було вирішено створити саме персональний вид такої програми.

Інтерфейс має бути пристосований для швидкої роботи та легкої адаптації. Крім того необхідно передбачити наступне:

* підключення до бази даних;
* додавання інформації;
* редагування інформації;
* видалення інформації.

Програма має бути спроектована таким чином: користувач повинен підключитися до бази даних, після успішного підключення, користувач може насолодитися всім функціоналом додатку.

**1.4 Обґрунтування вибору методу розв’язку задачі**

Для додавання, оновлення та додавання даних було вибрано метод маніпулювання з тим записом, який вибрав сам користувач. Це використовується для оптимізації використання запитів до бази даних, а також її оновлення.

Редагування виконується в два етапи, спочатку алгоритм бере дані з бази даних додаючи їх в масив, далі відбувається редагування користувачем, після чого дані перезаписуються в базу даних.

Був вибраний метод для створення з’єднання до бази даних. Цей метод базується на створення підключення тоді, коли потрібно виконувати якийсь запит. Коли запит виконано підключення з базою даних закривається.

## 1.5 Огляд засобів розробки

Основним засобом розробки стала мова програмування Java, та бібліотека JavaSwing. Для обміну даними було обрано СКБД MySQL, Робота з БД буде реалізована за допомогою прикладного програмного інтерфейсу (англ. Additional Program Interface, скорочено — API) JDBC.

Java — об’єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Swing — інструментарій для створення графічного інтерфейсу користувача (GUI) мовою програмування Java. Це частина бібліотеки базових класів Java (JFC, Java Foundation Classes). Swing розробляли для забезпечення функціональнішого набору програмних компонентів для створення графічного інтерфейсу користувача, ніж у ранішого інструментарію AWT.

MySQL — вільна система керування реляційними базами даних.

MySQL був розроблений компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця СКБД з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Java DataBase Connectivity (англ. Java DataBase Connectivity — з'єднання з базами даних на Java, скорочено JDBC) — API, який визначає методи, з допомогою яких програмне забезпечення на Java здійснює доступ до бази даних. JDBC — це платформо-незалежний промисловий стандарт взаємодії Java-застосунків з різноманітними СКБД.

# 2. ПРАКТИЧНА ЧАСТИНА

## 2.1 Проектна частина

### 2.1.1 Інформаційна структура моделі

Для роботи програми було використано таку структуру збереження інформації:

* Файли .txt;

Містять налаштування користувача для підключення до бази даних.

* Рисунки .png;

Використовуються для оформлення інтерфейсу програми.

Також було використано базу даних MySQL для збереження всіх даних(рисунок 2.1).

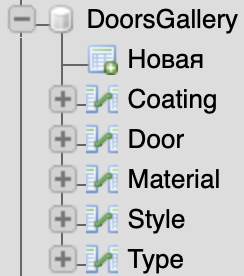


Рисунок 2.1 Структура бази даних проекту

Таблиця «door» — це таблиця, яка використовується для зберігання інформації. Тип таблиці InnoDB, це тип який використовується для зв’язування полів. Ця таблиця має таку структуру (рисунок 2.2).



Рисунок 2.2 Структура таблиці «dor»

Всі інші таблиці використовуються для авто-заповнення полів.

### 2.1.2 Опис алгоритму

Алгоритм роботи інтерфейсу програми побудований наступним чином:

* вибір налаштування «Підключення до бази даних»;
* загрузка головної форми та її компонентів;
* вибір одного з меню на формі;
* вибір або введення необхідних даних;
* виведення обробленої інформації на екран.
* збереження в БД або на ПК

Алгоритм перевірки підключення побудований наступним чином:

Перше — це користувач вносить свої дані для входу, якщо вони успішні, колір індикатора зміниться на зелений, в іншому випадку буде горіти червоним.

Після успішного підключення виконується робота з даними.

Для маніпулювання даними таблиць в базі даних було створено інтерфейс користувача.

Для видалення, оновлення та додавання даних було вибрано алгоритм маніпулювання з тим записом, який ввів/вибрав сам користувач. Це використовується для оптимізації використання запитів до бази даних, а також її оновлення.

Був вибраний алгоритм для створення з’єднання до бази даних. Цей алгоритм базується на створення підключення тоді, коли потрібно виконувати якийсь запит. Коли запит виконано успішно, то підключення з базою даних закривається.

### 2.1.3 Функціональна частина програми

Даний розділ містить матеріал, необхідний для повного розуміння внутрішньої архітектури проекту.

Структуру додатку в середовищі *NetBeans* подано на рисунку 2.9.

Головний пакет — **Main**. Даний пакет містить головні класи для роботи авторизації, головної форми, й інших її пунктів.

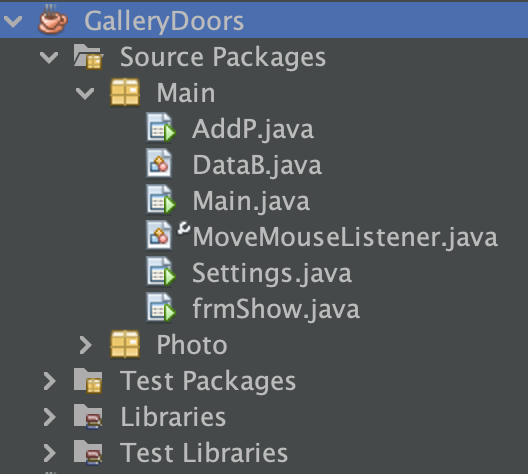


Рисунок 2.3 Внутрішня структура проекту (NetBeans IDE)

Макети вікон додатку зображено на рисунках 2.7 – 2.10



Рисунок 2.4 Макет вікна початку роботи

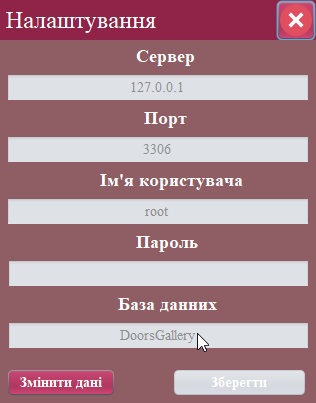


Рисунок 2.5 Макет вікна налаштувань

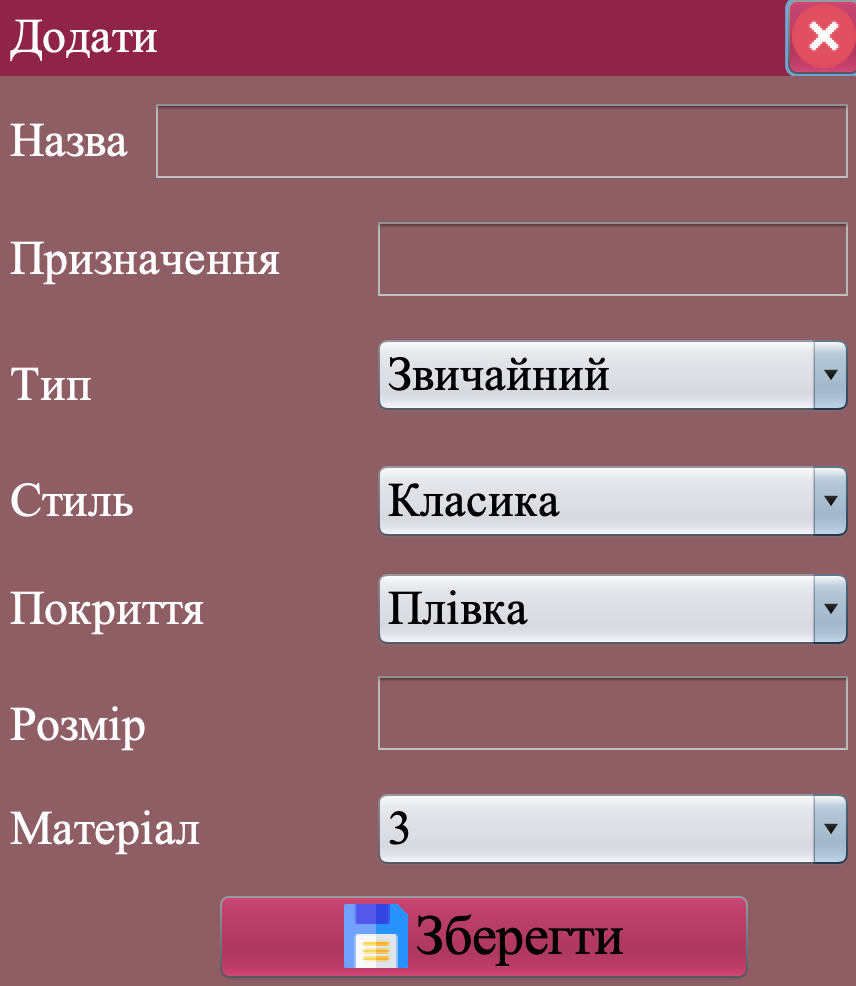


Рисунок 2.6 Макет вікна додавання



Рисунок 2.7 Макет головного робочого вікна програми

### 2.1.4 Фізична структура проекту

Фізична структура проекту складається з наборів файлів, які розміщені по різним директоріям. На рисунку 2.9 показано структуру файлів.

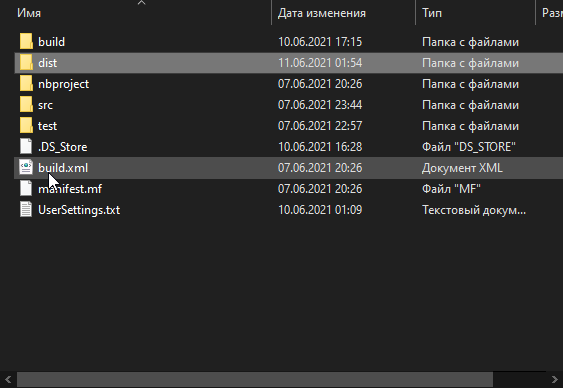


Рисунок 2.8 Структура проекту

Також для роботи з даних в базі даних MySQL, потрібно мати файл під назвою «dorsgallery», це файл в якому зберігається дані на створення бази даних, таблиць і так далі. Також для роботи програми потрібно мати запущений wampserver64 (рисунок 2.9)





Рисунок 2.9 Запущений сервер wampserver64

### 2.1.5 Опис структури даних та використаних змінних

|  |  |
| --- | --- |
| **Бібліотеки** | **Опис** |
| Java.io.\* | Робота з вводом/виводом текстових даних |
| Java.awt.\* | Робота з компонентами: колір, шрифт, розміщення, алгоритм малювання. |
| Java.uti;.\* | Містить платформу наборів, класи набору спадщини, модель подій, дату і засоби часу, інтернаціоналізацію, і різні службові класи (строковий токенізатор, генератор випадкових чисел, і маленький масив). |
| Java.swing.\* | Бібліотека для створення графічного дизайну для програм на мові java. |
| java.sql\* | Бібліотека, яка використовується для підключення програми до бази даних MySQL. За допомогою цієї бібліотеки створювалося підключення до бази даних та її таблиць. Ця бібліотека надає змогу виконувати всі sql запити до бази даних. |
| **Використані елементи:**   * Button (кнопки керування); * TextField (поле для вводу тексту); * Label(напис); * ChoiceBox (випадаючий список); * Pane (контейнер елементів); | |
| **Використані класи:**   * DataB * Addp * Main * MoveMouseListener * frmShow * Settings | |
| **Використані об’єкти:**  Всі об’єкти можна побачити в документації до програмного коду | |
| **Використані події:**  Всі використані події можна побачити в документації до програмного коду | |
| **Використані змінні:**  Всі використані змінні можна побачити в документації до програмного коду | |

### 2.1.6 Тестування програми

Тестування проекту відбудеться на таких етапах як:

* Тестування підключення до бази даних (рисунок 2.10-2.24);
* Хибні дані

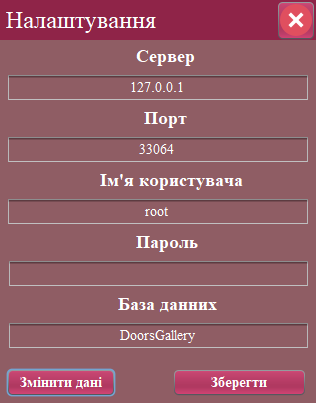


Рисунок 2.10 Хибні дані



Рисунок 2.11 Хибні дані

* Коректні дані



Рисунок 2.12 Коректні дані

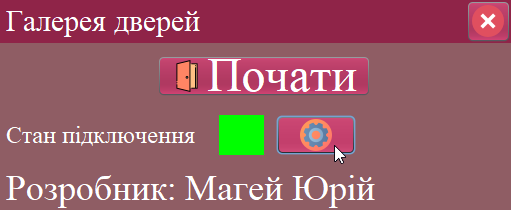


Рисунок 2.13 Коректні дані

* Тест додавання курсового проекту (рисунок 2.15-2.17);

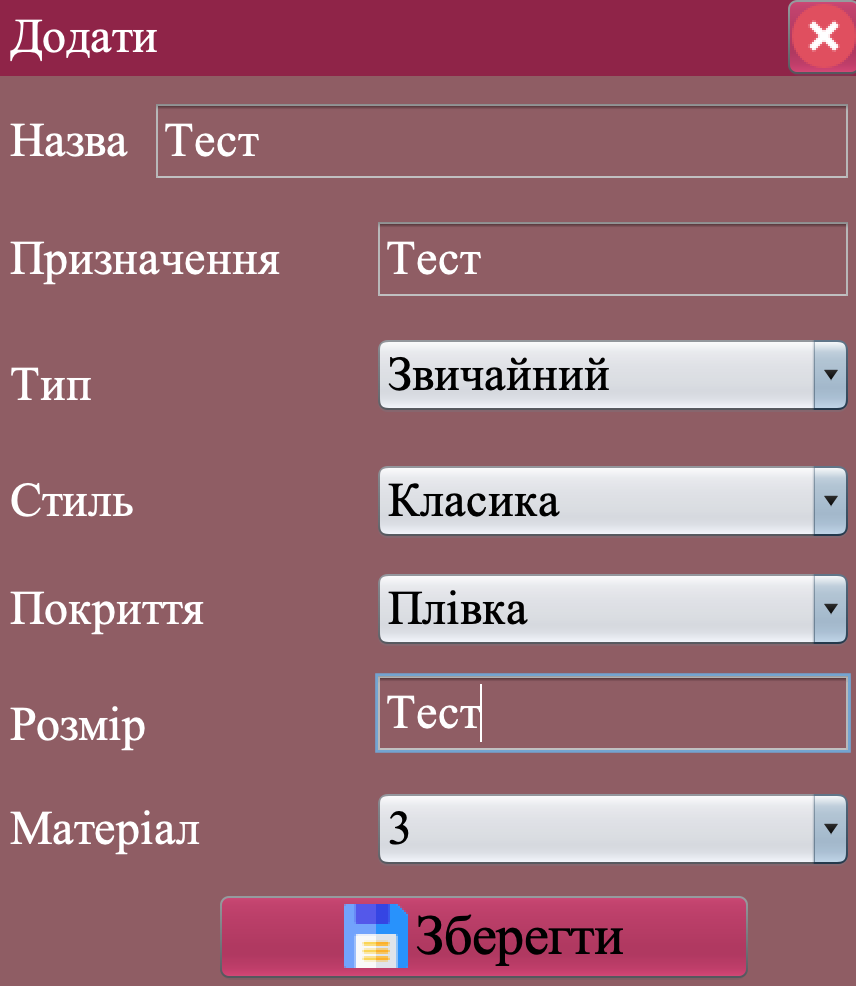


Рисунок 2.14 Додавання інформації

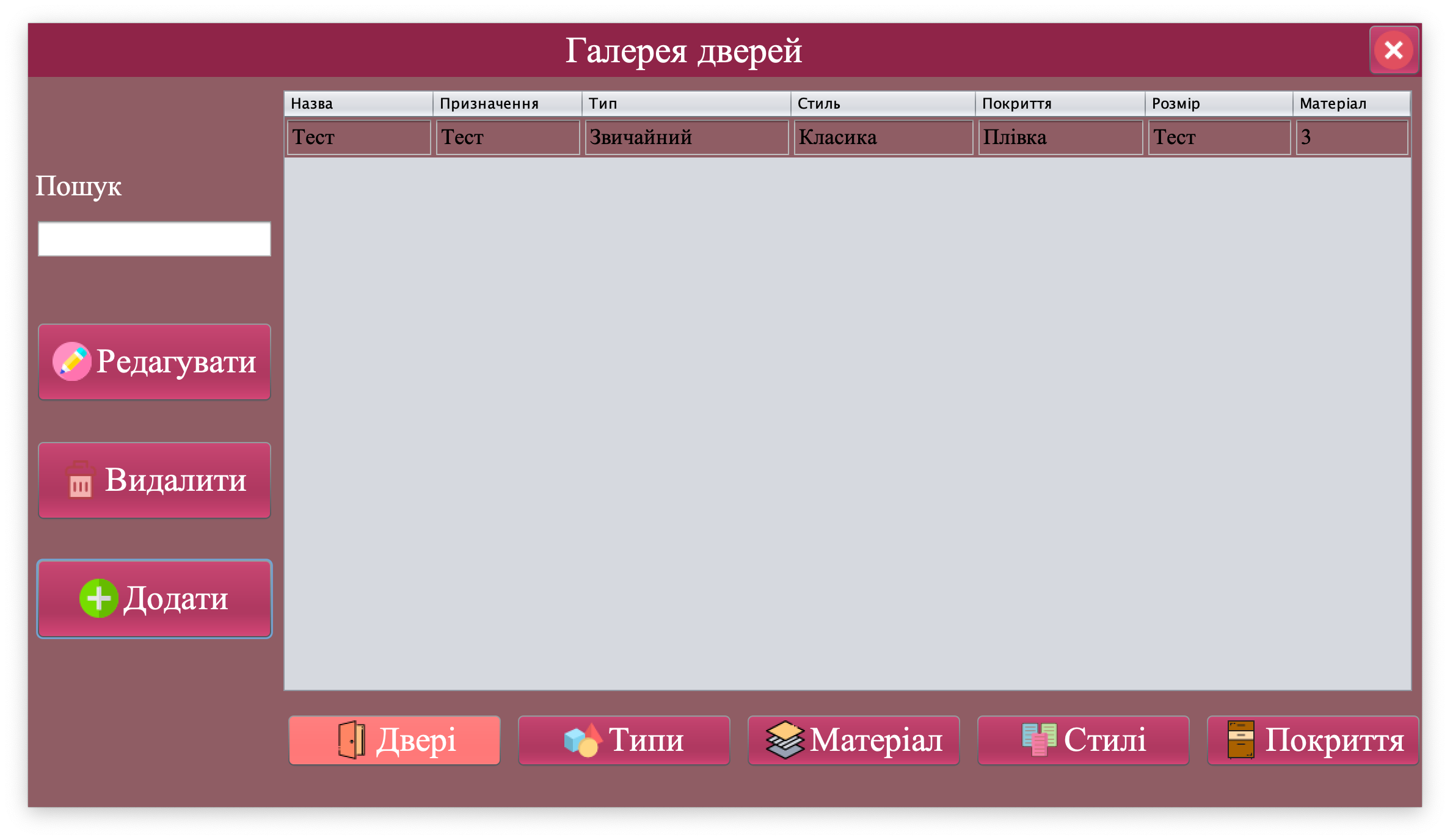


Рисунок 2.15 Головний екран після додавання

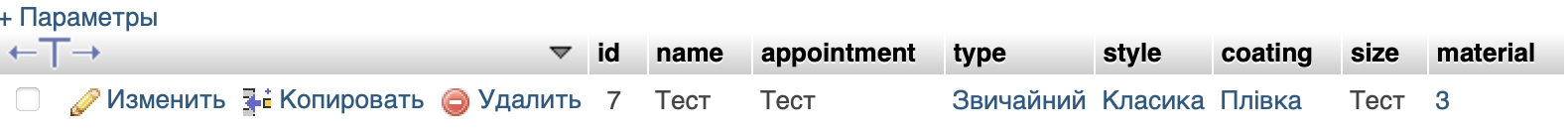


Рисунок 2.16 Вигляд запису в таблиці

* Тест редагування (рисунок 2.18-2.20);



Рисунок 2.17 Редагування файлу

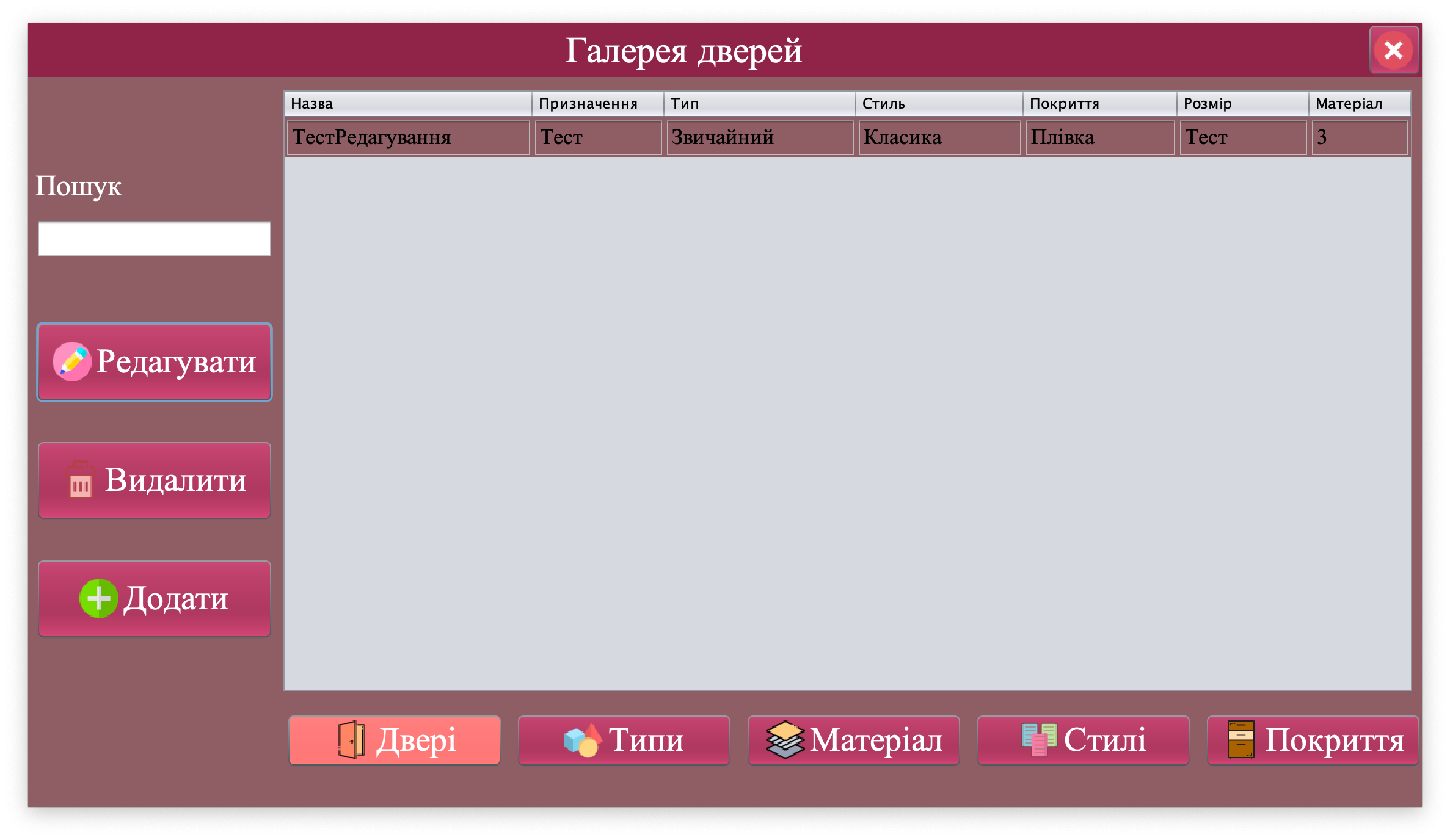


Рисунок 2.18 Редагування файлу

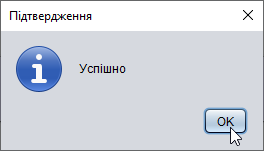


Рисунок 2.19 Підтвердження після редагування

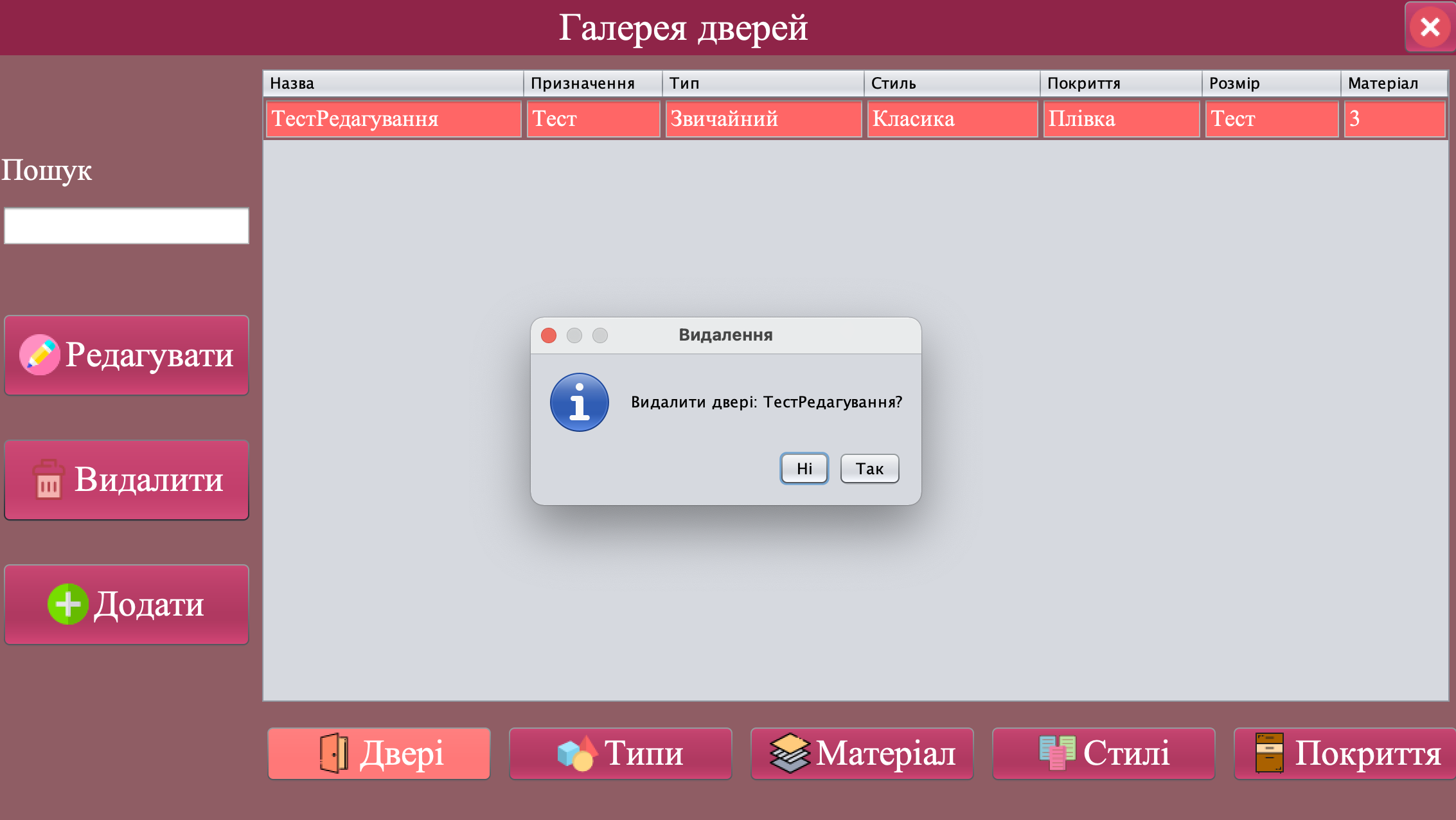


Рисунок 2.20 Видалення інформації

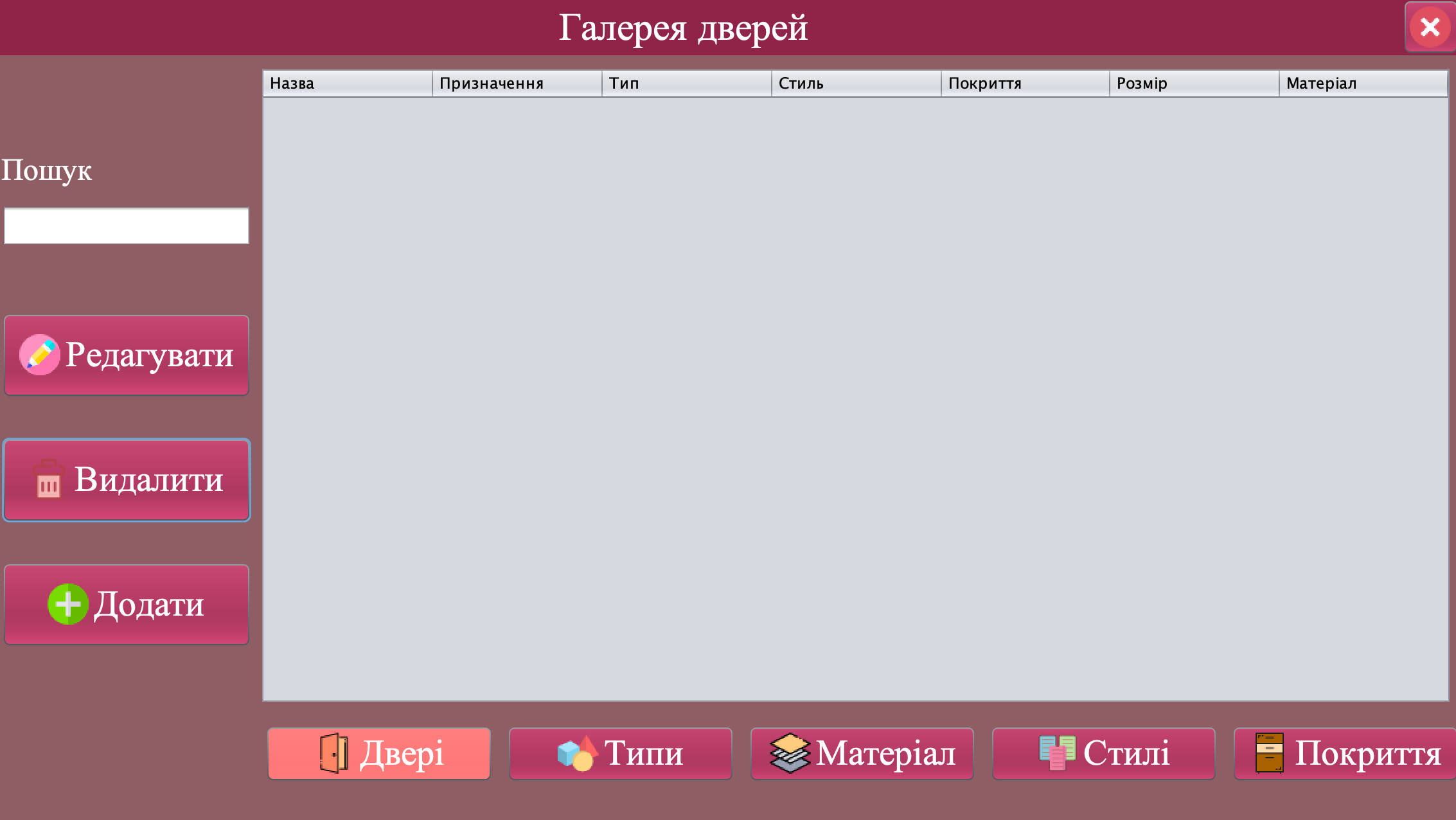


Рисунок 2.21 Головне вікно після видалення

Тестові випадки тестувались на комп’ютеризованому робочому місці зі наступними характеристиками:

* 64-х розрядна OC Windows Embedded 8.1 Industry Pro;
* процесор Intel Xeon L5410 з тактовою частотою 2.33 ГГц;
* 3 ГБ оперативної пам’яті типу DDR3.

## 2.2 Експлуатаційна частина

### 2.2.1 Вимоги до технічних засобів

Для коректної роботи програми на комп’ютері має бути встановлена операційна система сімейства Windows, версії 7 і вище. Вимоги до системи:

* 32, або 64-розрядний процесор з частотою не менш, ніж 1 ГГц;
* 1 ГБ оперативної пам’яті;
* не менш, ніж 400 МБ вільного місця на жорсткому диску;
* встановлена та налаштована платформа Java;
* клавіатура та мишка.

### 2.2.2 Склад інсталяційного пакету та інструкція

Для того щоб користуватися програмою у користувача повинна бути папка «GalleryDoors», в якій знаходяться ресурси для запуску програми, а саме папка dist з файлом «GalleryDoors.jar» (рисунок 2.20).

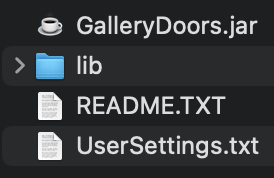


Рисунок 2.22 Вигляд необхідних файлів для запуску програми

Щоб запустити програму потрібно два рази клацнути на файл .jar, після чого на екрані користувача з’явиться форма для старту.

### 2.2.3 Опис інтерфейсу

Під час запуску програми відкривається нова форма для входу в робочу форму.

В головному меню користувач може відразу працювати з базою, створювати, редагувати, видаляти дані.

Багато елементів на формах займають текстові поля та кнопки, тобто користувачу досить вибрати необхідну дію, й програма виконає необхідну операцію.

Також для привабливого інтерфейсу було використано свій креатив.

### 2.2.4 Ієрархічна структура продукту

Навігація по проекту:

Після завантаження додатку користувача очікує вікно входу (рисунок 2.25)

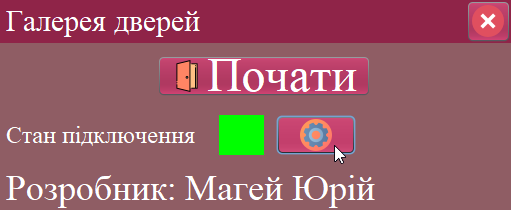


Рисунок 2.23 Вікно входу

Для того, щоб перейти до вікна налаштувань підключення потрібно в низу вибрати об’єкт з вікном налаштувань (Рисунок 2.24).

В даному вікні користувач може змінити дані на дані для підключення до бази даних.



Рисунок 2.24 Вікно налаштувань

Після входу програма переходить у головне меню з основним функціоналом

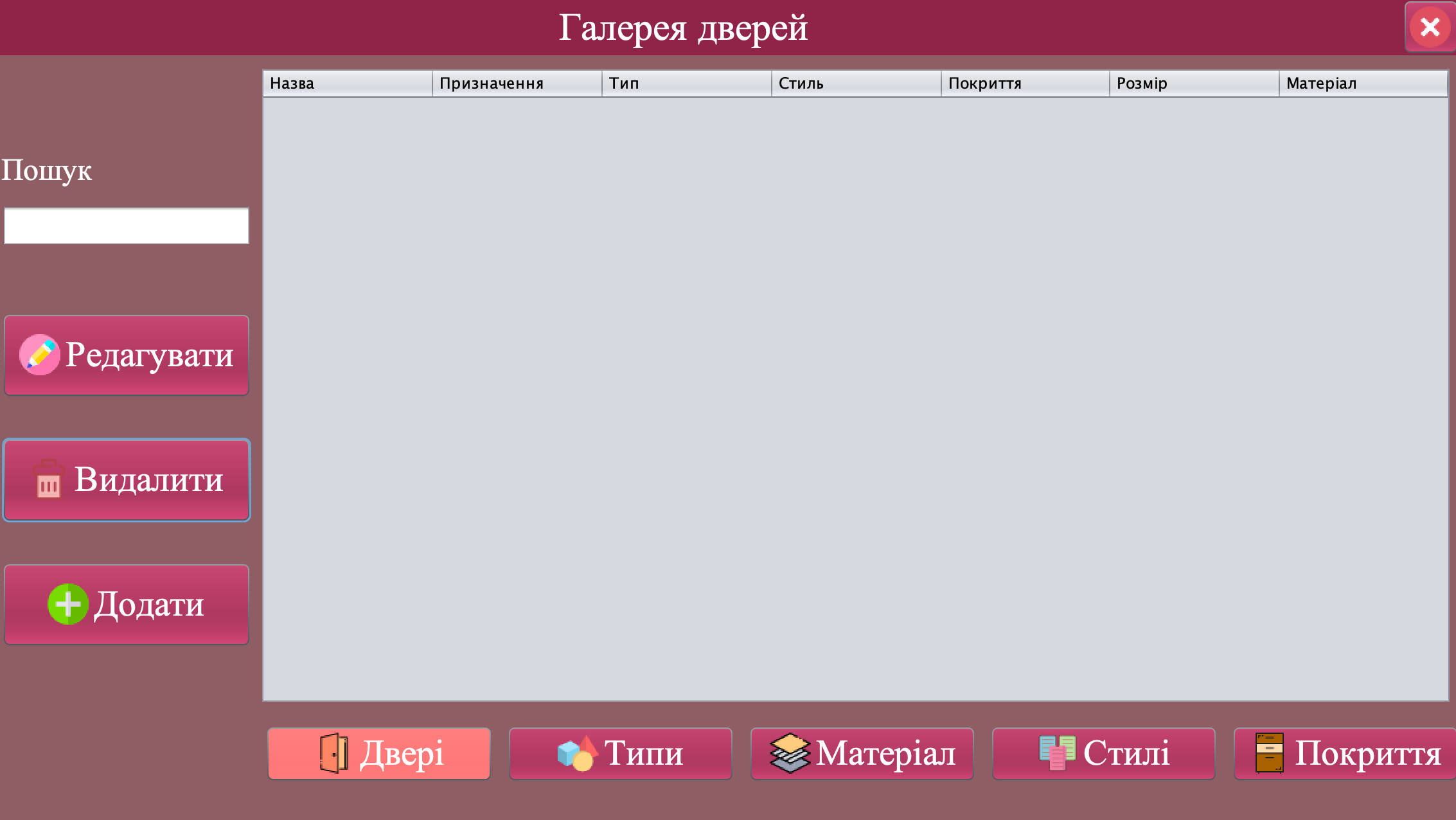


Рисунок 2.25 Головне меню програми

### 2.2.5 Керівництво користувача та приклади використання

### програмного продукту

Програмний продукт створювався з метою використання його по його головній цині. Для керування використовується мишка, там де потрібно вводити текст використовується клавіатура. Програма може бути запущена з різних ОС, такі як Windows або Linux. Щоб запустити програму потрібно два рази натиснути на файл BashEditor.jar. Програма дає такі можливості:

* створити скрипт;
* редагувати редагувати скрипт;
* вихід.

# ВИСНОВОК

В момент розробки курсового проекту було вивчено і охоплено досить велику кількість нової інформації в програмуванні та й в цілому. В ньому було освоєно такі великі теми як: робота з базами даних, покращено навички в дизайні, вдосконалено розуміння алгоритмів, а також покращено роботу з мовою програмування Java, в якій було використано декілька освоєних бібліотек для роботи з даними, різні спрощення читання та збільшення функціональності коду за допомогою лямбда-виразів і найголовніше використання об’єктно-орієнтованого підходу для побудови проекту.

Закріплено роботу з конструкторами класів, а також роботу з самими класами.

У курсовому проекті була використана база даних. Закріплено навички побудови бази даних, таблиць, структури таблиць, індексів, зв’язків і так далі. З програмної частини — це створення підключення до бази даних, виконання запитів на відбір, редагування, створення та видалення даних.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MySQL [Електронний ресурс] /. — Електронні текстові дані — https://uk.wikipedia.org/wiki/MySQL, Режим доступу: вільний.
2. Java [Електронний ресурс] /. — Електронні текстові дані — https://uk.wikipedia.org/wiki/Java, Режим доступу: вільний.
3. JavaSwing[Електронний ресурс] /. — Електронні текстові дані — https://uk.wikipedia.org/wiki/JavaSwing Режим доступу: вільний.
4. JDBC [Електронний ресурс] /. — Електронні текстові дані — https://uk.wikipedia.org/wiki/Java\_Database\_Connectivity, Режим доступу: вільний.
5. SQL [Електронний ресурс] /. — Електронні текстові дані — https://ru.wikipedia.org/wiki/SQL, Режим доступу: вільний.
6. Алан Бьюли — Изучаем SQL
7. MVC [Електронний ресурс] /. — Електронні текстові дані — https://medium.com/@alexdev13/mvc-проще-говоря-3b3318961f61, Режим доступу: вільний.
8. Слідкування за комп’ютером працівника [Електронний ресурс] /. — Електронні текстові дані — https://www.kickidler.com/ru/for-it/methods-of-working/slezhka-za-sotrudnikami.html, Режим доступу: вільний.
9. WhatPulse [Електронний ресурс] /. — Електронні текстові дані — http://www.wincore.ru/programs/1798-whatpulse-polnaya-statistika-ob-ispolzovanii-kompyutera.html, Режим доступу: вільний.
10. Энтони Молинаро "SQL. Сборник рецептов" (2009)