Міністерство освіти і науки України

Економіко-технологічний інститут ім. Роберта Ельворті

Кафедра інформаційних технологій

КУРСОВА РОБОТА

ПОЯСНЮВАЛЬНА ЗАПИСКА

з дисципліни «Об’єктно-орієнтоване програмування»

«Бібліотека класів для розробки

родового (Генеалогічного) дерева»

Керівник:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Комісія:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницький 2024

Економіко-Технологічний інститут

ім. Роберта Ельворті

Кафедра *інформаційних технологій*

Рівень вищої освіти *перший (бакалаврський)*

Дисципліна *Об’єктно-орієнтоване програмування*

Спеціальність *122 Комп’ютерні науки*

Освітня програма: *Комп’ютерні науки*

Курс *4*

Група *КН-20*

Семестр *8*

ЗАВДАННЯ на курсовий проект студента

Васильєвої Мар’яни Вікторівни

1 Тема проекту: Спроектувати та реалізувати бібліотеку класів для розробки родового (Генеалогічного) дерева. Паспортні дані членів деякого родового клану; посилання на дітей (або на батьків). Пошук всіх нащадків або всіх предків для вказаної особи. Ієрархічне відображення генеалогічного дерева обраної людини.

2 Термін здачі студентом закінченого проекту: “16” квітня 2024 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, знімки екрану*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

РЕФЕРАТ

Пояснювальна записка до курсової роботи: 44 с., 10 рис., 3 табл., 2

додатки, 7 джерел.

ЛОКАЛЬНИЙ СЕРВЕР РОЗРОБКИ, MVC, SQL, PHP, СУБД SQLITE, COMPOSER, HTML, CSS, ІНКАПСУЛЯЦІЯ, КЛАС, ОБ'ЄКТ, МЕТОД, ВЛАСТИВІСТЬ, КОНСТРУКТОР, SINGLETON, FACTORY.

Метою роботи є розробка програми «Family Tree» на засадах об'єктно-орієнтованого програмування. Методи розробки базуються на використанні локального середовища розробки PHP, мови програмування PHP та принципів ООП.

В результаті отриманий веб-додаток під назвою «Family Tree», що дозволяє зберігати, опрацьовувати інформацію про осіб та їх взаємозв’язки в контексті створення генеалогічного дерева.

Реалізовано можливості користувацького інтерфейсу:

- додавати, редагувати, видаляти осіб в таблицю “People”;

- додавати, редагувати, видаляти інформаційні записи на головну сторінку.

Реалізовано можливості вебдодатку:

- формувати родове дерево для обраної особи;

- виводити загальну інформацію по всім особам у вигляді таблиці.

Реалізовано додаткові серверні можливості

- консольні команди для демонстрації роботи проєкту.

ЗМІСТ

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

ВСТУП

У сучасному світі інформаціних технологій, зокрема в веб-розробці (програмування веб додатків), особливо значущим стає вміння застосувати об’єктно-орієнтовані принципи для створення масштабованих і легко модифікованих програмних рішень. Дана курсова робота, виконана в рамках дисципліни "Об’єктно-орієнтоване програмування", має на меті не лише закріпити базові поняття і принципи об’єктно-орієнтованого програмування, а й демонструє їх практичне застосування через розробку спеціалізованої бібліотеки класів.

Головна задача даної курсової роботи — спроектувати та реалізувати бібліотеку класів для розробки генеалогічного (родового) дерева, здатну зберігати паспортні дані членів деякого родового клану, включно з посиланнями на дітей та батьків. Окрім того, бібліотека дозволить здійснювати пошук всіх нащадків або всіх предків для вказаної особи та ієрархічно відображати генеалогічне дерево обраної людини.

Реалізація проекту здійснена з використанням мови програмування PHP, шаблонізатора Blade та системи управління базами даних SQLite, що забезпечує гнучкість та зручність управління даними. Застосування цих технологій дозволяє ефективно інтегрувати об’єктно-орієнтовані підходи у веб-розробку, створюючи міцну основу для подальших інноваційних розробок у цій галузі.

Таким чином, дана курсова робота відображає не тільки теоретичне осмислення об’єктно-орієнтованих принципів, а й їх практичне застосування в реалізації конкретного програмного продукту, що має значний потенціал застосування у сфері родових досліджень та управління генеалогічною інформацією.

РОЗДІЛ 1. Специфікація програми

1.1. Визначення робочої середи розробки та обрання інструментів розробки.

Програмна реалізації бібліотеки класів для обраної теми (проєктування та реалізація бібліотеки класів для розробки родового дерева) необхідно чітко визначити робочу середу розробки та обрати відповідні інструменти.

Для розробки бібліотеки класів, призначеної для створення родового дерева, вибрано наступну технологічну стеку:

**Мова програмування:** Обрано PHP через її широкі можливості у веб-розробці та зручність інтеграції з різними базами даних та шаблонізаторами (template engine).

**Інтегроване середовище розробки (IDE)**: Вибрано JetBrains PhpStorm, оскільки це одне з найбільш потужних та зручних середовищ для роботи з PHP. PhpStorm надає розширену підтримку синтаксису PHP, інтеграцію з системами керування версіями, засоби для дебагінгу та рефакторингу коду, що забезпечує високу продуктивність розробки.

**Пакетний менеджер**: Використовуємо Composer для керування залежностями PHP проєкту. Composer дозволяє легко встановлювати, оновлювати та керувати бібліотеками, що використовуються у проєкті, що є важливим для підтримки коду в актуальному стані.

**Система контролю версій**: Для керування версіями коду обрано Git. Git — це найпопулярніша система контролю версій, яка дозволяє розробникам з легкістю відстежувати та управляти змінами в коді проєкту. Використання Git сприяє кращій координації роботи команди, дозволяє здійснювати паралельну розробку, швидко відкатувати зміни та інтегрувати роботу з віддаленими репозиторіями, такими як GitHub або Bitbucket. Інтеграція Git з PhpStorm також підвищує продуктивність роботи, оскільки дозволяє виконувати всі основні операції безпосередньо з IDE.

Це забезпечує високу гнучкість та ефективність у процесі розробки, що є критично важливим для успішного ведення сучасних IT-проектів.

**Локальний сервер розробки (вбудований вебсервер РНР)** - це середовище для розробки вебдодатків на локальному комп'ютері. Локальний сервер дозволяє запускати вебсервер, базу даних, та інші сервіси для розробки та тестування вебдодатків. Локальний сервер дозволяє розробникам працювати над вебдодатками без доступу до Інтернету та забезпечує швидкий доступ до ресурсів.

**База даних**: Використовується SQLite через її простоту інтеграції та використання без необхідності налаштування складних серверів. SQLite ідеально підходить для розробки додатків середнього та малого розміру, де не передбачається великий обсяг даних або високе навантаження.

**Інтерфейс користувача**: Інтерфейс реалізований за допомогою шаблонізатора Blade та фреймворка Bootstrap. Blade надає зручні інструменти для створення динамічного контенту, дозволяючи ефективно управляти HTML шаблонами і сприяє швидкій розробці. Bootstrap використовується для швидкого та адаптивного дизайну веб-сторінок, що дозволяє забезпечити гарний вигляд інтерфейсу на будь-яких пристроях.

Ця комбінація технологій дозволяє ефективно проєктувати та реалізовувати бібліотеку класів для створення родового дерева, забезпечуючи високу продуктивність розробки та зручність подальшої підтримки проєкту.

1.2. Обрання архітектурного шаблону.

Для проєктування та реалізації бібліотеки класів, що спрямована на розробку родового дерева, ключовим є вибір архітектурного шаблону, який визначатиме структуру та методи взаємодії компонентів системи. Враховуючи специфіку проєкту, обрано наступні архітектурні рішення:

**Модель-вид-контролер (MVC)**: MVC є стандартним шаблоном проектування, що розділяє додаток на три основні компоненти: модель (дані), вид (інтерфейс користувача), та контролер (логіка обробки). Цей шаблон сприяє чіткому розділенню логіки та інтерфейсу, що полегшує тестування, розширення та підтримку коду. Для родового дерева, MVC дозволить ефективно управляти даними користувачів, їхніми взаємозв'язками та представленням цієї інформації.

**Singleton**: Патерн Singleton застосовується для забезпечення існування лише одного екземпляра класу в усьому додатку. Це корисно для управління глобальними ресурсами, такими як з'єднання з базою даних. Використання Singleton для SQLite бази даних забезпечить, що з'єднання буде створено один раз та використовуватиметься повсюдно в рамках проєкту.

**Factory**: Патерн Factory використовується для створення об'єктів без специфікації точних класів об'єктів, які будуть створені. Цей підхід ідеально підходить для створення об'єктів класів у нашій бібліотеці, де може бути кілька типів елементів родового дерева, таких як користувачі, зв'язки, події тощо. Factory забезпечить гнучкість у створенні цих об'єктів, дозволяючи легко впроваджувати нові типи.

**Принципи об'єктно-орієнтованого програмування (ООП)**: Основні принципи ООП, такі як інкапсуляція, наслідування, поліморфізм і абстракція, відіграватимуть центральну роль у проектуванні та реалізації нашої бібліотеки. Інкапсуляція забезпечує, що дані класу захищені та сховані від зовнішнього доступу, наслідування дозволяє створювати нові класи на базі вже існуючих, поліморфізм сприяє використанню одного інтерфейсу для різних базових форм даних, а абстракція дозволяє мінімізувати складність, виділяючи ключові характеристики об'єкта.

Використання цих шаблонів та принципів забезпечить розробку гнучкої, масштабованої та легко підтримуваної системи для родового дерева.

РОЗДІЛ 2. Проєктна специфікація

РОЗДІЛ 3. Інструкція користувача

ВИСНОВКИ

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

Додаток А