

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Пояснювальна записка**

Тема: Шаблони проєктування в ООП

Варіант: Записник - телефони, дати, зустрічі, завдання

| Виконав  студент II курсу  Тітов Єгор Павлович КП-01 |  | Погоджено  керівник курсової роботи  Заболотня Тетяна Миколаївна |
| --- | --- | --- |

Київ 2022

## ЗМІСТ

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ** 3

## ВСТУП 4

## СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ 6

* 1. Модульна організація програми 6
  2. Функціональні характеристики 8

## ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ 9

* 1. Обґрунтування вибору та опис шаблонів проектування для

програмної реалізації 9

* 1. Діаграма класів 15
  2. Опис результатів роботи програми 18

## ВИСНОВКИ 21

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

* БД - база даних, що містить дані про елементи програмного додатку.
* ПЗ - програмне забезпечення.

## ВСТУП

Дана курсова робота присвячена розробці програмного забезпечення записника за допомогою використання шаблонів проектування.

В повсякденному житті кожної людини виникає необхідність запам’ятовувати багато різних даних. Як-от дні народження родичів і знайомих, телефони різних людей і служб, дата, час і місце назначених зустрічей тощо. Оскільки це все важко постійно тримати у себе в голові, то було би доцільніше записувати все необхідне в якомусь одному записнику. Комп’ютери мають можливість майже необмежено створювати такі прості записи, при чому легко змінюючи їх чи видаляючи.

Саме тому пропонується розробка програмного додатку записника.

Використання розробленого додатку дозволить здійснювати операції створення/редагування/видалення/дуплікації/пошуку/експорту над різними типами записів.

*Об’єктом* дослідження є процес обліку записів за допомогою програмного додатку.

*Метою роботи* є розроблення програмного забезпечення записника з використанням шаблонів проєктування.

Для досягнення визначеної мети необхідно виконати такі *завдання*:

* абстрагувати об’єкти предметної галузі;
* розробити структурну організацію ПЗ за допомогою застосування основних принципів ООП та шаблонів проектування;
* визначити та описати функціональні характеристики програми;
* обґрунтувати вибір шаблонів проектування, використаних для побудови програми;
* розробити користувацький інтерфейс;
* виконати реалізацію програмного забезпечення відповідно до вимог технічного завдання;
* виконати тестування розробленої програми;
* оформити документацію з курсової роботи.

Розроблене ПЗ записника складається з двох компонентів, що містять модулі: ConsoleApp та ProcessData.

Реалізовані шаблони проектування: Стратегія, Шаблонний метод, Фабричний метод, Прототип, Фасад.

До функціональних можливостей програми належать: створення і обробка записів, їх дуплікація та експорт, надання та отримання запитів з бази даних.

Для функціонування розробленої програми необхідно забезпечити наявність на комп’ютері 20 Мб вільного дискового простору та встановленої додаткової бібліотеки: sqlite.

Розроблене програмне забезпечення може бути використане в повсякденному житті будь-якою людиною.

Пояснювальна записка складається зі вступу, трьох розділів, загальних висновків. Робота містить 31 рисунок. Загальний обсяг роботи – 21 друкованих сторінки.

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМИ**
   1. **Модульна організація програми**

Структура системи побудована по шаблону багаторівневої архітектури та складається з рівнів абстракцій, що є взаємоізольовані. Система складається з трьох рівнів: користувацький інтерфейс, бізнес-логіка (або серверна частина) та база даних.

Шаблон багаторівневої архітектури був обраний, спираючись на потребу в побудові ієрархічної взаємодії рівнів абстракції та простоту реалізації.

Програмні засоби реалізовані у вигляді десктопного додатку з консольним інтерфейсом користувача.

Структурна схема системи зображена на рис.

|  |
| --- |

*Рис 1.1.1. Структурна схема системи*

Проєкт розбитий на такі модулі:

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

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

*Рис 1.1.2. Модулі проєкту*

Діаграма залежностей модулів:

|  |
| --- |

База даних представлена компактною SQLite. У БД зберігаються дані про:

* Записи телефонів
* Записи дат
* Записи зустрічей
* Записи завдань
  1. **Функціональні характеристики**

Система являє собою програмний додаток для керування записами. Вона дозволяє здійснювати операції створення/редагування/видалення/дуплікації/ пошуку/експорту даних.

Щоб запустити екземпляр програмного додатку необхідно створити фасад репозиторіїв передавши шлях до бази даних через конструктор. Далі необхідно створити стратегії вибору категорії та дії над категорією. Після цього користувач має змогу вводити команди і отримувати на них відповіді, передбачена поведінкою програми. У разі помилки про некоректність вводу користувач отримає сповіщення і зможе спробувати ввести команду знову.

Ключові функції системи:

* Створення/Видалення запису
* Редагування запису
* Пошук/Перегляд/Фільтрація записів
* Експорт записів
* Дуплікація записів

1. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ** 
   1. **Обґрунтування вибору та опис шаблонів проектування** **для програмної реалізації записника**
2. **Стратегія (Strategy)**

Стратегія - це поведінковий патерн проєктування, який визначає сімейство схожих алгоритмів і розміщує кожен з них у власному класі. Після цього алгоритми можна змінювати один на інший прямо під час виконання програми.

*Використання стратегії у програмі*:

Програма має два схожих алгоритма вибору для користувача. Перший: вибор категорії записів (дати, телефони, зустрічі, завдання). Другий: вибор дії над категорією (створення, редагування, читання, видалення, фільтрація, дуплікація, пошук, експорт). Кожен вибір був оформлений у окремому класі як відповідна стратегія, і під час виконання програми виконується то одна, то інша стратегія.

|  |
| --- |

*Рис. 2.1.1. Діаграма класів, які входять до шаблону «Стратегія»*

*Учасники шаблону:*

* Клас IChooseStrategy - загальний інтерфейс стратегій вибору.
* Клас ChooseActionStrategy - конкретна стратегія, що реалізує вибір дії, що буде виконана відповідно категорії.
* Клас ChooseCategoryStrategy - конкретна стратегія, що реалізує вибір категорії, над якою будуть виконуватися операції.
* Клас Context - зберігає посилання на об’єкт певної стратегії, працюючи з ним через інтерфейс.

*Обґрунтування використання шаблону:*

* Необхідність винесення алгоритмів вибору у окремі класи для гнучкості їх використання та зручності додавання нових алгоритмів вибору при необхідності.
* Можливість змінювати стратегії вибору під час виконання програми.

1. **Фасад (Facade)**

Фасад - це структурний патерн проєктування, який надає простий інтерфейс до складної системи класів.

*Використання фасаду у програмі*:

Програма має 4 репозиторії для роботи з 4 категоріями нотатків. Для зручного використання 4 репозиторії були об’єднані в одному класі - фасаді, що робить можливим використовувати лише один клас для взаємодії зі всіма даними бази даних.

|  |
| --- |

*Рис. 2.1.2. Діаграма класів, які входять до шаблону «Фасад»*

*Учасники шаблону:*

* Клас RepositoryFacade - надає спільний інтерфейс доступу до всіх репозиторіїв.
* Класи DatesRepository / MeetingsRepository / PhonesRepository / TasksRepository - класи підсистеми, яку використовує фасад.

*Обґрунтування використання шаблону:*

Система має 4 репозиторії, що є досить незручним для передачі і використання їх в різних методах. Шаблон Фасад дозволяє створити єдиний простий інтерфейс для всіх цих репозиторіїв, що значно полегшує доступ і використання бази даних.

1. **Прототип (Prototype)**

Даний шаблон задає види створюваних об’єктів за допомогою екземпляра-прототипу і створює нові об’єкти шляхом копіювання цього прототипу. Це єдиний шаблон з серії Породжуючих, котрий для створення нових об’єктів використовує не явне інстанціювання (ключове слово new), а клонування.

Шаблон проектування «Прототип» було обрано для інтерфейсу INote, та, відповідно, до всіх класів, що реалізують даний інтерфейс.

*Використання прототипу у програмі*:

Шаблон прототип використовується для дуплікації запису в базу даних.

|  |
| --- |

*Рис. 2.1.3. Діаграма класів, які входять до шаблону «Прототип»*

*Учасники шаблону*:

* Інтерфейс INote - описує операцію клонування
* Класи DateNote / PhoneNote / MeetingNote / TaskNote - реалізують операції клонування самого себе

*Обґрунтування використання шаблону:*

Оскільки класи нотаток мають багато приватних полів і неможливо створити спільний інтерфейс, який давав би доступ до всіх полів всіх класів, єдиним варіантом дуплікації буде застосування шаблону прототип, оскільки він дає змогу клонувати об’єкт, не вдаючись у подробиці їхньої реалізації.

1. **Фабричний метод**

Фабричний метод - це породжувальний патерн проєктування, який визначає загальний інтерфейс для створення об’єктів у суперкласі, дозволяючи підкласам змінювати тип створюваних об’єктів.

*Використання фабричного методу у програмі*:

Використовується для створення об’єктів-трансформаторів, що трансформують дані у csv та tsv формат.

|  |
| --- |

*Рис. 2.1.4. Діаграма класів, які входять до шаблону «Фабричний метод»*

*Учасники шаблону*:

* Інтерфейс ITransformer- визначає загальний інтерфейс об’єктів, що може створювати творець та його підкласи.
* Класи CSVTransformer / TSVTransformer - містять код трансформування.
* Клас TransformerCreator - оголошує фабричний метод, який має повертати нові об’єкти трансформерів.
* Класи CSVTransformerCreator / TSVTransformerCreator - по своєму реалізують фабричний метод

*Обґрунтування використання шаблону:*

Фабричний метод дозволяє легко додавати нові типи фабрик і продуктів з однаковими інтерфейсами, що дозволить у майбутньому додавати нові типи об’єктів, без необхідності переписування старого коду.

1. **Шаблонний метод**

Поведінковий шаблон, який визначає функціональність конкретних методів в рамках лише абстрактних сутностей. Визначає основу алгоритму та дозволяє підкласам перевизначити деякі кроки алгоритму, не змінюючи структуру в цілому.

*Використання шаблонного методу у програмі*:

Використовується для реалізації алгоритму експорту різних типів.

|  |
| --- |

*Рис. 2.1.5. Діаграма класів, які входять до шаблону «Шаблонний метод»*

*Учасники шаблону*:

* Клас FileExport - визначає кроки алгоритму і містить шаблонний метод, що складається з викликів цих кроків.
* Клас FileCSVExport - перевизначає деякі кроки алгоритму
* Клас FileTSVExport - перевизначає деякі кроки алгоритму

*Обґрунтування використання шаблону:*

В даному випадку шаблонний метод використовується для експорту даних у файл. Оскільки файли можуть бути різного формату (.csv, .tsv), то необхідно створити різні класи. Але сам алгоритм отримання - трансформування - запису даних є однаковим для всіх класів - експортерів, отже можемо використати шаблонний метод для уникнення повторення коду і стандартизації алгоритму.

* 1. **Діаграма класів**

|  |
| --- |

*Рис. 2.2.1, Повна діаграма*

|  |
| --- |

*Рис. 2.2.2, Перша половина діаграми*

|  |
| --- |

*Рис. 2.2.3, Друга половина діаграми*

* 1. **Опис результатів роботи програми**

| Початок роботи і вибір категорії |
| --- |
|  |

| Вибір дії над категорією |
| --- |
|  |

| Створення запису |
| --- |
|  |

| Редагування запису |
| --- |
|  |

| Читання записів |
| --- |
|  |

| Видалення запису |
| --- |
|  |

| Дуплікація запису |
| --- |
|  |

| Фільтрація записів |
| --- |
|  |

| Пошук записів |
| --- |
|  |

| Експортування записів |
| --- |
|  |
|  |

| Команда допомоги |
| --- |
|  |

| Вихід з програми |
| --- |
|  |

## ВИСНОВКИ

Метою даної курсової роботи було розроблення програмного забезпечення записника з використанням шаблонів проектування. Підставою для розроблення стало завдання на виконання курсової роботи з дисципліни «Програмування» студентами ІІ курсу кафедри ПЗКС НТУУ «КПІ ім. І.Сікорського».

Для досягнення поставленої мети у повному обсязі виконано завдання, визначені у аркуші завдання на курсову роботу; розроблено графічні матеріали; реалізовано всі вимоги до програмного продукту, програмного та апаратного забезпечення, наведені у технічному завданні; створено відповідну документацію.

В процесі виконання роботи було розроблено програмний засіб для записника у вигляді десктопного додатку з консольним інтерфейсом користувача. За допомогою цього додатку користувач має змогу швидко і легко створювати, редагувати, видаляти, шукати, фільтрувати, експортувати записи з таких категорій як: дати, телефони, зустрічі, завдання.

Програму створено на основі використання шаблонів проєктування: зокрема до структури розробленого програмного забезпечення входить реалізація п’яти шаблонів, які належать до різних груп шаблонів проектування (структурних, породжувальних та поведінкових).

Для розроблення програмного забезпечення використано мову програмування C#. Платформи: Windows.

Перспективним напрямком подальшого дослідження даної тематики є розширення функціоналу даного записника шляхом додання підтримки нових способів експорту та нових категорій записів.