



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота №2
із дисципліни «Технології розроблення програмного забезпечення»
Тема: «ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ. СЦЕНАРІЇ
ВАРІАНТІВ ВИКОРИСТАННЯ. ДІАГРАМИ UML. ДІАГРАМИ КЛАСІВ.
КОНЦЕПТУАЛЬНА МОДЕЛЬ СИСТЕМИ»

Виконав:
Студент групи ІА-24
Боднар А. Д.

Перевірив:
Мягкий М.Ю.

Київ-2024

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.
3. Намалюйте діаграму класів для реалізованої частини системи.
4. Виберіть 3 прецеденти і напишіть на їх основі прецеденти.
5. Розробити основні класи і структуру системи баз даних.
6. Класи даних повинні реалізувати шаблон Репозиторію для взаємодії з базою даних.
7. Підготувати звіт про хід виконання лабораторних робіт.

Звіт, що подається повинен містити: діаграму прецедентів, діаграму класів системи, вихідні коди класів системи, а також зображення структури бази даних.

Хід роботи

Теоретичні відомості

Прецеденти (Use Case Diagram)

Діаграма прецедентів слугує для ілюстрації функціональних вимог до системи. Вона демонструє, як користувачі (актори) взаємодіють із системою через певні сценарії використання. Головними складовими діаграми є актори, прецеденти та зв'язки між ними. Прецеденти допомагають визначити основні функції системи та сприяють їх загальному розумінню.

Діаграма класів (Class Diagram)

Діаграма класів призначена для моделювання статичної структури системи. Вона відображає класи, їх атрибути, методи (операції) та зв'язки між класами, такі як асоціації, агрегації, композиції та успадкування. Класи представляють ключові компоненти системи, їхні характеристики та поведінку, а зв'язки показують, як ці класи взаємодіють.

База даних та її структура

База даних є організованим набором даних, збережених у структурованому вигляді, зазвичай у таблицях. Таблиці містять рядки (записи) та стовпці (поля), які містять атрибути даних. Структура бази даних визначає взаємозв'язки між даними. Основні елементи включають таблиці, ключі (первинні та зовнішні) та зв'язки між таблицями (один-до-одного, один-до-багатьох, багато-до-багатьох).

Шаблон Репозиторію (Repository Pattern)

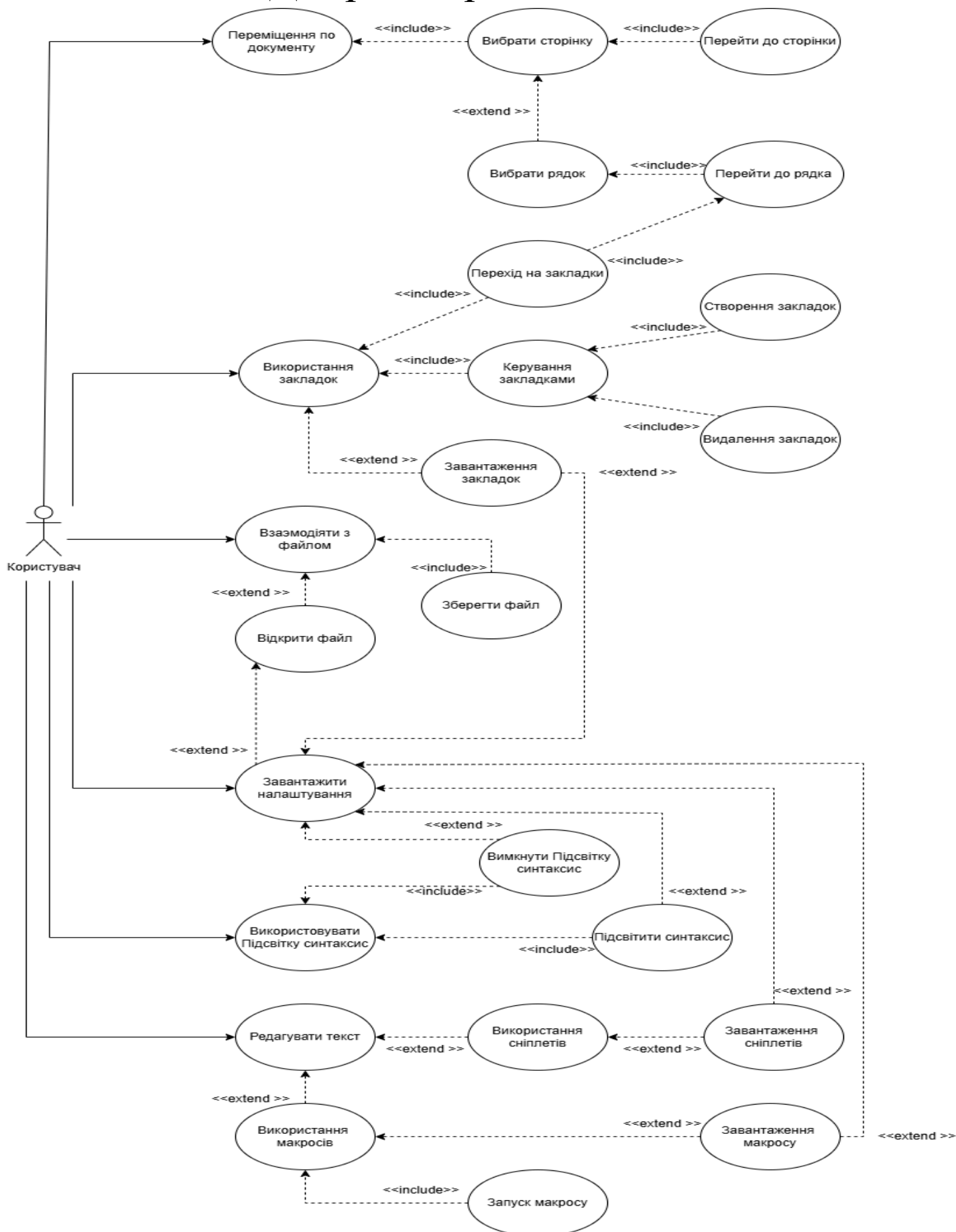
Шаблон Репозиторію застосовується для абстракції доступу до даних. Він дозволяє взаємодіяти з базою даних через клас-репозиторій, що об'єднує всі операції збереження, отримання, оновлення та видалення даних. Це знижує залежність між бізнес-логікою та логікою доступу до даних, роблячи систему більш адаптивною до змін і модернізацій.

Вибір прецедентів та аналіз

Прецеденти допомагають виявити найбільш відповідні сценарії використання для вашої системи. Важливо провести аналіз кількох аналогічних систем, щоб оцінити, як їх функціональність відповідає вимогам вашого проекту. Це сприяє розробці ефективних і зручних рішень для користувачів.

Тема “Текстовий редактор”

Діаграма прецедентів



Прецеденти

Три сценарії використання:

Сценарій 1: Збереження файлу

Передумови:

Користувач відкрив або створив документ для редагування.

Післяслів'я:

Файл успішно збережено в обраному місці на диску. У разі помилки стан документа не змінюється.

Взаємодіючі сторони:

Користувач, текстовий редактор.

Короткий опис:

Цей варіант використання визначає збереження змін до файлу після редагування.

Основний потік подій:

1. Користувач натискає "Зберегти файл" або використовує відповідну комбінацію клавіш.
2. Система запитує місце збереження файлу (якщо це перше збереження).
3. Користувач вибирає шлях для збереження файлу та натискає "ОК".
4. Система зберігає файл в обраному місці, повідомляючи про успішне завершення операції.

Винятки:**Виняток №1:**

Недостатньо місця на диску. Якщо під час збереження виявляється, що на диску недостатньо місця, система виводить повідомлення про помилку. Користувач може або вибрати інше місце для збереження, або звільнити місце на диску.

Примітки:

Можливість автоматичного збереження документа також може бути реалізована.

Сценарій 2: Використання закладок

Передумови:

У документі існують закладки, і користувач хоче швидко перейти до певної частини документа.

Післяслів'я:

Користувач успішно перейшов до розділу документа, позначеного закладкою.

Взаємодіючі сторони:

Користувач, текстовий редактор.

Короткий опис:

Цей варіант використання описує процес переходу користувача до закладки в документі.

Основний потік подій:

1. Користувач відкриває меню закладок або використовує відповідну комбінацію клавіш.
2. Система відображає список створених закладок у документі.

3. Користувач вибирає потрібну закладку зі списку.
4. Система автоматично переміщує користувача до відповідного місця в документі.

Винятки:

Виняток №1:

Закладка відсутня. Якщо користувач намагається вибрати закладку, яка була видалена або зміщена через редагування документа, система виводить повідомлення про помилку. Користувач може створити нову закладку або вибрати іншу.

Примітки:

Закладки можуть бути автоматично оновлювані або прив'язуватися до певних елементів тексту.

Сценарій 3: Підсвітка синтаксису

Передумови:

Користувач працює з програмним кодом або спеціалізованими файлами, де потрібна підсвітка синтаксису.

Післяслів'я:

Синтаксис коду підсвічено відповідно до обраного мови програмування.

Взаємодіючі сторони:

Користувач, текстовий редактор.

Короткий опис:

Цей варіант використання описує процес ввімкнення підсвітки синтаксису для більш зручної роботи з кодом.

Основний потік подій:

1. Користувач відкриває файл з кодом або іншим структурованим текстом.
2. Система автоматично розпізнає тип файлу або користувач вибирає мову програмування вручну.
3. Система активує підсвітку синтаксису, виділяючи різні елементи коду різними кольорами.

Винятки:

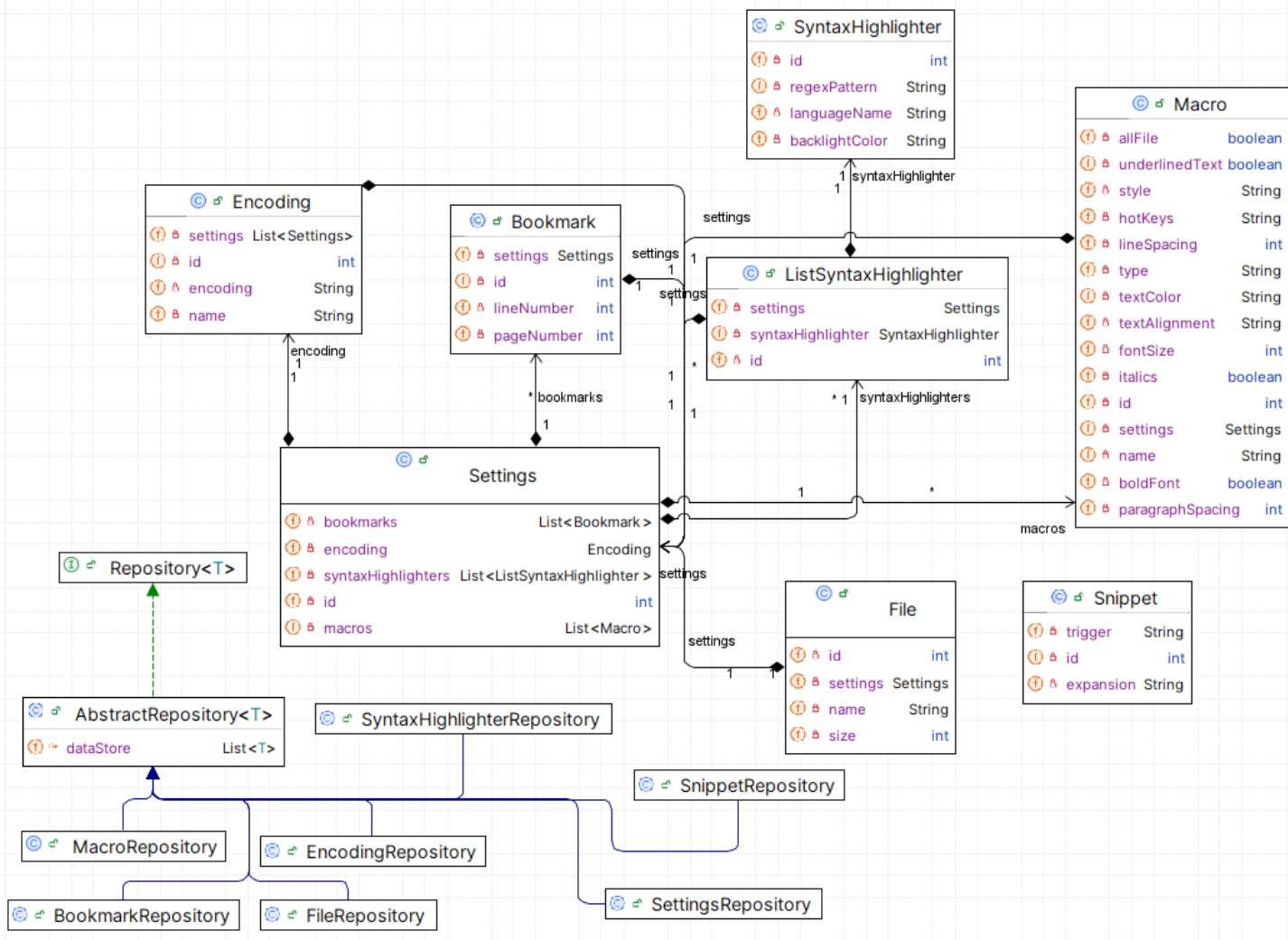
Виняток №1:

Невідомий формат файлу. Якщо система не може розпізнати тип файлу, вона виводить повідомлення про відсутність підсвітки. Користувач може вибрати мову вручну або продовжити без підсвітки.

Примітки:

Можна додати можливість ручного ввімкнення/вимкнення підсвітки синтаксису для різних форматів файлів.

Діаграма класів



File

- Клас представляє файл у системі. Він має посилання на налаштування, які використовуються для роботи з файлом, а також містить інформацію про шлях до файлу та його розмір.

Settings

- Клас зберігає налаштування для роботи з файлом, такі як кодування та стиль форматування тексту. Має зв'язок із таблицею ListSyntaxHighlighter для визначення підсвіток синтаксису, які застосовуються до цього набору налаштувань. Один Settings може бути пов'язаний з багатьма файлами.

ListSyntaxHighlighter

- Клас для забезпечення відношення М між Settings і SyntaxHighlighter. Зберігає пари id_Settings і id_SyntaxHighlighter, які визначають, які підсвітки синтаксису використовуються у конкретному налаштуванні.

SyntaxHighlighter

- Клас, який представляє окрему підсвітку синтаксису. Містить інформацію про мову, регулярні вирази для підсвітки та колір підсвітки. Один SyntaxHighlighter може бути використаний у кількох Settings через ListSyntaxHighlighter.

Bookmark

- Клас представляє закладку в документі. Зберігає номер сторінки та рядка, на який вказує закладка, а також посилання на налаштування, до яких ця закладка належить.

Snippet

- Клас для представлення сніплету. Кожен сніплет має тригер і текст розширення. Сніпети є загальними для всіх проектів і не пов'язані з конкретним налаштуванням або файлом.

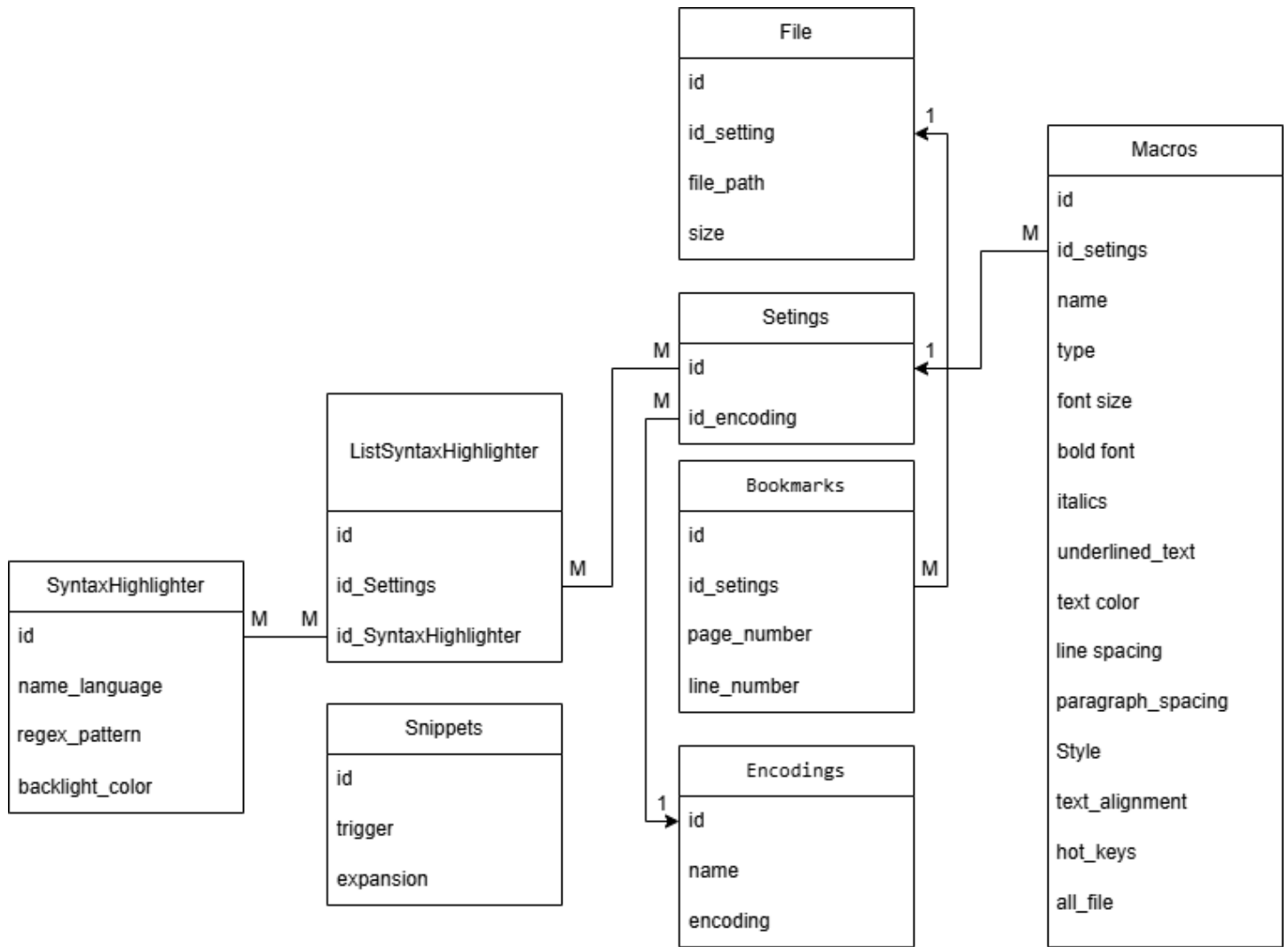
Macro

- Клас представляє макрос, який містить назву, тип, форматування тексту (шрифт, розмір, колір тощо), вирівнювання та гарячі клавіші. Макроси прив'язані до налаштувань, з якими можуть використовуватися.

Encoding

- Клас зберігає інформацію про кодування тексту. Один об'єкт Encoding може використовуватися в багатьох Settings, дозволяючи застосовувати єдине кодування в кількох налаштуваннях.

Структура бази даних



Класи та репозиторії можна знайти в директорії за таким шляхом:
TRPZ_BodnarA_IA24/TRPZ-TextEditor/src

Висновок: У цій лабораторній роботі було розроблено різні діаграми та структури для текстового редактора з розширеними функціями, такими як макроси, сніплетти, закладки, підсвічування синтаксису та налаштування.

- 1. Аналіз теми та схема прецедентів:** Проаналізовано основні функціональні можливості текстового редактора і побудовано схему прецедентів для демонстрації основних сценаріїв використання системи.

2. **Діаграма класів:** Розроблено діаграму класів, що відображає ключові компоненти системи, зокрема класи для зберігання інформації про файли, макроси, сніплети, закладки, а також взаємозв'язки між ними. Ці класи забезпечують підтримку різноманітних функцій редактора.
3. **Прецеденти:** Було вибрано три ключових сценарії, що відображають важливі функціональні процеси редактора, такі як створення закладок, підсвічування синтаксису та використання макросів.
4. **Структура бази даних:** Спроектовано структуру бази даних для зберігання інформації про файли, макроси, сніплети, закладки та підсвічування синтаксису. Кожна таблиця отримала відповідний клас у системі, а для взаємодії з базою даних було реалізовано шаблон Репозиторію.