ЗАСОБИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗА РАХУНОК ВПРОВАДЖЕННЯ ЗАЛЕЖНОСТЕЙ

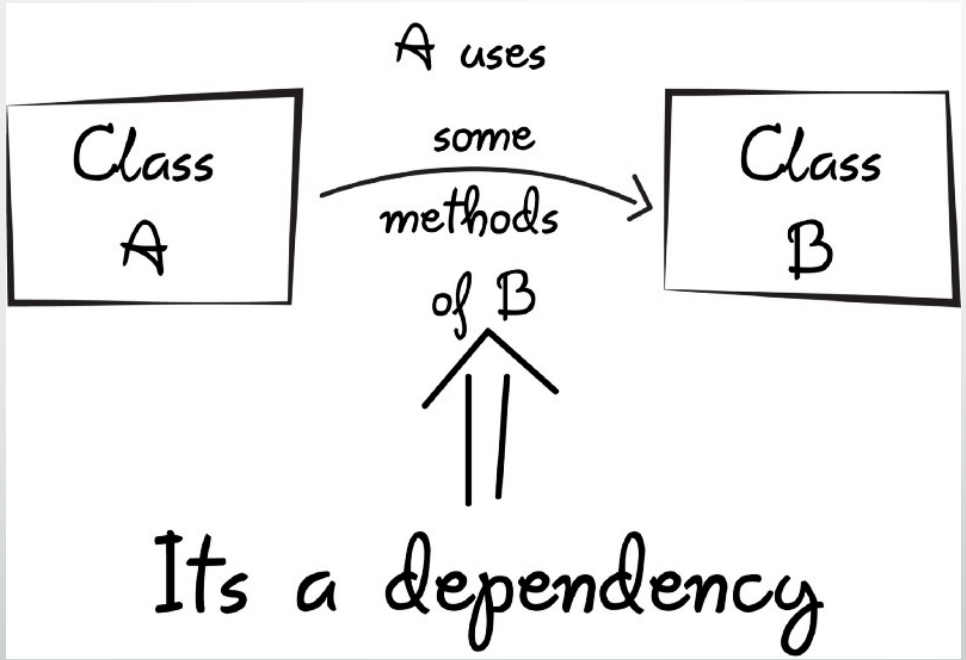
Автор: Руслан Зимовець

БП-4, ІПЗ

Вступ

- У цій роботі розглянуто поняття Впровадження Залежностей та засоби, якими його можна досягти. Досліджено Контейнери Впровадження Залежностей як засіб Впровадження Залежностей та їх користь для мови C++. Також було розроблено бібліотеку мовою C++, що містить Динамічний Контейнер Впровадження Залежностей та засоби для гнучкого конфігурування такого контейнера. Як приклад використання Контейнерів Впровадження Залежностей було створено застосунок Задачник, який демонструє переваги та недоліки таких контейнерів.

Впровадження Залежностей



Контейнери Впровадження Залежностей

- **Контейнер Впровадження Залежностей** – це програмна сутність, відповідальність якої полягає в автоматизації Впровадження Залежностей шляхом інкапсуляції коректного створення об'єктів, враховуючи Впроваджені Залежності їх класів, та надання гнучкого інтерфейсу для керування таким створенням або наданням доступу до створених об'єктів.

Конфігурація Контейнера Впровадження Залежностей

- Налаштування на основі конфігураційних файлів – спосіб конфігурування, у якому вся інформація про налаштування Контейнера Впровадження Залежностей зберігається в одному або декількох файлів.
- Конфігурація на основі коду – спосіб конфігурування, у якому інформація про налаштування Контейнера Впровадження Залежностей цілком зберігається у вигляді коду програми, яка працює на основі цього контейнера.

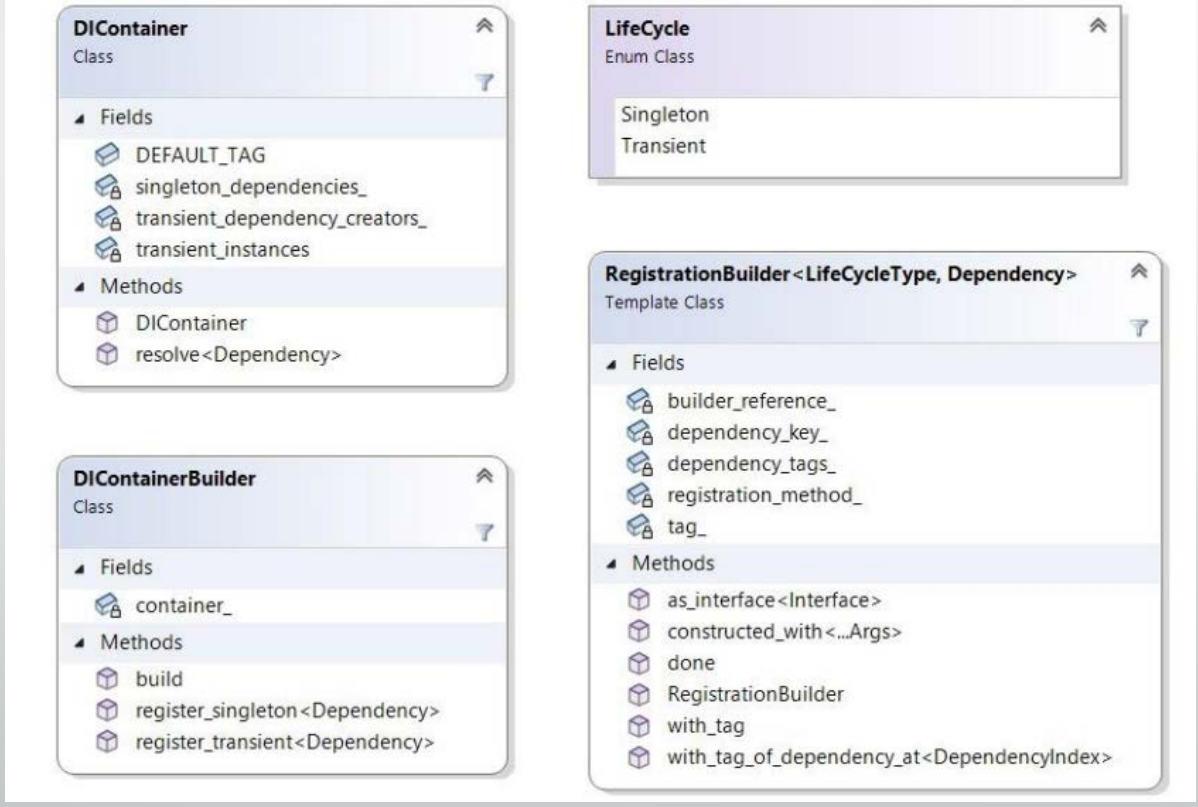
Задача розробки Динамічного Контейнера Впровадження Залежностей

- В межах результатів даної роботи лежить розробка Динамічного Контейнера Впровадження Залежностей мовою C++, який задовольняє наступні умови:
- Можливість автоматизовано реалізувати Впровадження Залежностей
- Можливість досягнення пізнього зв'язування не є обов'язковою
- Можливість здійснювати конфігурацію на основі коду
- Присутній набір інструментів для покращення паралельної розробку
- Можливість впровадження різних реалізацій одного інтерфейсу залежно від конфігурацій

Типи життєвих циклів об'єктів

- **Єдиносутній Життєвий Цикл (Singleton Lifecycle)** – тип життєвого циклу об'єкта, який існує в єдиному екземплярі протягом всього виконання програми або починаючи з моменту, коли контейнер отримав запит на створення цього об'єкту, та до кінця виконання програми.
- **Тимчасовий Життєвий Цикл (Transient Lifecycle)** – тип життєвого циклу об'єкта, який наново створюється кожного разу, коли контейнер отримав запит на створення цього об'єкта, та існує доти, запитувач використовує його.

Архітектура розробленого Динамічного Контейнера Впровадження Залежностей



Висновки

- У роботі було досліджено взірець проектування Впровадження залежностей, його критерії (підтримуваність коду та пізнє зв'язування), переваги та недоліки, а також Контейнери Впровадження Залежностей як засоби його реалізації; було описано та порівняно способи конфігурування таких контейнерів: налаштування на основі конфігураційних файлів та конфігурація на основі коду. Було визначено задачу розробки Динамічного Контейнера Впровадження Залежностей, яка полягала у розробленні бібліотеки fast_di, що містить Динамічний Контейнер Впровадження Залежностей та засоби для простих налаштувань та Глобальних Конфігурацій такого контейнера.

Дякую за увагу!