



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

Лабораторна робота №3
З дисципліни «Технології розроблення програмного забезпечення»
Тема: **«Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та
послідовностей.»**
Flex Automatical tool

Виконав:
Студент групи ІА-22
Сидорін Д.О.

Перевірив:
М'який М. Ю.

Київ-2024

Зміст

Тема:.....	3
Мета:	3
Завдання:.....	3
Хід роботи	3
1. Діаграма розгортання.....	3
2. Діаграма компонентів	5
3. Діаграма послідовностей.....	6
Висновки:	7

Тема:

Діаграма розгортання. Діаграма компонентів. Діаграма взаємодій та послідовностей.

Мета:

Засвоїти основні типи діаграм, які використовуються для моделювання програмних систем, зокрема діаграми розгортання, компонентів, взаємодій і послідовностей. Навчитися будувати й аналізувати такі діаграми, а також застосовувати їх для опису архітектури та процесів у програмних системах.

Завдання:

Візуальний додаток для складання "карт пам'яті" з можливістю роботи з декількома картами (у вкладках), автоматичного промальовування ліній, додавання вкладених файлів, картинок, відеофайлів (попередній перегляд); можливість додавання значків категорій / терміновості, обведення областей карти (поділ пунктирною лінією).

Хід роботи

1. Діаграма розгортання



Рис. 1 — Діаграма розгортання

Ця діаграма розгортання демонструє архітектуру програмного забезпечення для інструменту автоматизації, який складається з трьох основних компонентів: клієнтського додатку, сервісів для зовнішніх інтеграцій і планувальника завдань.

1. Клієнтський додаток (Client Application):

- Використовує середовище виконання для настільного додатку.

- Це головний інтерфейс для користувача, що дозволяє налаштовувати автоматизацію завдань, вводити правила для автоматичного скачування контенту, зміни статусів в комунікаторах та інші дії.
- Здійснює взаємодію з користувачем для налаштування макросів, планування завдань і створення правил для автоматизації.
- Використовує API для зв'язку із зовнішніми сервісами та планувальником завдань.

2. Зовнішні сервіси (External Services):

- Використовує інтерфейси для інтеграції з іншими системами, такими як API для Skype, торрент-трекери або системи для скачування контенту.
- Забезпечує автоматичне виконання завдань, таких як скачування фільмів, книг, зміна статусу в комунікаторах або передача даних на сервери.
- Взаємодіє з клієнтським додатком через API для отримання запитів та виконання відповідних дій.

3. Планувальник завдань (Task Scheduler):

- Здійснює обробку та виконання завдань, що були попередньо налаштовані користувачем.
- Проводить запуск заданих дій на основі часу, наприклад, завантаження нових файлів о 5 ранку або зміна статусу в Skype при відсутності активності.
- Взаємодіє з клієнтським додатком для отримання налаштувань та виконання дій на запланованих інтервалах.

Ця архітектура побудована з метою забезпечення масштабованості, надійності та ефективності автоматизації завдань, з розподілом відповідальностей між компонентами, що дозволяє легко додавати нові функції та інтеграції без впливу на основну систему.

2. Діаграма компонентів

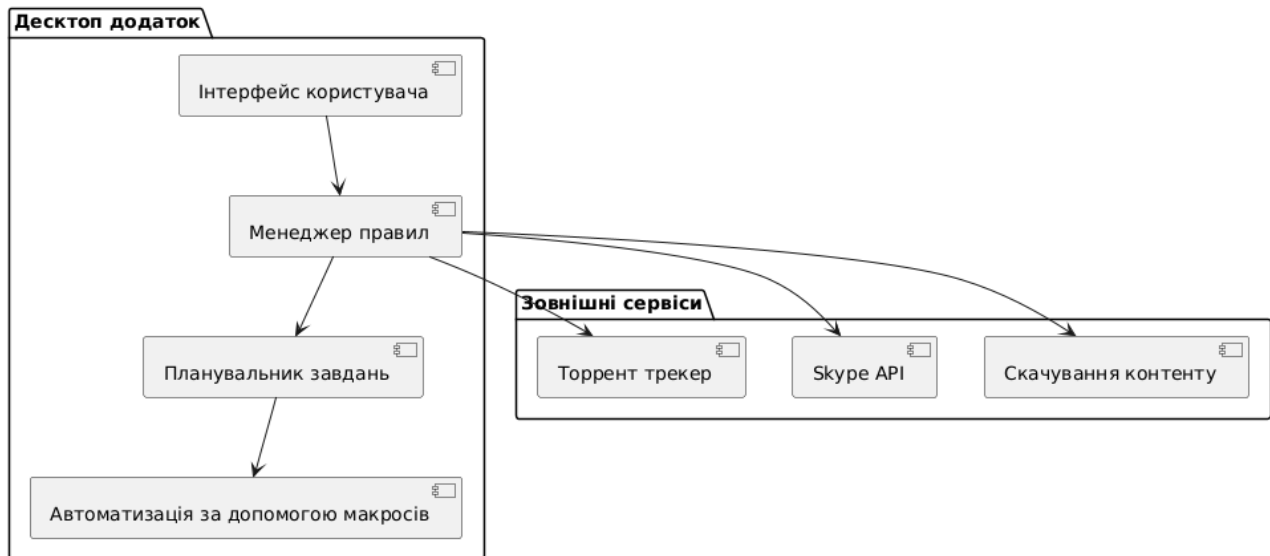


Рис. 2 — Діаграма компонентів

Ця діаграма компонентів описує архітектуру програмного забезпечення для інструменту автоматизації, що забезпечує виконання автоматичних дій за правилами користувача. Архітектура поділяється на три основні компоненти:

1. Клієнтська частина (Client Application)

- **Інтерфейс користувача:** Головний компонент, який надає користувачеві можливість взаємодіяти з програмою. Тут користувач може налаштовувати правила автоматизації, записувати макроси, планувати завдання, а також переглядати та редагувати існуючі налаштування.
- **Менеджер правил:** Компонент, відповідальний за зберігання та обробку правил автоматизації, таких як скачування контенту або зміна статусів у комунікаторах. Він взаємодіє з іншими частинами системи для виконання вказаних дій.
- **Планувальник завдань:** Компонент для керування часом та виконанням автоматизованих дій на основі заданого часу або події. Наприклад, планує завантаження фільмів у певний час або змінює статус у Skype при відсутності активності.

2. Зовнішні сервіси (External Services)

- **API для Skype:** Компонент, що забезпечує інтеграцію з Skype для автоматичної зміни статусу користувача, наприклад, при тривалій відсутності активності.
- **Система скачування контенту:** Відповідає за взаємодію з веб-сайтами або торрент-сервісами для автоматичного завантаження нових фільмів, книг або іншого контенту за вказаними правилами.

- **Торрент трекер:** Сервіс для автоматичного завантаження файлів через торрент, що дозволяє додаткам отримувати нові файли, зокрема для автоматичних роздач.

3. Модуль автоматизації (Automation Module)

- **Макроси:** Відповідає за виконання записаних макросів, таких як натискання клавіш або дії миші. Цей модуль допомагає автоматизувати рутинні дії за допомогою запису послідовності команд.
- **Інтерфейс API для зовнішніх сервісів:** Зв'язує клієнтську частину з зовнішніми сервісами для виконання запитів та отримання відповідей, що стосуються автоматизації завдань, таких як завантаження контенту чи зміна статусу.

Взаємодія компонентів:

- Клієнтська частина взаємодіє з зовнішніми сервісами через API для виконання таких дій, як завантаження контенту або зміна статусів у Skype.
- Планувальник завдань координує виконання автоматичних дій на основі заданого часу, використовуючи інтерфейс клієнтської частини та макроси.
- Модуль автоматизації записує та відтворює макроси для виконання конкретних дій, таких як натискання клавіш або запуск певних процесів.

3. Діаграма послідовностей



Рис. 3 — Діаграма послідовностей

Ця діаграма послідовностей демонструє основні взаємодії користувача з додатком для автоматизації завдань, включаючи налаштування правил, виконання автоматичних дій, планування завдань та взаємодію з зовнішніми сервісами.

1. Налаштування правила автоматизації:

- Користувач ініціює процес налаштування нового правила (наприклад, для скачування контенту або зміни статусу).

- Інтерфейс користувача передає запит до менеджера правил, який зберігає це правило.
- Менеджер правил викликає API для взаємодії з планувальником завдань, що дозволяє встановити час виконання завдання або тригер для його активації.
- Після збереження даних, інтерфейс користувача оновлюється, відображаючи підтвердження про успішне налаштування.

2. Запуск автоматичного завдання:

- Користувач або система ініціює виконання запланованого завдання.
- Планувальник завдань перевіряє наявність активних завдань і запускає відповідну дію (наприклад, завантаження фільмів).
- Планувальник завдань звертається до зовнішнього сервісу через API для виконання необхідної дії (наприклад, скачування контенту).
- Зовнішній сервіс повертає результат (наприклад, файл чи підтвердження), після чого планувальник завдань завершує процес.

3. Взаємодія з комунікатором (наприклад, Skype):

- Користувач налаштовує правило для зміни статусу в комунікаторі (наприклад, Skype).
- Менеджер правил передає запит до API для взаємодії з комунікатором.
- API комунікатора змінює статус користувача відповідно до заданого правила (наприклад, встановлення статусу "Не турбувати" після певного часу бездіяльності).
- Після виконання дії, інтерфейс користувача отримує підтвердження про зміну статусу.

4. Збереження змін в налаштуваннях:

- Користувач ініціює збереження змін в налаштуваннях автоматизації (наприклад, зміна правил або додавання нового завдання).
- Інтерфейс користувача передає запит до менеджера правил, який виконує збереження налаштувань.
- Менеджер правил звертається до планувальника завдань і зовнішніх сервісів для оновлення даних про виконувані дії.
- Після успішного збереження налаштувань, інтерфейс користувача отримує підтвердження про успішне оновлення.

Ця діаграма показує ефективну взаємодію компонентів для забезпечення автоматизації завдань, планування їх виконання та інтеграції з зовнішніми сервісами для виконання конкретних дій, таких як скачування контенту або зміна статусів.

Висновки:

У даній роботі було створено діаграму послідовностей, яка ілюструє основні процеси взаємодії користувача з додатком, діаграму розгортання, що описує фізичну архітектуру системи, та діаграму компонентів, яка демонструє модульну структуру додатку і взаємозв'язки між його частинами. Ці діаграми забезпечують візуалізацію архітектури та логіки роботи додатку для подальшої ефективної розробки та підтримки.