

ПРОГРАМНА БІБЛІОТЕКА ДЛЯ ВИКОНАННЯ CLI КОМАНД

Багато розробників використовують мову програмування Java для написання великих додатків, що надають програмні рішення, які забезпечують можливість імплементувати бізнес логіку у код, який легко читати та підтримувати. Існують вимоги, що потребують реалізації можливості виконання команд для того, щоб взаємодіяти з операційною системою і отримати певну корисну інформацію, що може бути використана програмою. Мова програмування Java має вбудований функціонал “ProcessBuilder” для запуску зовнішніх процесів.

“ProcessBuilder” дає можливість запустити CLI команди використовуючі низькорівневий підхід. Це означає, що розробнику необхідно написати багато коду для того, щоб виконати прості речі.

Такий підхід має певні недоліки:

Витрачається багато часу на написання коду і це може викликати деякі проблеми коли завдання треба виконати швидко.

Важко реалізувати складну логіку в коді, яка потребує багатопоточність та запуск команд у певному порядку на основі зворотніх викликів.

Читабельність логіки ускладнюється великою кількістю коду і це порушує продуктивність програміста.

Підтримка коду ускладнюється пропорційно кількості написаного коду, тому згодом цей недолік може викликати дуже негативні наслідки замовнику.

Існує велика ймовірність написання коду, що буде повторюватися в багатьох місцях і це потенційно викликає проблему створення помилок, що будуть створюватися при редагуванні існуючого програмного коду.

Проаналізувавши вище перелічені недоліки можна зробити висновок, що існує потреба у створенні програмної бібліотеки, яка має високорівневу архітектуру використання та дозволяє програмісту реалізувати вимоги швидко і просто. На даний час аналогів подібної бібліотеки не існує, тому більшість компаній пишуть самостійно програмний код, що виконує команди. Таке рішення викликає ряд проблем та недоліків, що вже були перелічені. Деякі компанії розробляють внутрішні бібліотеки для того щоб спростувати процес написання коду для виконання команд, але такі реалізації мають залежність від проектного коду, що не є ефективним підходом для розробки бібліотек загального використання. Існують такі основні вимоги для написання даної бібліотеки загального використання:

- Мінімальна кількість залежностей від зовнішніх бібліотек.
- Легковагість забезпечує можливість економії оперативної пам’яті та процесорного ресурсу.
- Масштабування дає здатність додавання нового функціоналу в бібліотеку швидко та просто.
- Декларативний інтерфейс програмного забезпечення виражає виконання логіки без надання детальної інформації як це працює під капотом.

Бібліотека повинна підтримувати багатопоточність для того, щоб мати можливість вирішувати завдання оптимізації алгоритмів, а також функціонал використання паралельного програмування повинен бути простим для розуміння та прозорим для використання в коді.

Отже, можна зробити наступні висновки:

1. «ProcessBuilder» інтерфейс програмного забезпечення має багато недоліків у використанні, які переростають у справжні проблеми під час реалізації вимог продукту.

2. Існує необхідність у розробці програмної бібліотеки для реалізації можливості написання програмного коду у декларативному стилі для виконання CLI команд, щоб програміст міг легко імплементувати вимоги замовника та побудувати код, що легко підтримується та читається.

3. Бібліотека повинна бути спроектована таким чином, щоб її використання було можливим для будь якого додатку.

Список літератури

1. Огляд “ProcessBuilder” інтерфейсу для виконання команд операційної системи [Електронний ресурс] – Режим доступу до ресурсу: <https://www.baeldung.com/java-lang-processbuilder-api>.

Базові концепції “ProcessBuilder” інтерфейсу [Електронний ресурс] – Режим доступу до ресурсу: <https://bhuvanagoli.gitbooks.io/basic-concepts-in-java-se-8/content/javalang-package/memory-management-process-builder.html>.