

Після безпосередньої розробки продукту слід проаналізувати його якість. Тобто те, наскільки чітко він дозволяє задовольнити потреби користувача. Лише після того, як програма стане корисною користувачу, тобто матиме попит, можна говорити про повне завершення розробки та алгоритмізації.

Отож, розкриваючи суть питання можна відмітити, що розробка та алгоритмізація програмного продукту це багатофункціональний процес, у якому беруть участь спеціалісти з різних галузей, кожен з яких виконує свою частину роботи для подальшого успіху проєкту та можливості його покращення в майбутньому.

Список літератури

1. Дегтярьова Л.М., Гроза П.М., Сомов С.В. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». Полтава: ПолтНТУ, 2017. 218 с.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. К.: ВПЦ "Київський Університет", 2021. 200 с.
3. Трофименко О.Г., Козін О.Б., Задерейко О.В., Плачінда О.Є. Веб-технології та веб-дизайн: навч. посібник. Одеса: Фенікс, 2019. 284 с.

УДК 004.43

*Солодун Т. Р., студентка 3 курсу
Ніколюк П.К., д. ф.-м.н., професор,
професор кафедри інформаційних
технологій*

ОСОБЛИВОСТІ ВИКОРИСТАННЯ MIDDLEWARE В МОВІ ПРОГРАМУВАННЯ JAVA

Донецький національний університет імені Василя Стуса, м. Вінниця

Анотація. У даному дослідженні подана інформація про поняття проміжного програмного забезпечення. Методологічною основою роботи є розгляд транзакцій у мові програмування Java. Специфіка досліджуваної теми передбачає розгляд даного поняття на прикладі використання у реалізації робочих проєктів.

Ключові слова: middleware, java, JDBC.

Для початку варто ознайомитись з основними поняттями для кращого розуміння даної теми. Отже, Middleware – це програмний рівень, розташований між додатками та операційними системами. Проміжне програмне забезпечення зазвичай використовується в розподілених системах, чим спрощує розробку програмного забезпечення [1].

Проміжне програмне забезпечення отримало свою назву тому, що перший час зазвичай виконувало роль посередника між інтерфейсом програми або клієнтом і серверним ресурсом - наприклад, базою даних, спеціалізованим апаратним пристроєм, з якого клієнт міг запитувати дані. Простий приклад роботи middleware зображено на Рисунку 1:

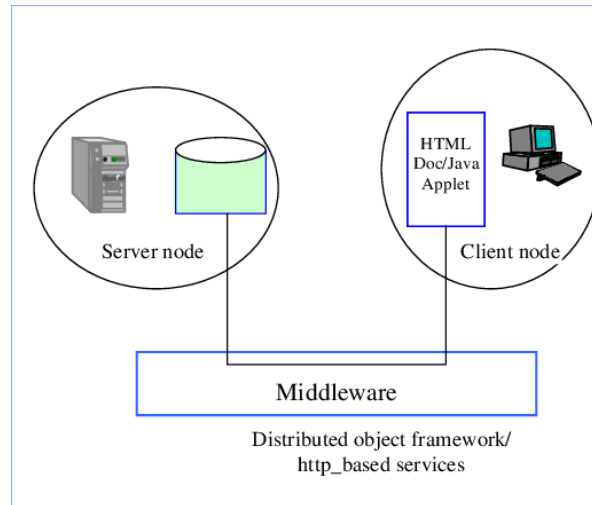


Рисунок 1

Однак, сучасне проміжне програмне забезпечення працює далеко за межами цієї сфери. Проміжне програмне забезпечення бази даних, наприклад, включає власне сховище даних. Також варто зазначити про введення нового класу проміжного програмного забезпечення, що використовує контейнерну технологію, щоб допомогти розробникам підключатися до ресурсів, розподілених у кількох хмарах [2].

Саме сервіси middleware представляють програмам різноманітні функції API, які, у порівнянні з функціями операційних систем та мережевих служб, забезпечують:

- прозорий доступ до інших мережевих сервісів та програм;
- незалежність з інших мережевих сервісів;
- високу надійність та постійну готовність.

Слід зазначити, що найчастіше розбіжності у функціональності операційної системи та проміжного ПЗ є умовними. Зокрема, деякі можливості, що раніше представлялися виключно засобами middleware, тепер реалізуються на рівні ядра операційних систем. Типовим прикладом є стек TCP/IP, підтримка якого включена майже у всі ОС.

Найшвидше зростаючою категорією проміжного програмного забезпечення Java сьогодні є сервери додатків. Однак важливо реалізувати широкий спектр серверів додатків (та інших видів проміжного програмного забезпечення), які існують.

Також існує багато типів технологій проміжного програмного забезпечення, реалізованих у Java: JDBC, RMI, бібліотеки HTTP, JMS/MQSeries/MQTT, Jini/JavaSpaces, тощо

Основне проміжне ПЗ доступу до БД являє собою зовнішні засоби middleware, спеціально розроблені для звернення до баз даних. Сюди відносяться як технологічні рішення (наприклад, ODBC та JDBC), так і концептуальні, що становлять узагальнену архітектуру інформаційної системи з розподіленими БД (наприклад, EDA або DQB).

Одним з цікавих прикладів є JDBC (Java DataBase Connectivity) - прикладний програмний інтерфейс для звернення до баз даних Java-додатків. Засоби JDBC є платформонезалежними і виконуються в будь-якому середовищі під керуванням віртуальної машини Java. У порівнянні з ODBC цей тип проміжного програмного забезпечення баз даних пропонує більше можливостей з інтеграції додатків і розширює сферу використання БД (наприклад, у бік мобільних додатків). Більше того, специфікація JDBC передбачає можливість доступу Java додатків до БД через ODBC за допомогою спеціального засобу, т.зв. "моста" (ODBC/JDBC-bridge).[3]

Загалом існує 5 типів драйверів, що використовуються в JDBC, найефективнішим з них є тип 3 — це єдиний драйвер JDBC, який використовується для доступу до сервера проміжного програмного забезпечення, який, у свою чергу, здійснює відповідні виклики до бази даних.

Його перевагами є:

1. Цей драйвер базується на сервері, тому на клієнтських машинах не потрібна бібліотека бази даних постачальника.
2. Цей драйвер повністю написаний на Java і, отже, портативний. Він підходить для Інтернету.
3. Є багато можливостей для оптимізації портативності, продуктивності та масштабованості.
4. Мережевий протокол може бути розроблений так, щоб зробити клієнтський драйвер JDBC дуже малим і швидким для завантаження.
5. Драйвер типу 3 зазвичай підтримує такі функції, як кешування (з'єднання, результати запитів тощо), балансування навантаження та розширене адміністрування системи, наприклад ведення журналів та аудит.
6. Цей дуже гнучкий драйвер дозволяє отримати доступ до кількох баз даних за допомогою одного драйвера.

А недоліком лише — для встановлення та обслуговування потрібен інший серверний додаток. Обхід набору записів може зайняти більше часу, оскільки дані надходять з бекенду через сервер.

Хорошим прикладом драйвера JDBC типу 3 є драйвер JDBC DataDirect SequeLink. Схематичне зображення типу три зображено на Рисунку 2[4].

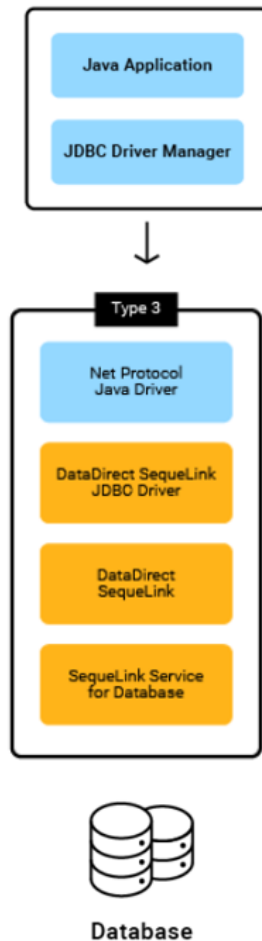


Рисунок 2

Список використаної літератури

1. <https://uk.theastrologypage.com/middleware>
2. <https://www.ibm.com/cloud/learn/middleware#toc-how-middle-J4WLHBej>
3. <https://www.4stud.info/networking/lecture6.html>
4. <https://www.progress.com/faqs/datadirect-jdbc-faqs/what-are-the-types-of-jdbc-drivers>

УДК 004.4

Степанюк О. С., студентка 4
курсу спеціальності 122
«Комп'ютерні науки»
Потапова Н. А., к.е.н., доцент,
доцент кафедри інформаційних
технологій

ПРОЕКТ ДОДАТКУ ДЛЯ ВІДСЛІДКУВАННЯ ФІЗИЧНОЇ АКТИВНОСТІ ДЛЯ ПЛАТФОРМИ IOS

Донецький національний університет імені Василя Стуса, м. Вінниця

Сьогодні мобільні пристрої відіграють одну із ключових ролей у нашому житті. Вони допомагають нам у вирішенні буденних задач, забезпечують вільний