

3. Керівництво з програмування на C# [Електронний ресурс]
Режим доступу: <https://msdn.microsoft.com>
4. Апаратне та програмне забезпечення ПК [Електронний ресурс]
- Режим доступу:
http://eprints.zu.edu.ua/18/1/Konspekt_modul_1_Windows.pdf.
5. Створення додатків Windows Forms у Visual Studio з C #
[Електронний ресурс] - Режим доступу: <https://docs.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2019>.

СКІДАН В.В., ДЕМКІВСЬКА Т.І.

АНАЛІЗ АРХІТЕКТУРНИХ СТИЛІВ ПРИ РОЗРОБЦІ WEB- ДОДАТКІВ

SKIDAN V.V., DEMKIVSKA T.I

ANALYSIS OF ARCHITECTURAL STYLES IN THE DEVELOPMENT OF WEB - APPLICATIONS

The article describes the main architectural styles or templates used in the development of software, in particular in the web applications. Such architectural styles as: component architecture, multi-level architecture, client/server architecture, N-tier/3-tier architecture, service-oriented architecture (SOA), object-oriented architecture, message bus are described in the article. The article presents the main characteristics of the architectural styles, the scope of their application and examples of type of application where their use is recommended.

Keywords: software architecture, architecture of web applications, architectural styles, architectural templates, component architecture, multi-level architecture, client-server architecture, service-oriented architecture, object-oriented architecture.

Вступ

В наш час збільшуються вимоги до проектування програмного забезпечення інформаційних систем, головною складовою якого є створення архітектури програмного застосунку. Правильно підібрана архітектура збільшує гнучкість системи, полегшує її модифікацію та супровід при неминучій зміні вимог в процесі її функціонування.

Основна частина

Архітектурний стиль визначає сукупність компонентів та з'єднань, які можуть використовуватися в екземплярах цього стилю, разом із набором обмежень на їх комбінування [1]. Вони можуть містити топологічні обмеження архітектурних рішень (наприклад, відсутність циклів). Інші обмеження, наприклад, пов'язані з необхідністю обробляти семантику виконання. До основних архітектурних стилів відносяться наступні стилі [2, 3].

Компонентна архітектура описує підхід до проектування і розробки систем з використанням методів проектування програмного забезпечення. У цьому підході поділ дизайну на окремі логічні або функціональні компоненти, що відносяться до чітко визначених інтерфейсів, що містять

методи властивості та події, є особливо важливим і створює більш високий рівень абстракції, при порівнянні з об'єктно-орієнтованим стилем, також не концентрується увага на питаннях загального стану або протоколи зв'язку. До переваг можна віднести: простоту розробки, простоту розгортання, спрощення системи з технічного погляду, меншу вартість розробки та обслуговування, можливість повторного використання.

Багатошарова архітектура забезпечує угруповання зв'язаної функціональності застосування в різних шарах, що вибудовуються вертикально, поверх один одного. Загальні засади: інкапсуляція, абстракція, можливість повторного використання шару, висока пов'язаність усередині шару, слабка зв'язаність між шарами, чіткий поділ функціональності. Приклади таких програм: системи бухгалтерського обліку, веб-додатки або веб-сайти, додатки, які використовують централізовані сервери програм для бізнес-логіки.

N-рівнева/3-рівнева архітектура описує розподіл функціональності на сегменти, як і в багатошаровій архітектурі, але в даному випадку ці сегменти можуть розташовуватися на різних комп'ютерах, їх називають рівнями. Як правило, для зв'язку використовуються методи платформи, а не повідомлень. До характеристик N-рівневої архітектури ПЗ можна віднести: сервісні компоненти та їх розподілене розгортання (забезпечує масштабованість та доступність), ефективність та керованість використання ресурсів функціональну декомпозицію програми.

Незалежність рівнів від інших, крім тих, із якими він безпосередньо стикається. N-му рівню необхідно знати, як обробити запит від $n+1$ рівня, як передати цей запит на $n-1$ рівень, і як обробити результати запиту. Зв'язок між рівнями, як правило, асинхронний, для кращої масштабованості. До переваг можна віднести: масштабованість, гнучкість, доступність, зручність підтримки. Наприклад: Веб-додаток з високими вимогами безпеки.

Клієнт-серверна архітектура дає опис розподіленим системам, що складаються з окремих сервера та клієнта та мережі, що їх з'єднує. Прикладами програми можуть бути: веб-додаток, що виконується у внутрішніх мережах компанії або Інтернет, настільні програми, що працюють з віддаленими ресурсами. Основні переваги: простота обслуговування, велика безпека, централізований доступ до даних, до мінусів можна віднести складність розширюваності, масштабування та залежність від центрального сервера [4].

Заснована на шині повідомлень архітектура описує варіант використання програмної системи, в якій надсилаються та приймаються повідомлення по одному або декільком каналам зв'язку, дозволяючи взаємодіяти додаткам без детальних знань один про одного. Взаємодія реалізується шляхом передачі повідомлень через шину, зазвичай,

асинхронної. Типове використання маршрутизатора повідомлень або шаблону Публікація/Підписка (Publish/Subscribe) та системи обміну повідомленнями, такі як Черга повідомлень (Message Queuing). Шина дозволяє забезпечити обробку: засновану на повідомленнях взаємодії; складної логіки обробки; зміну логіки обробки; інтеграцію із різними інфраструктурами. Основні переваги: гнучкість, розширюваність, слабке зв'язування, масштабованість, простота додатків, невисока складність.

Сервісно-орієнтована архітектура (COA, SOA) дозволяє створювати додатки, що використовують програмні послуги, і дозволяє надавати функціональність програмного забезпечення у вигляді набору сервісів. Сервіси слабо пов'язані завдяки використанню заснованих на стандартах інтерфейсів, які можна викликати, опублікувати та виявити.

COA дозволяє запакувати бізнес-процеси в сервіси, що підтримують можливість взаємодії та використання різних форматів даних та протоколів. Переваги: абстракція; можливість виявлення та автоматичного підключення через інтерфейс, узгодження предметних областей, раціоналізація, можливість взаємодії.

Об'єктно-орієнтована архітектура – це парадигма проектування, заснована на розділенні ПЗ або окремої системи на самостійні, придатні для повторного використання, об'єкти, кожен з яких містить дані і поведінку, що відносяться до цього об'єкту. При об'єктно-орієнтованому проектуванні система розглядається не як набір підпрограм і процедурних команд, а як набори взаємодіючих об'єктів. Об'єкти відособлені, незалежні і слабо зв'язані; обмін даними між ними відбувається через інтерфейси шляхом виклику методів і властивостей інших об'єктів і відправки/прийому повідомлень. Основні переваги: можливість повторного використання за допомогою поліморфізму або абстракції, зрозумілість, покращена тестованість, розширюваність, висока зв'язність.

Висновки

Проаналізувавши переваги та недоліки різних архітектурних стилів було зроблено висновок, що для розробки мобільного застосунку роботи куратора академічної групи є доцільним обрання клієнт-серверної архітектури. Основні переваги архітектурного стилю клієнт-сервер:

1. Безпека. Всі дані зберігаються на сервері, який зазвичай забезпечує більший контроль безпеки, чим клієнтські комп'ютери.

2. Централізований доступ до даних. Оскільки дані зберігаються тільки на сервері, адміністрування доступу до даних набагато простіше, ніж в інших архітектурах.

3. Простота обслуговування. Ролі і відповідальність обчислювальної системи розподілені між декількома серверами, що взаємодіють один з одним по мережі.

Література

1. Рад Б. Я. Архітектура інформаційних систем / Б. Я. Рад, А. І. Водяхо, В. А. Дубенецький, В. В. Цехановський. – М.: Академія, 2012. 220с
2. Рой філдінг Архітектурні стилі та дизайн мережових архітектур програмного забезпечення [Електронний ресурс]. – Режим доступу: <https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>.
3. Харченко О. Г. Експерта система проектування архітектури програмного забезпечення / О. Г. Харченко, І. О. Боднарчук, В. В. Яцишин // Комп'ютерні технології друкарства. – № 29. – 2013. – с. 10-26.
4. Клієнт-серверна архітектура [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура.

РЕЗАНОВА В. Г., ОПАЛЕНИК В.В.

ДОСЛІДЖЕННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЗНАЧЕННЯ ВПЛИВУ ДОБАВОК НА ВЛАСТИВОСТІ МІКРОВОЛОКОН

REZANOVA V. G., OPALENYK V.V.

RESEARCH AND DEVELOPMENT OF SOFTWARE TO DETERMINE THE INFLUENCE OF ADDITIVES ON THE PROPERTIES OF MICROFIBERS

The active introduction of information technology in research is an important priority in the most economically developed countries. As a result, it changes global development trends in the direction of significantly expanding the capabilities of a wide range of industries: pharmacology, pharmaceuticals, chemistry, aeronautics and astronautics, construction, energy, defense, aviation, transport and more.

The use of melts of polymer blends is a leading global trend in the field of chemistry and technology of polymers, which also allows not only to combine the properties of two polymers in one product, but also provides unique effects. The phenomenon of fiber formation allows the production of ultra-thin synthetic fibers with unique properties.

To establish the mechanisms of action of compatibilizer additives on the properties of melts of mixtures, it was of interest to investigate their influence on the patterns of flow of melts of the initial components. The research results show different effects of individual additives and their binary mixtures on the characteristics of melts. Mixtures of compatibilizers are more effective than individual substances.

Given that research is currently conducted mainly experimentally, and the results are stored on paper, the development of software that automates these processes is relevant.

Вступ

Активне впровадження інформаційних технологій у наукові дослідження має важливий пріоритет у найбільш економічно розвинених країнах. Це в результаті змінює світові тенденції розвитку в напрямку значного розширення можливостей широкого кола галузей економіки: