

## ОСОБЛИВОСТІ ПРОЕКТУВАННЯ АРХІТЕКТУРИ КОРПОРАТИВНОГО ДОДАТКУ НА ПРИКЛАДІ СИСТЕМИ ОБЛІКУ РОБІТ ПІДПРИЄМСТВА

*В роботі проведено опис критеріїв, яким повинно задовольняти програмне забезпечення обліку робіт аграрного підприємства. Здійснено визначення предметних підобластей та взаємозв'язків між ними, на основі яких розроблено архітектуру програмного забезпечення обліку робіт аграрного підприємства. Використовуючи спроектовану архітектуру, було розроблено програмне забезпечення обліку робіт аграрного підприємства.*

*Ключові слова:* архітектура програмного додатку, модульність, аграрне підприємство.

O.V. IVANOV, A.O. NICHEPORUK, R.YE. BEL'FER, B.O. SAVENKO

Khmelnitskyi National University

### DESIGN FEATURES AN ARCHITECTURE OF ENTERPRISE APPLICATION BASED ON THE EXAMPLE OF THE ACCOUNTING WORKS ENTERPRISES

*The paper presents a description of the criteria which must be met by software registration of work of the agrarian enterprise. Implemented definition of the subject sub areas and relationships between them on the basis of which developed a software architecture included in the work of the agrarian enterprise. Using the designed architecture has been developed accounting software of work of the agrarian enterprise.*

*Keywords:* software application architecture, modularity, agricultural company.

**Вступ.** Проектування архітектури програмного забезпечення є важливим етапом розробки для проектів зі складною внутрішньою структурою і численними внутрішніми зв'язками. Помилка у виборі способу розв'язання конкретного модулю може сильно вплинути на подальший розвиток проекту, привести до лавиноподібного зростання складнощів та помилок. При невірному або несвоєчасному проектуванні архітектури програмного забезпечення можливий момент, коли стає економічно та ресурсоефективно заново переробити програмне забезпечення, ніж виконувати пошук помилок та здійснювати перепроєктування розроблених компонентів. Тому, проектування архітектури є важливим етапом розробки програмного забезпечення, виконання якого дозволяє підвищити загальну ефективність проектування програмних додатків.

**Критерії, яким повинно задовольняти програмне забезпечення обліку робіт аграрного підприємства.** З метою забезпечення гнучкості при написанні, тестуванні та відлагодженні, виокремимо набір критеріїв, яким повинно задовольняти програмне забезпечення обліку робіт аграрного підприємства [1–5]:

**Ефективність системи.** В першу чергу, програмне забезпечення повинно вирішувати поставлені завдання та якісно виконувати закладені функції, враховуючи різні умови використання. До цього критерію можна віднести такі характеристики як надійність, безпеку, продуктивність, здатність справлятися зі збільшенням навантаження (масштабованість) і т.д.

**Гнучкість системи.** Будь-який програмний додаток доводиться змінювати з часом – змінювати вимоги, додавати нові. Чим швидше і зручніше можна внести зміни в існуючий функціонал та чим менше проблем і помилок це викличе – тим гнучкіше і конкурентоздатною виявиться система. При виконанні проектування програмних систем слід передбачати ситуації при якій, наприклад, архітектурне рішення виявиться невірним, і яку кількість коду при цьому необхідно переписати.

**Можливість розширення системи.** Можливість додавати в систему нові сутності та функції, не порушуючи її основної структури. На початковому етапі в систему має сенс закладати лише основний і самий необхідний функціонал (принцип YAGNI) Але при цьому архітектура повинна дозволяти легко нарощувати додатковий функціонал в міру необхідності. Причому так, щоб внесення найбільш ймовірних змін вимагало найменших зусиль. Це означає, що додаток слід проектувати так, щоб зміна його поведінки і додавання нових функцій було б досягнуто за допомогою написання нового коду (розширення), і при цьому не доводилося б змінювати вже існуючий код. В такому випадку поява нових вимог не спричинить за собою модифікацію існуючої логіки, а зможе бути реалізовано перш за все за рахунок її розширення.

**Масштабованість процесу розробки.** Можливість скоротити термін розробки за рахунок додавання до проекту нових розробників. Архітектура повинна дозволяти розпаралелювання процесу розробки, так щоб багато розробників могли працювати над програмою одночасно.

**Тестопридатність.** Код, який легше тестувати, буде містити менше помилок і надійніше працювати. Але тести не тільки покращують якість коду. Багато розробники приходять до висновку, що вимога «гарна тестованість» є також спрямовуючою силою, що автоматично веде до хорошого дизайну, і одночасно одним з найважливіших критеріїв, що дозволяють оцінити його якість.

**Супровід.** Над програмою, як правило, працює багато людей - одні йдуть, приходять нові. Після написання супроводжувати програму теж, як правило, доводиться людям, які не брали участі в її розробці.

\*Тому хороша архітектура повинна давати можливість відносно легко і швидко розібратися в системі новим

людей. Проект повинен бути добре структурований, не містити дублювання, мати добре оформлений код і бажано документацію. І по можливості, в системі слід застосовувати стандартні, загальноприйняті рішення, що звичайні для програмістів. Чим екзотичніше система, тим складніше її зрозуміти іншим (Принцип найменшого подиву – Principle of least astonishment; зазвичай, він використовується відносно призначеного для користувача інтерфейсу, але застосуємо і до написання коду).

**Особливості проектування архітектури корпоративного додатку на прикладі системи обліку робіт аграрного підприємства.** Аграрна галузь є однією з найскладніших галузей промисловості. Вона об'єднує у собі до прикладу автомобільну, геодезичну, біологічну сфери, сферу документообігу, складського обліку та безлічі інших. Саме тому першим кроком проектування архітектури корпоративного додатку є виявлення предметних підобластей та встановлення зв'язків між ними. У таблиці 1 представлено визначення предметних підобластей та взаємозв'язків між ними.

Таблиця 1

Визначення предметних підобластей

Предметна підобласть	Контекст підобласті
Транспорт	Типи, підтипи, моделі, транспортних засобів, типи та моделі агрегатів
Об'єкти	Контурні об'єкти, поля, склади, адміністративні одиниці, тощо
Культури	Культури що вирощуються на полях
Матеріали	Паливо, насіння, хімічні добрива, тощо
Документація	Види робіт, фази обробітку, наряди на виконання робіт, облікові листи, тощо

Таблиця 2

Визначення взаємозв'язків між підобластями в аграрній галузі

	Транспорт	Об'єкти	Культури	Матеріали	Документація
Транспорт					X
Об'єкти			X		X
Культури		X		X	X
Матеріали			X		X
Документація	X	X	X	X	

Основна взаємодія між підобластями відбувається завдяки підобласті «Документація» яка являється своєрідним буфером між іншими підобластями накопичуючи та розподіляючи інформацію про контексти інших підобластей. Окрім цієї основної зв'язки існує ще дві: зв'язка «Матеріали – Культури» та зв'язка «Культури - Об'єкти». Відношенням «Матеріали – Культури» є, до прикладу, наявність у контексті матеріалів насіння певної культури, а відношенням «Культури - Об'єкти» – належність певної культури до контуру поля, на якому вона вирощується.

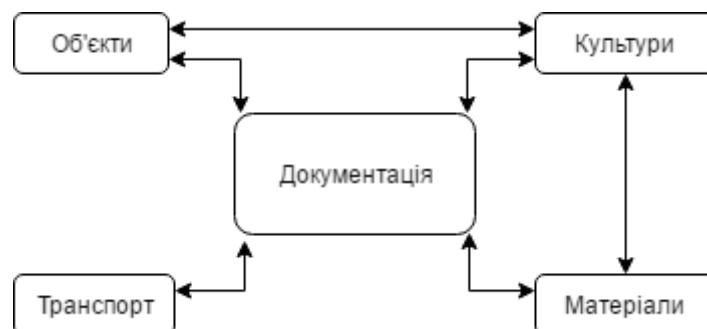


Рис. 1. Карта контекстних областей предметної області

На рис. 2 наведено архітектуру програмного додатку для обліку аграрного підприємства.

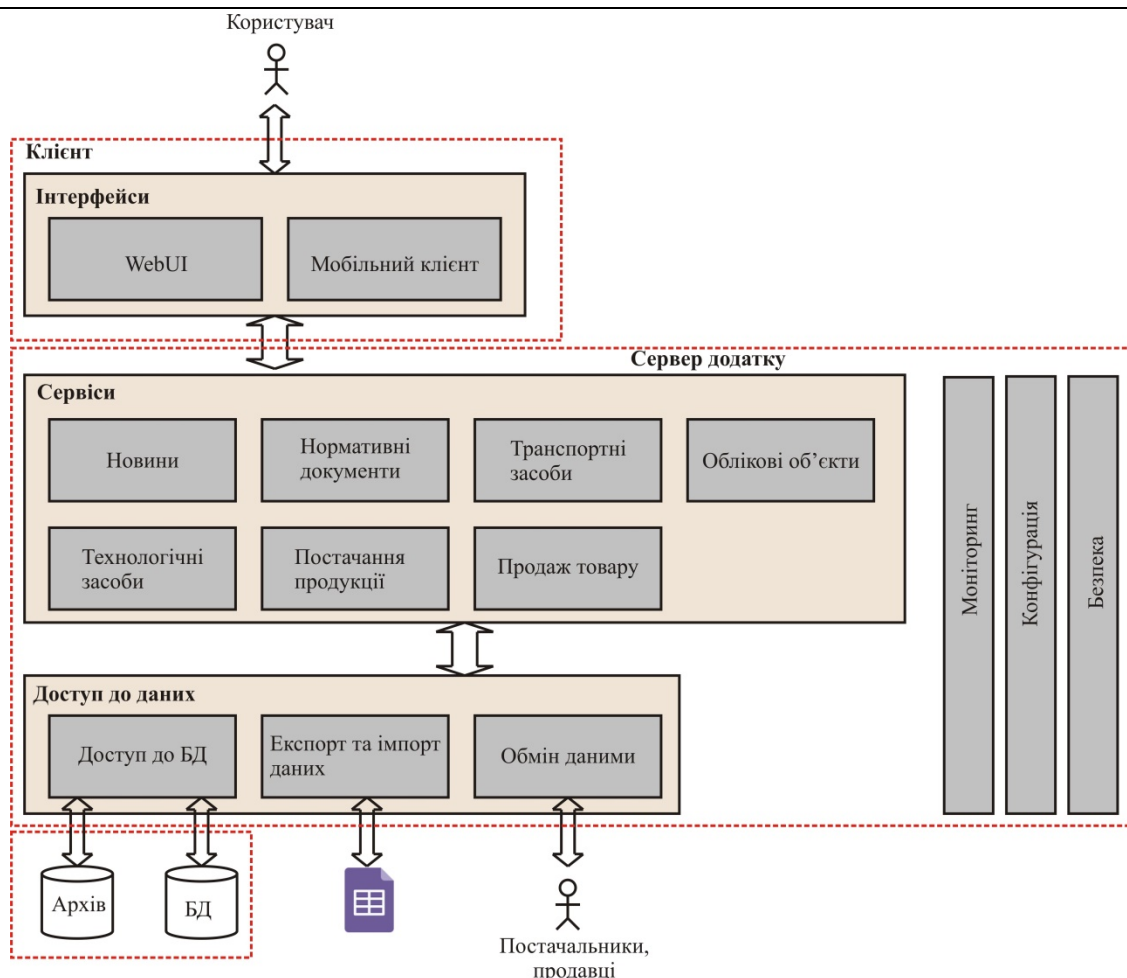


Рис. 2. архітектура програмного додатку для обліку аграрного підприємства

а) Форма визначення початкової інформації

б) Форма вибору технічної операції

в) Форма вибору транспортних засобів

г) Форма вибору облікових об'єктів

Рис. 3. Віконні форми програмного додатку обліку аграрного підприємства: а) форма визначення початкової інформації, б) форма вибору технічної операції, в) Форма вибору транспортних засобів, г) форма вибору облікових об'єктів, д) форма вибору облікових об'єктів (для базових операцій)

При проектуванні програмного додатку облік нарядів повинен бути у вигляді списку де

відображатиметься базова інформація про роботи. Іншою вимогою є забезпечення детального перегляду кожного наряду де відображатиметься повна та детальна інформація. Також необхідно передбачити відображення активних нарядів у режимі реального часу та відображення історії скасованих та завершених нарядів використовуючи сторінкову навігацію. Окрім того, додаток повинен містити можливість фільтрації нарядів за основними інформативними полями. Редактор нарядів повинен складатись з декількох вкладок - для кожного виду робіт: основної механізованої, допоміжної механізованої та допоміжної ручної. Віконні форми розробленого додатку обліку аграрного підприємства наведено на рис. 3.

### Висновок

В роботі проведено опис критеріїв, яким повинно задовольняти програмне забезпечення обліку робіт аграрного підприємства. Здійснено визначення предметних підобластей та взаємозв'язків між ними на основі яких розроблено архітектуру програмного забезпечення обліку робіт аграрного підприємства. Використовуючи спроектовану архітектуру було розроблено програмне забезпечення обліку робіт аграрного підприємства.

### Література

1. Fowler, M., Rice D., Foemmel, M., Hieatt, E., Mee, Stafford, R.: Patterns of Enterprise Application Architecture, Addison-Wesley, pp. 533 (2014).
2. Gramma, E., Helm, R., Johnson, R., Vlissides J.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, pp. 403 (2000).
3. Richter, J.: CLR via C# (Developer Reference), 4th Edition, Microsoft Press, pp. 896-902 (2012).
4. Garlan, D., Shaw, M.: An Introduction to Software Architecture, Technical Report, University Pittsburgh, PA, USA, pp. 457 (1994).
5. Clements, P.C., Heitmeyer, C.L., Labaw, B.G., Rose, A.T.: MT: A toolset for specifying and analyzing real-time systems, In Proc. of the Real-Time Systems Symposium 93, pp. 12-22 (1993).

Отримана/Received : 15.5.2017 р. Надрукована/Printed : 10.6.2017 р.  
Рецензент: д.т.н., проф. Боровик О.В.

УДК 004.415

О.А. КРАВЧУК

Хмельницький національний університет

## ПРАКТИЧНИЙ ОГЛЯД PHP 7

В статті розглядається PHP7 – мова програмування загального призначення з відкритим вихідним кодом. PHP сконструйований спеціально для ведення веб-розробок. PHP – це кросплатформенна технологія. Дистрибутив PHP доступний для більшості операційних систем, включаючи Linux, багатьох модифікацій Unix, Microsoft Windows, Mac OS. PHP підтримується на більшості веб-серверів, таких як Apache, Nginx, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server та інших.

Ключові слова: PHP7, мова програмування, 7 версія, висока швидкодія, оптимізація, мультиплексування, веб-сервер, відкритий код, веб-розробка.

О.А. KRAVCHUK

Khmelnytsky National University

### PRACTICAL PHP 7 REVIEW

In the article we review PHP7 - general-purpose programming language. It is open source software. PHP is designed specifically for web development. PHP it is cross-platform technology. Distribution PHP is available for most operating systems, including Linux, many modifications of Unix, Microsoft Windows, Mac OS. PHP is supported on most web servers such as Apache, Nginx, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server, and many others. The most easily recognizable advantage of the new PHPNG engine is the significant performance improvement. The development team of PHPNG refactored the Zend Engine, and remarkably optimized memory usage. By using PHP 7 not only your code will be executed faster but you will also need fewer servers to serve the same amount of users.

Keywords: PHP7, programming language, version 7, high performance, optimization, multiplexing, Web server, open source, web development.

**Постановка проблеми.** Однією з найбільш значних подій, що відбулися в світі програмування, став випуск PHP 7. Цілих 10 років відділяють його від випуску першого релізу PHP з номером 5. Зі збільшенням першої цифри в номері версії, в PHP 7 з'явилася маса нововведень, збільшилася й швидкість роботи.

Практично всі оновлення додавали невеликі поліпшення продуктивності PHP. Однак на цей раз продуктивність PHP, в порівнянні з ранніми версіями, зросла куди значно більше, ставши однією з найочікуваніших особливостей PHP 7. Це було частиною проекту "PHPNG" ("php new generation"), яка