



MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE



NATIONAL UNIVERSITY OF LIFE AND ENVIRONMENTAL
SCIENCES OF UKRAINE



WROCLAW UNIVERSITY OF ENVIRONMENTAL AND LIFE
SCIENCES



XII International scientific conference of young scientists

**INFORMATION TECHNOLOGY:
ECONOMICS, TECHNICS,
EDUCATION '2021**

November 11– 12, 2021

Kyiv, NULES Ukraine

Kyiv 2021



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ



ВРОЦЛАВСЬКИЙ ПРИРОДНИЧИЙ УНІВЕРСИТЕТ



XII Міжнародна науково-практична конференція молодих вчених

**ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ:
ЕКОНОМІКА, ТЕХНІКА,
ОСВІТА '2021**

11-12 листопада 2021 року

Київ, НУБіП України

Київ 2021

УДК 004

Рекомендовано до друку вченою радою факультету інформаційних технологій
Національного університету біоресурсів і природокористування України

Відповідальний за випуск: Д.Ю. Ящук

Збірник матеріалів XI Міжнародної науково-практичної конференції молодих вчених
«ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ЕКОНОМІКА, ТЕХНІКА, ОСВІТА '2021», 11-12
листопада 2021 року, НУБіП України, Київ. – 229 с. (електронне видання)

Відповідальність за зміст публікацій несуть автори.

*Передрук матеріалів, а також використання їх будь-якій формі допускається лише з
дозволу авторів*

© Національний університет біоресурсів
і природокористування України, 2021

ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ В СЕРВІСІ SMART- ВАГІВ <i>Хвост В.В., науковий керівник Смолій В.В.</i>	192
КОМБІНОВАНІ ФОТОЕЛЕКТРИЧНІ МОДУЛІ <i>Чорноус О.Б., Місюра М.Д.</i>	194
ДОСЛІДЖЕННЯ СУМІЩЕНОГО АВТОМАТА З ЕЛЕМЕНТАРНИМИ ЛАНЦЮГАМИ СТАНІВ <i>Шадрін М. А., науковий керівник Зеленцова І.Я.</i>	196
МАСШТАБУВАННЯ ХМАРНИХ СЕРЕДОВИЩ <i>Шевченко В.М., науковий керівник Шкарупило В.В.</i>	198
РОЗРОБЛЕННЯ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ СТАНУ РОСЛИН У ТЕПЛИЦЯХ <i>Шолудивий Б.В., науковий керівник Шкарупило В.В.</i>	200
SECTION 5. DIGITAL EDUCATIONAL TECHNOLOGIES: DEVELOPMENT AND IMPLEMENTATION / СЕКЦІЯ 5. ЦИФРОВІ ОСВІТНІ ТЕХНОЛОГІЇ: РОЗРОБКА ТА ВПРОВАДЖЕННЯ	202
ТИПИ ЕЛЕКТРОННИХ НАВЧАЛЬНИХ РЕСУРСІВ В УМОВАХ ВІДКРИТОГО ОСВІТНЬОГО СЕРЕДОВИЩА УНІВЕРСИТЕТУ <i>Гавриленко А.О., науковий керівник Глазунова О.Г.</i>	202
СИСТЕМА АНАЛІЗУ ЯКОСТІ НАВЧАЛЬНОГО КОНТЕНТУ У ДИСТАНЦІЙНІЙ ОСВІТІ <i>Демеркова А.А. науковий керівник Глазунова О.Г.</i>	204
ПРИНЦИПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ПІДХОДУ «ЧИСТОЇ АРХІТЕКТУРИ» <i>Деренівський Ярослав Володимирович, науковий керівник Барна Ольга Василівна</i>	207
ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ І ДОДАТКІВ ДЛЯ МОБІЛЬНИХ ПРИСТРОЇВ У ФОРМУВАННІ ФАХОВИХ ЦИФРОВИХ КОМПЕТЕНЦІЙ ЗДОБУВАЧІВ АГРАРНОГО ПРОФІЛЮ ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ МОЛОДШИЙ БАКАЛАВР <i>Євстрат'єв С. В., науковий керівник Самойленко О.М.</i>	209
ДОПОВНЕНА РЕАЛЬНІСТЬ – СУЧАСНА ТЕНДЕНЦІЯ РОЗВИТКУ В ОСВІТІ <i>Заверткін А.А., науковий керівник Глазунова О.Г.</i>	211
АНАЛІЗ, СТВОРЕННЯ І ВПРОВАДЖЕННЯ ЧАТ - БОТІВ В ОСВІТНІЙ ПРОЦЕС <i>Літвінчук О. А., Іскоростенський О. О., наукові керівники Швиденко М.З., Іванченко Н.О.</i>	213
ЗАСТОСУВАННЯ ТЕХНОЛОГІЇ КРАУДСОРСИНГУ ДЛЯ ОРГАНІЗАЦІЇ ПРАКТИЧНОГО НАВЧАННЯ ТА ПРОФОРІЄНТАЦІЙНОЇ ДІЯЛЬНОСТІ <i>Мороз О.О., науковий керівник Кузьмінська О.Г.</i>	215

ПРИНЦИПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ПІДХОДУ «ЧИСТОЇ АРХІТЕКТУРИ»

Деренівський Ярослав Володимирович, науковий керівник Барна Ольга Василівна

Актуальність теми. Програмування давно вже стало інструментом для вирішення сучасних задач. Зараз існує велика кількість мов програмування, які використовують в залежності від поставленого технічного завдання. Цей вибір зумовлює необхідність дотримання методологій розробки, наприклад, таких як принципи об'єктно-орієнтоване програмування та акроніму S.O.L.I.D. Також важливу роль для сучасного програмного забезпечення відіграє архітектура. У зв'язку із цим є потреба уточнення принципів розробки програмного забезпечення, що є **метою даного дослідження**.

Виклад основного матеріалу. Поняття «чистої архітектури» (clean architecture) вперше згадав Uncle Bob (Robert Martin) у своїй однойменній книзі про чисту архітектуру [1]. На думку автора в архітектурі програмного проекту кожен модуль повинен бути незалежним. Основною перевагою такого підходу є те що коли будь-яка із зовнішніх частин системи стає застарілою, як-от база даних або веб-фреймворк, то ці застарілі елементи можна легко поміняти.

Принцип «чистої архітектури» суміщає ідеї кількох інших архітектурних підходів, а саме: бути тестованою; не залежати від інтерфейсу; бази даних, зовнішніх фреймворків і бібліотек. Це досягається розділенням на рівні і слідуванням так званим *правилами залежності (Dependency Rule)* [2]. Правила залежності вказують, що внутрішні шари не повинні залежати від зовнішніх. На рисунку 1 це правило зображено стрілками, що вказують усередину.

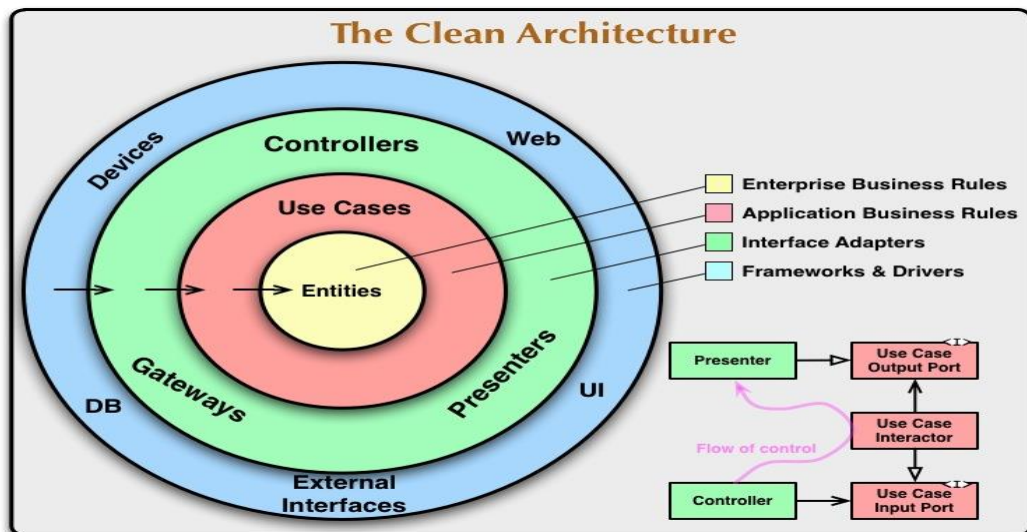


Рис. 1. Модель чистої архітектури
Джерело: <https://i.stack.imgur.com/RXfYa.png>

Принцип відкритості/закритості (open-closed principle/OCP) декларує, що програмні сутності (класи, модулі, функції тощо) мають бути відкриті для розширення, але закриті для зміни. Це означає, що ці сутності можуть змінювати свою поведінку без зміни їхнього вихідного коду.

Виділяють 4 шари:

- Entities – бізнес-логіка, яка загальна для багатьох програм.
- Use Cases (Interactors) – логіка програми.

- Interface Adapters – включає адаптери між Use Cases та зовнішнім світом. Сюди потрапляють Presenters з MVP, а також Gateways (популярніша назва репозиторії).
- Frameworks. Містить: інтерфейс (UI), базу даних, http-клієнт, і т.п.
- Важливим, на нашу думку, є розуміння шаблону MVC. Маємо:
 - Model - відповідає за сутності предметної області;
 - Controller - відповідає за бізнес логіку ПЗ;
 - View - відповідає за візуальну частину.

Якщо, все це візуалізувати, то отримуємо, що в нас кожен модуль отримує дані від попереднього, а саме:

1. Користувач взаємодіє з візуальною частиною;
2. Система створює запит для контролерів;
3. Контролер звертається до сутностей;
4. Сутності, звертаються до контролера, в якому виконується бізнес логіка;
5. Контролер повертає результат у візуальну частину (рис.1).

Зауважимо, що S.O.L.I.D – це аббревіатура з п'яти основних принципів проектування в об'єктно-орієнтованому програмуванні, яка запропонована Робертом Мартіном [3]. Ці принципи дозволяють будувати на базі ОВП масштабовані та супроводжувані програмні продукти зі зрозумілою бізнес-логікою.

Принцип єдиного обов'язку/відповідальності (*single responsibility principle/SRP*) означає, що кожен об'єкт повинен мати один обов'язок і цей обов'язок повинен бути повністю інкапсульований у клас. Усі його послуги мають бути спрямовані виключно на забезпечення цього обов'язку.

Принцип підстановки Барбери Лісков (Liskov substitution principle/LSP) у формулюванні Роберта Мартіна: «функції, які використовують базовий тип, повинні мати можливість використовувати підтипи базового типу, не знаючи про це».

Принцип поділу інтерфейсу (*interface segregation principle/ISP*) означає, що занадто складні інтерфейси необхідно розділяти на більш маленькі та специфічні, щоб клієнти маленьких інтерфейсів знали лише про методи, які їм необхідні в роботі. У результаті при зміні методу інтерфейсу не повинні змінюватися клієнти, які не використовують цей метод [1].

Принцип інверсії залежностей (*dependency inversion principle/DIP*) - модулі верхніх рівнів не повинні залежати від модулів нижніх рівнів, а обидва типи модулів повинні залежати від абстракцій; самі абстракції не повинні залежати від деталей, а деталі повинні залежати від абстракцій.

Висновок. Розділивши програмне забезпечення на рівні та дотримуючись *правила залежності*, можна створити систему, яку можна легко перевірити та змінити. Коли будь-яка із зовнішніх частин системи стає застарілою, як-от база даних або веб-фреймворк, можна легко і швидко замінити ці застарілі елементи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SOLID - принципи об'єктно-орієнтованого програмування URL: <https://web-creator.ru/articles/solid> (дата звернення 07.11.2021).
2. Помилки Clean Architecture. URL: <https://habr.com/ru/company/mobileup/blog/335382/> (дата звернення 07.11.2021).
3. The Clean Code Blog by Robert C. Martin (Uncle Bob). The Clean Architecture. URL: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html> (дата звернення 07.11.2021).