



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ



Кафедра кібербезпеки та програмного забезпечення

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Патерни проектування програмного забезпечення

першого рівня вищої освіти

КРОПИВНИЦЬКИЙ

1. Загальна інформація

Назва дисципліни	Патерни проектування програмного забезпечення
Розробник(и)	– Коваленко Олександр Володимирович , доктор технічних наук, доцент кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету – Software Engineer, Україна м. Львів «EPAM Systems» Кобець М.О.
Викладач	Лектор – Коваленко Олександр Володимирович , доктор технічних наук, доцент, https://kbpz.kntu.kr.ua/kafedra/kovalenko-oleksandr-volodymyrovych https://scholar.google.com.ua/citations?user=Za6XQuoAAAAJ https://publons.com/researcher/2914876/alexander-kovalenko https://orcid.org/0000-0001-9297-0650
Контактний телефон	службовий: (0522)390-449 – робочі дні з 8 ³⁰ до 14 ²⁰
E-mail:	dr.kovalenkoov@gmail.com
Консультації	<i>очні</i> – відповідно до затвердженого графіку консультацій; <i>онлайн</i> – е-листування, у месенджері, вебінари на платформах Zoom, Discord

2. Анотація дисципліни

Навчальний курс «Патерни проектування програмного забезпечення» призначений для набуття теоретичних знань та практичних навичок роботи з патернами проектування, тобто вирішення певної проблеми, що часто зустрічається при проектуванні архітектури програмного забезпечення. Під час вивчення матеріалу надається можливість вивчити загальний принцип вирішення певного комплексу проблем, які майже завжди треба підлаштовувати для потреб того чи іншого програмного забезпечення.

3. Мета і завдання дисципліни

Метою викладання навчальної дисципліни «Патерни проектування програмного забезпечення» є забезпечення здобувачів вищої освіти комплексом знань, умінь та навичок, необхідних для застосування в професійній діяльності патернів проектування програмного забезпечення.

Основними **завданнями** вивчення навчальної дисципліни є формування наступних компетенцій:

- Вміння виявляти, ставити та вирішувати проблеми.
- Здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.
- Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення.

4. Формат дисципліни

Для денної форми навчання:

Викладання курсу передбачає для засвоєння дисципліни традиційні лекційні заняття із застосуванням мультимедійних презентацій, у поєднанні з лабораторними заняттями.

Формат очний (*Face to face*)

Для заочної форми навчання:

Під час сесії формат очний (*Face to face*), у міжсесійний період – дистанційний (*online*).

5. Результати навчання

У результаті вивчення навчальної дисципліни студент буде:

Вміти:

- застосовувати знання для ідентифікації, формулювання і розв’язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
- системно мислити та застосовувати творчі здібності до формування нових ідей.

Набувати навичок автономії і відповідальності:

- Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

6. Обсяг дисципліни

Ознака дисципліни, вид заняття	Кількість годин
Кількість кредитів / годин	4/120
Кількість змістових модулів	2
Нормативна / вибіркова	вибіркова
Вид підсумкового контролю:	залік

7. Технічне і програмне забезпечення / обладнання

Лекційні заняття проводяться в аудиторіях обладнаних мультимедійним проектором. Лабораторні роботи виконуються у аудиторіях кафедри кібербезпеки та програмного забезпечення, обладнаних відповідним апаратним та програмним забезпеченням (ауд. 501, 507, 508, 517), з відкритою бездротовою мережею Wi-Fi, вільним доступом до Інтернету. Оскільки при вивченні дисципліни використовуються інформаційні технології навчання, система дистанційної освіти Moodle, студенту необхідно мати комп'ютерну техніку (з виходом у Internet) та оргтехніку для комунікації з викладачами, виконання тестових завдань в системі дистанційної освіти. Застосовується матеріально-технічна база кафедри кібербезпеки та програмного забезпечення.

8. Політика дисципліни

Організація освітнього процесу. Учасники освітнього процесу повинні дотримуватися вимог Положення про організацію освітнього процесу ЦНТУ, Кодексу академічної доброчесності ЦНТУ, Положення про дотримання академічної доброчесності НПП та здобувачами вищої освіти, інших нормативних актів університету <http://www.kntu.kr.ua/?view=univer&id=4>.

Академічна доброчесність:

Очікується, що студенти будуть дотримуватися принципів академічної доброчесності, усвідомлювати наслідки її порушення. Детальніше за посиланням URL : <http://www.kntu.kr.ua/doc/dobro.pdf>

Відвідування занять

Є важливою складовою навчання. Очікується, що всі студенти відвідають лекції і лабораторні заняття курсу.

Пропущені заняття повинні бути відпрацьовані не пізніше, ніж за тиждень до залікової сесії.

Поведінка на заняттях

Недопустимість: запізнь на заняття, списування та плагіат, несвоєчасне виконання поставленого завдання.

При організації освітнього процесу в Центральнотехнічному національному університеті студенти, викладачі та адміністрація діють відповідно до: Положення про організацію освітнього процесу; Положення про організацію вивчення навчальних дисциплін вільного вибору; Положення про рубіжний контроль успішності і сесійну атестацію студентів ЦНТУ, Кодексу академічної доброчесності ЦНТУ.

9. Навчально-методична карта дисципліни

Лекція 1. Основи об'єктно-орієнтованої методології розробки програмного забезпечення. Поняття об'єкта і класу, інкапсуляція, наслідування та поліморфізм як основа моделювання предметної області і побудови масштабованих програмних систем.

Лекція 2. Основи використання патернів проектування програмного забезпечення. Архітектура та базові принципи проектування програмного забезпечення. Архітектура та базові принципи проектування програмного забезпечення. Роль патернів у проектуванні, розмежування відповідальностей, абстракції та модульність, а також взаємозв'язок патернів із архітектурними рішеннями та принципами підтримуваності коду.

Лекція 3. П'ять базових принципів об'єктно-орієнтованого програмування та проектування SOLID. Зміст і практичне застосування SRP, OCP, LSP, ISP, DIP для зниження зв'язності, підвищення розширюваності та керованості складністю під час розвитку програмних продуктів.

Лекція 4. Породжувальні патерни Builder, Factory Method, Singleton. Керування процесом створення об'єктів, інкапсуляція логіки ініціалізації та конфігурації, а також контроль життєвого циклу екземплярів у прикладних і серверних застосунках.

Лекція 5. Структурні патерни Facade, Adapter. Спрощення доступу до підсистем через єдиний інтерфейс, узгодження несумісних інтерфейсів і забезпечення інтеграції компонентів без зміни їх внутрішньої реалізації.

Лекція 6. Структурні патерни Decorator, Strategy. Динамічне розширення функціональності без модифікації базових класів, заміна поведінки через взаємозамінні алгоритми та побудова гнучких механізмів конфігурації на рівні виконання.

Лекція 7. Поведінкові патерни State, Iterator. Керування зміною поведінки об'єкта залежно від внутрішнього стану та організація стандартного проходження колекцій із приховуванням деталей структури даних і забезпеченням єдиного способу ітерації.

Лекція 8. Поведінкові патерни Observer та Command. Подієво-орієнтована взаємодія об'єктів, слабке зв'язування компонентів та інкапсуляція запитів у сучасних програмних системах.

Лабораторна робота 1. Основи використання породжувального патерну Factory Method.

Лабораторна робота 2. Основи використання породжувального патерну Builder.

Лабораторна робота 3. Основи використання породжувального патерну Singleton.

Лабораторна робота 4. Основи використання структурного патерну Facade.

Лабораторна робота 5. Основи використання структурного патерну Adapter.

Лабораторна робота 6. Основи використання структурного патерну Decorator.

Лабораторна робота 7. Основи використання поведінкового патерну Strategy

Лабораторна робота 8. Основи використання поведінкових патернів Observer та Command.

10. Система оцінювання та вимоги

Види контролю: поточний, підсумковий.

Методи контролю: спостереження за навчальною діяльністю, усне опитування, письмовий контроль, тестовий контроль.

Форма підсумкового контролю: залік.

Контроль знань і умінь здобувачів (поточний і підсумковий) здійснюється згідно з кредитною трансферно-накопичувальною системою організації освітнього процесу в ЦНТУ. Рейтинг студента із засвоєння дисципліни визначається за 100-бальною шкалою. Він складається із рейтингу з поточної навчальної роботи впродовж семестру, для оцінювання якої призначається 100 балів (по 50 балів за кожен змістовний модуль).

При виставленні оцінки враховуються результати навчальної роботи студента протягом семестру

Критерії оцінки заліку:

– «зараховано» – студент має стійкі знання про основні поняття дисципліни, може сформулювати взаємозв'язки між поняттями.

– «незараховано» – студент має значні пропуски в знаннях, не може сформулювати взаємозв'язку між поняттями, що вивчаються в курсі, не має уявлення про більшість основних понять дисципліни, що вивчається.

11. Рекомендовані література й джерела

Базова

1. Kovalenko O., Khudov H., Mynko P., Ikhsanov S., Diakonov O., Solomonenko Y., Drob Y., Kharun O., Cherkashyn S., Serdiuk O. «Development A Method For Determining The Coordinates Of Air Objects By Radars With The Additional Use Of Multilateration Technology» Eastern-European Journal of Enterprise Technologies Volume 5, 2021, Pages 6-16. Режим доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85119667497&origin=resultslist> (**Scopus**).
2. Коваленко О.В. Управління ризиками розробки програмного забезпечення за умови обмеженості коштів виділених на усунення помилок безпеки. Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Випуск 31. – Кропивницький: ЦНТУ. – 2018. – С. 128 – 140. Режим доступу: <http://dspace.kntu.kr.ua/jspui/bitstream/123456789/8130/1/18.pdf> (**Фахове видання. Категорія «Б»**)
3. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137. Режим доступу: http://www.tech.vernadskyjournals.in.ua/journals/2018/5_2018/part_1/25.pdf (**Фахове видання. Категорія «Б»**)
4. Коваленко О.В. Масштабування імітаційної моделі технології тестування безпеки. Системи управління, навігації та зв'язку. – Випуск 6 (46). – Полтава: ПолтНТУ. – 2017. – С. 181-184. Режим доступу: http://nbuv.gov.ua/UJRN/suntz_2017_6_42 (**Фахове видання. Категорія «Б»**)
5. Коваленко О.В., Смирнов О.А. Розробка методу управління ризиками розробки програмного забезпечення. Інформаційні технології: проблеми та перспективи: **монографія** / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.

Допоміжна

6. Eric Freeman Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. O'Reilly Media, 2021. 669 с.
7. Eric Freeman, Elisabeth Robson Head First Design Patterns. 2020. 672 pages.
8. Alexey Soshin. Kotlin Design Patterns and Best Practices. Packt Publishing Ltd. 2022. 356 с.
9. Kasun Indrasiri, Sriskandarajah Suhothayan Design Patterns for Cloud Native Applications: Patterns in Practice Using APIs, Data, Events, and Streams. O'Reilly Media. 2021. 314 с.
10. Jorge Luis Ortega-Arjona Patterns for Parallel Software Design. Wiley; 1st edition. 2010. 438 с.
11. Ivan Nikolov Scala Design Patterns: Design modular, clean, and scalable applications by applying proven design patterns in Scala, 2nd Edition. Packt Publishing. 2018. 396 с.
12. James W. Cooper Python Programming with Design Patterns 1st Edition. 2021. 352 с.
13. David Baron Game Development Patterns with Unity 2021: Explore practical game development using software design patterns and best practices in Unity and C#. Packt Publishing. 2021. 246 с.
14. Martin Fowler Patterns of Enterprise Application Architecture. Addison-Wesley Professional. 2002 560 с.
15. Stoyan Stefanov JavaScript Patterns: Build Better Applications with Coding and Design Patterns. O'Reilly Media. 2010. 236 с.
16. Lamis Chebbi Reactive Patterns with RxJS for Angular: A practical guide to managing your Angular application's data reactively and efficiently using RxJS 7. Packt Publishing. 2022. 224 с.
17. Dusty Phillips Python 3 Object-Oriented Programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8. Packt Publishing. 2018. 466 с.
18. Kamon Ayeva, Sakis Kasampalis Mastering Python Design Patterns: A guide to creating smart, efficient, and reusable software. Packt Publishing. 2018. 248 с.

19. Harry Percival Architecture Patterns with Python: Enabling Test-Driven Development, Domain-Driven Design, and Event-Driven Microservices. O'Reilly Media. 2020. 304 с.
20. International Standard ISO/IEC 14882:2017 – Information technology – Programming languages – C++. – ISO/IEC, 2017.
21. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення. Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137. Режим доступу: http://www.tech.vernadskyjournals.in.ua/journals/2018/5_2018/part_1/25.pdf
22. Ерік Фрімен, Елізабет Робсон, Берт Бейтс, Кеті Сієрра First. Патерни проєктування. 2020. 672 с.

Методичне забезпечення

23. Коваленко О.В., Коваленко А.С., Савеленко О.К. «Патерни проєктування програмного забезпечення». Методичні вказівки до виконання лабораторних робіт для студентів денної форми навчання галузі 12 Інформаційні технології. – Кропивницький: ЦНТУ – 2022. – 57 с.
24. Коваленко О.В., Коваленко А.С., Савеленко О.К. «Патерни проєктування програмного забезпечення». Методичні вказівки до виконання контрольних робіт для студентів заочної форми навчання галузі 12 Інформаційні технології. – Кропивницький: ЦНТУ – 2022. – 55 с.

Інформаційні ресурси

25. Основи створення мобільних додатків [Електронний ресурс]. – Режим доступу: <https://developer.android.com/guide>
26. Онлайн-курси UDEMY. – URL: <https://www.udemy.com/> – платформа онлайн-курсів різних ІТ тематик.
27. Онлайн-курси Prometheus. – URL: <https://prometheus.org.ua/> – українська платформа безкоштовних онлайн-курсів
28. Онлайн-курси Coursera. – URL: <https://www.coursera.org> – платформа онлайн-курсів різних ІТ тематик.
29. <https://habr.com> – колективний блог з новинами та аналітичними статтями про інформаційні технології та програмування.
30. <http://stackoverflow.com/> – система питань і відповідей для професійних програмістів та новачків у програмуванні.
31. <https://dou.ua/> – український веб-сайт з елементами колективного блогу, створений для розповсюдження новин, аналітичних статей та свіжої інформації пов'язаної із інформаційними технологіями.
32. <https://www.google.com/> – основна пошукова платформа.
33. <https://www.youtube.com> – Відеохостинг, що надає користувачам послуги зберігання, доставки та показу відео. На платформі розміщено багато курсів ІТ спрямованості.
34. <https://biblprog.org.ua/ua/programming/> – каталог безкоштовних середовищ розроблення ПЗ.
35. Національна бібліотека України імені В. І. Вернадського: Електронні ресурси НБУВ [Електронний ресурс]. – Режим доступу: <http://www.nbuv.gov.ua/>.