



Силабус освітнього компонента

Програма навчальної дисципліни



Архітектура та проектування програмного забезпечення

Шифр та назва спеціальності

121 – Інженерія програмного забезпечення

Інститут

ННІ Комп'ютерних наук та інформаційних технологій

Освітня програма

Інженерія програмного забезпечення

Кафедра

Програмна інженерія та інтелектуальні технології управління (321)

Рівень освіти

Бакалавр

Тип дисципліни

Спеціальна (фахова), Обов'язкова

Семестр

5-6

Мова викладання

Українська, англійська

Викладачі, розробники



Двухглавов Дмитро Едуардович

dmytro.dvukhhlavov@khipt.edu.ua

к.т.н., доцент, доцент кафедри програмної інженерії та інтелектуальних технологій управління

Google Scholar:

<https://scholar.google.com/citations?user=OAzyFg8AAAAJ&hl>

ORCID: <https://orcid.org/0000-0002-3361-3212>

Scopus: <https://www2.scopus.com/authid/detail.uri?authorId=57211294555>

Web of Science: <https://www.webofscience.com/wos/author/record/E-8279-2019>).

[Детальніше про викладача на сайті кафедри](#)

Загальна інформація

Анотація

Викладення дисципліни забезпечує готовність студентів до участі у вирішенні питань обґрунтування вибору архітектури програмного продукту, визначення складу та структури його компонентів із врахуванням встановлених вимог до якості, застосуванню патернів проектування при розробці структури ПЗ. В ході вивчення дисципліни забезпечується розуміння студентами змісту процесу створення програмного забезпечення у ІТ-компаніях, а також відпрацювання базових навиків з проектування програмного забезпечення та сумісної роботи із кодом проекту, необхідних для роботи у команді розробників програмного забезпечення.

Мета та цілі дисципліни

Викладення дисципліни має на меті поглиблення знань про життєвий цикл програмного забезпечення та стандарти, що регламентують реалізацію його етапів, надання знань про архітектуру програмного забезпечення, архітектурні стилі сучасного програмного забезпечення, відомі шаблони проектування програмного забезпечення, а також фреймворки опису архітектури програмного забезпечення, про принципи організації розробки програмного забезпечення на сучасному етапі розвитку ІТ-індустрії, про технології створення програмного забезпечення, про стратегії та методології проектування та розробки програмного забезпечення, про представлення

результатів проектування ПЗ, про принципи проектування інтерфейсу користувачів, а також виробку навиків розробки програмних рішень із застосуванням деяких шаблонів проектування, навиків створення моделей програмного рішення на основі шаблонів проектування в Enterprise Architect, розробки проекту програмного забезпечення на основі визначеного переліку вимог до програмного забезпечення, навиків застосування системи контролю версій git та репозиторію github під час створення ПЗ. Засвоєння дисципліни є важливим елементом для виконання дипломної роботи.

Формат занять

Лекції, лабораторні роботи, самостійна робота, консультації, курсовий проект. Підсумковий контроль – іспит в кінці кожного семестру.

Компетентності

- K01. Здатність до абстрактного мислення, аналізу та синтезу.
- K02. Здатність застосовувати знання у практичних ситуаціях.
- K05. Здатність вчитися і оволодівати сучасними знаннями.
- K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- K07. Здатність працювати в команді.
- K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
- K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.
- K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.
- K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.
- K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- K26. Здатність до алгоритмічного та логічного мислення.

Результати навчання

- ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.
- ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.
- ПР06. Уміння вибирати та використовувати методологію створення програмного забезпечення відповідну до задачі.
- ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
- ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.
- ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.
- ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
- ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.
- ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу,

проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПР16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення

ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.

Обсяг дисципліни

Загальний обсяг дисципліни:

5 семестр - 120 год. (4 кредити ECTS): лекції – 16 год., лабораторні роботи – 32 год., самостійна робота – 72 год.;

6 семестр - 120 год. (4 кредити ECTS): лекції – 32 год., лабораторні роботи – 32 год., самостійна робота – 56 год.

Передумови вивчення дисципліни (пререквізити)

Моделі та структури даних

Об'єктно-орієнтоване програмування

Особливості дисципліни, методи та технології навчання

Методи викладання та навчання

Основним методом навчання в ході лекцій є пояснювально-ілюстративний метод. Для активізації пізнавальної діяльності передбачені виступи студентів та організація дискусій за окремими питаннями лекцій.

Виконання лабораторних робіт у 5 семестрі передбачає створення студентами моделі рішення у вигляді набору діаграм UML та програмного коду для імплементації типового функціоналу програм на основі відомих шаблонів проектування. Для імплементації студенту визначається власна предметна область. Готових алгоритмів для вирішення не надається, що спонукає до прояву творчої активності студентів.

Виконання лабораторних робіт у 6 семестрі передбачає тренування навиків за двома напрямками – відпрацювання навиків роботи із системою контролю версій та відпрацювання навиків проектування програмного забезпечення на основі системи визначених вимог до нього.

Виконання лабораторних робіт з проектування ПЗ фактично відповідає етапам виконання курсового проекту, який має бути захищений на заліковому тижні. Предметна область для проектування є індивідуальною. При розробці курсового проекту акцентується увага на те, що отримані навички важливі для дипломного проектування.

Форми оцінювання:

Засвоєння теорії перевіряється у формі експрес-опитування під час лекцій (CAS), опитування або автоматизованого тестування на початку лабораторних робіт (CAS).

Контроль засвоєння матеріалу для самостійного вивчення передбачає підготовку та захист рефератів за індивідуальною тематикою (2 реферати в кожному семестрі) (CAS).

Перевірка рівня практичних навиків перевіряється на лабораторних роботах, що виконуються за індивідуальними варіантами (CAS).

Підсумковий контроль здійснюється на захисті курсового проекту та під час семестрових екзаменів (FAS).

На екзамені в 5 семестрі передбачається тестування з теоретичних питань.

На екзамені в 6 семестрі передбачається тестування з теоретичних питань та створення моделі невеликого бізнес-процесу та поведінкової моделі UML за індивідуальним завданням у обмежений час.

Програма навчальної дисципліни

Теми лекційних занять

5 семестр:

Тема 1.

Поняття «архітектура програмного забезпечення». Основні положення стандарту IEEE 1471 та інших нормативних документів програмної інженерії. Фреймворки для опису архітектури програмного забезпечення (ПЗ) («4+1», RM-ODP, SOMF).

Тема 2.

Вимоги до архітектури сучасного ПЗ. Сутність використання шаблонів для забезпечення сучасних вимог до ПЗ. Класифікація шаблонів проєктування (Design pattern by GoF). Типовий опис шаблонів проєктування. Детальний опис популярних шаблонів проєктування Шаблони розподілу відповідальностей GRASP.

Тема 3.

Архітектурні стилі. Визначення архітектурного стилю. Огляд архітектурних стилів сучасності.

Тема 4.

Парадигми програмування.

6 семестр:

Тема 5.

Сучасний погляд на процес розробки програмного забезпечення. Фреймворк SCRUM: призначення та принципи застосування під час розробки. Зміст CD/CI (continuous integration/ continuous delivery). Програмне забезпечення для автоматизації розробки програмного забезпечення.

Тема 6.

Проєктування як складова життєвого циклу розробки програмного забезпечення. Основі визначення та нормативні документи. Нотації та представлення результатів проєктування.

Тема 7

UML як нотація для представлення результатів проєктування. Класифікація діаграм UML. Застосування діаграм UML для опису поведінки та структури програмного забезпечення. Програмне забезпечення для створення діаграм UML.

Тема 8.

Стратегії та методології розробки програмного забезпечення.

Тема 9.

Проєктування інтерфейсу користувачів.

Теми практичних занять

Практичні заняття в рамках дисципліни не передбачені.

Теми лабораторних робіт

5 семестр:

ЛР 1. Дослідження процесу установки Enterprise Architect та його застосування для створення базових діаграм UML

ЛР 2. Дослідження особливостей реалізації шаблону BRIDGE

ЛР 3. Дослідження особливостей реалізації шаблону COMPOSITE

ЛР 4. Дослідження особливостей реалізації шаблону COMMAND

ЛР 5. Дослідження особливостей реалізації шаблонів SINGLETON та PROXY

ЛР 6. Дослідження особливостей реалізації шаблону FACTORY METHOD

ЛР 7. Дослідження особливостей реалізації шаблону TEMPLATE METHOD

6 семестр:

ЛР 8. Встановлення та налаштування системи git. Робота в git-bash

ЛР 9. Управління гілками з використанням git

ЛР 10. Використання репозиторію github

ЛР 11. Формалізоване представлення бізнес-процесів та вимог до програмного забезпечення із застосуванням CASE-системи

ЛР 12. Проектування поведінки системи для різних категорій користувачів на основі діаграм UML

ЛР 13. Проектування схеми для зберігання даних

ЛР 14. Проектування структури компонентів програмного забезпечення та схеми його розгортання

Самостійна робота

5 семестр

Студентам рекомендуються додаткові матеріали (відео, статті) для самостійного вивчення та опрацювання з тематики, представленої нижче.

Детальне вивчення стандартів, що визначають поняття «архітектура ПЗ» та її опису. Приклади застосування фреймворків для опису архітектури ПЗ. Встановлення Enterprise Architect. Створення проектів та моделей у Enterprise Architect. Панелі інструментів Enterprise Architect. Створення діаграм класів, послідовності та активності у Enterprise Architect. Призначення шаблонів проектування. Типові ситуації, переваги та недоліки використання шаблонів проектування. Принципи застосування шаблонів розподілу відповідальностей GRASP. Огляд реалізацій застосування шаблонів розподілу відповідальностей. Детальний опис архітектурних стилів. Аналіз реалізації стилю у відомих програмних продуктах. Сучасні погляди на підходи до проектування програмного забезпечення.

6 семестр

Студентами здійснюється курсове проектування, в ході якого відпрацюються питання, представлені нижче.

Вибір задачі для автоматизації та визначення об'єкту автоматизації. Опис бізнес-процесів об'єкту автоматизації. Визначення функціональних та нефункціональних вимог до проектуемого програмного забезпечення. Розробка SRS для створюваного програмного забезпечення. Проектування доменної моделі об'єктів предметної області. Розробка use case діаграм та їх уточнення шляхом розробки сценаріїв або діаграм взаємодії. Проектування Entity-Relation diagram. Обґрунтований вибір системи керування БД. Обґрунтований вибір архітектури програмного забезпечення. Розробка та опис структури компонентів програмного забезпечення та схеми його розгортання. Розробка components and deployment diagram.

Література та навчальні матеріали

Основна література

1. Sommerville Ian. Software Engineering (6th Edition) / https://www.academia.edu/6826193/Ian_Sommerville_Software_Engineering_6th_Edition.
2. Freeman Eric, Robson Elisabeth. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition. – O'Reilly Media, 2020. – 672 p.
3. Larman Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. – Pearson, 2005. – 736 p.
4. Git - Documentation (git-scm.com) // <https://git-scm.com/doc>.
5. Грицюк Ю. І. Аналіз вимог до програмного забезпечення. Навчальний посібник. 2018. – 456 с.
6. Піхлер Р. Agile продукт-менеджмент за допомогою Scrum. Фабула, 2019. – 128 с.
7. Мартін Роберт. Чистий код. – Фабула, 2019. – 416 с.

Додаткова література

Стандарти

6. ISO/IEC/IEEE 12207 Systems and software engineering — Software life cycle processes.
7. IEEE 610-1990. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries.
8. ISO 25010. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models.
9. IEEE 1471. Recommended Practice for Architectural Description of Software-Intensive Systems.
10. ISO/IEC/IEEE 42010. Systems and software engineering — Architecture description.

11. IEEE/ISO/IEC 29148-2018. Systems and software engineering – Life cycle processes – Requirements engineering.
12. IEEE 1016. IEEE Recommended Practice for Software Design Descriptions.

Ресурси інтернет

13. <https://refactoring.guru/ua/design-patterns>.
14. Dr. Winston W. Royce. Managing the development of large software systems // <http://www.scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>.
15. Microsoft Solutions Framework (MSF). Team Model // <https://www.microsoft.com/en-us/download/details.aspx?id=3214>.
16. Manifesto for Agile Software Development // <http://agilemanifesto.org/>
17. UML modeling tools for Business, Software, Systems and Architecture // <https://www.sparxsystems.com/>
18. <https://www.smart-it.com/ru/2021/08/12-best-software-development-methodologies-with-pros-and-cons/>
19. <https://refactoring.guru/ua/design-patterns>.
20. <https://sparxsystems.com>

Система оцінювання

Критерії оцінювання успішності студента та розподіл балів

Оцінка за дисципліну складається з балів, набраних під час навчання в ході семестру, та балів, що виставляються за складання екзамену. В ході 5 семестру студент може отримати до 70 балів за:

- засвоєння теорії (тематики самостійної роботи) (до 30 балів);
- виконання 1ої лабораторної роботи (4 бали);
- виконання 2-7 лабораторних робіт (по 6 балів за кожну).

За складання екзамену студент може отримати до 30 балів.

В ході 6 семестру студент може отримати 80 балів за:

- засвоєння теорії (до 15 балів);
- виконання ЛР 1-3 (робота із Git) (до 15 балів);
- виконання ЛР 3-7 (проектування ПЗ) та захист курсового проекту (до 50 балів).

За складання екзамену студент може отримати до 20 балів.

Шкала оцінювання

Сума балів	Національна оцінка	ECTS
90–100	Відмінно	A
82–89	Добре	B
75–81	Добре	C
64–74	Задовільно	D
60–63	Задовільно	E
35–59	Незадовільно (потрібне додаткове вивчення)	FX
1–34	Незадовільно (потрібне повторне вивчення)	F

Норми академічної етики і політика курсу

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність. Конфліктні ситуації повинні відкрито обговорюватися в навчальних групах з викладачем, а при неможливості вирішення конфлікту – доводитися до відома співробітників дирекції інституту. Нормативно-правове забезпечення впровадження принципів академічної доброчесності НТУ «ХПІ» розміщено на сайті: <http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/>

Погодження

Силабус погоджено

11.04.2023

Завідувач кафедри
Ігор ГАМАЮН

08.06.2023

Гарант ОП
Юлія ЛІТВІНОВА