

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ЧАСТИНА 1)

СИЛАБУС

Шифр і назва спеціальності	121 – Інженерія програмного забезпечення	Інститут / факультет	Комп'ютерних наук і програмної інженерії
Назва програми	«Інженерія програмного забезпечення»	Кафедра	програмної інженерії та інформаційних технологій управління
Тип програми	Освітньо-професійна	Мова навчання	Українська, англійська

ВИКЛАДАЧ

ПІБ, електронна пошта

Двухглавов Дмитро Едуардович,
Dmytro.Dvukhhlavov@khpi.edu.ua



к.т.н., доцент, доцент кафедри програмної інженерії та інформаційних технологій управління.
Підготував і опублікував понад 40 публікацій, 1 стаття у виданнях, індексованих у Scopus, 2 навчальних посібника з грифом внз, 2 методичних посібника до практичних видів робіт.
h-index = 3, i10-index = 0 у Google Академії-<https://scholar.google.com/citations?user=OAzYFg8AAAAJ&hl=ru>; ідентифікатор ORCID-<https://orcid.org/0000-0002-3361-3212>).
Провідний лектор з курсів: Поглиблений курс програмування Java (*бакалаври*) (*українською мовою*), Web-додатки на основі Java (*бакалаври*) (*українською мовою*), Архітектура та проектування програмного забезпечення (частина 1) (*бакалаври*) (*англійською та українською мовами*), Архітектура та проектування програмного забезпечення (частина 2) (*бакалаври*) (*англійською та українською мовами*).

ЗАГАЛЬНА ІНФОРМАЦІЯ ПРО КУРС

Анотація	Дисципліна «Архітектура та проектування програмного забезпечення (частина 1)» є навчальною дисципліною з блоку обов'язкових дисциплін спеціальної (фахової) підготовки студента рівня підготовки «бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення». Вона викладається у п'ятому семестрі в обсязі 150 год. (5 кредитів ECTS), зокрема: лекції – 32 год., лабораторні – 32 год., самостійна робота – 86 год. Підсумкова звітність з дисципліни – екзамен. Викладення дисципліни забезпечує готовність студента до участі у вирішенні питань обґрунтування вибору архітектури програмного продукту, визначення складу та структури його компонентів із врахуванням встановлених вимог до якості.
Цілі курсу	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, а також виробку навичок розробки програмних рішень із застосуванням деяких шаблонів проектування, а також виробку навичок створення моделей програмного рішення на основі шаблонів проектування в Enterprise Architect.

Формат	Лекції, лабораторні роботи, консультац. Підсумковий контроль — екзамен.						
Семестр	5						
Обсяг (кредити) / Тип курсу	5 / Обов'язкова	Лекції (години)	32	Лабораторні заняття (години)	32	Самостійна робота (години)	86
Програмні компетентності	<p>K02. Здатність застосовувати знання у практичних ситуаціях</p> <p>K05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>K15. Здатність розробляти архітектури, модулі та компоненти програмних систем</p> <p>K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.</p> <p>K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>K24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p>						
Результати навчання	<p>PR01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки</p> <p>PR03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>PR06. Уміння вибирати та використовувати методологію створення програмного забезпечення відповідну до задачі.</p> <p>PR07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>PR10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p> <p>PR11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>PR12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>PR13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>PR14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>PR15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>PR16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.</p> <p>PR17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>PR19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p> <p>PR20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>PR23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p>						
Методи викладання та	Основним методом навчання в ході лекцій є пояснювально-ілюстративний метод. Для активізації пізнавальної діяльності передбачені виступи студентів та організація дискусій за окремими питаннями лекцій.						

навчання	Виконання лабораторних робіт 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.
Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)	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) (CAS). Рівень практичних навиків оцінюється за результатами виконання лабораторних робіт (CAS). Підсумковий/семестровий контроль здійснюється під час екзамену, що передбачає письмовий опит з двох теоретичних питань та створення коду згідно завдання (FAS).

СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів для оцінювання успішності	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	Засвоєння теорії (тематика самостійної роботи)	30 балів
	90-100	A	відмінно		Відпрацювання завдань лабораторного практикуму: ЛР №1 ЛР№2-7	40 балів
	82-89	B	добре			4 бали
	74-81	C			задовільно	6 балів за кожну
	64-73	D	незадовільно з можливістю повторного складання			Екзамен
	60-63	E			незадовільно з обов'язковим повторним вивченням дисципліни	Разом за дисципліну
	35-59	FX				
	0-34	F				

Політика курсу	<p>Студенти зобов'язані відвідувати заняття згідно розкладу. У разі відсутності студента на лекції він відпрацьовує конспект лекцій до початку наступного заняття.</p> <p>Участь у лабораторних роботах передбачає необхідність повторення лекційного матеріалу та самостійного вивчення рекомендованих джерел. На початку лабораторного заняття відбувається опит студентів за матеріали лекцій та самостійної роботи.</p> <p>Виконання завдань лабораторних робіт вимагає попередньої підготовки та завчасного опрацювання всіх необхідних матеріалів для продуктивних дискусій під час заняття та їх оперативного виконання. Для отримання підсумкової оцінки з дисципліни виконання всіх лабораторних робіт обов'язкове.</p> <p>Важливим елементом підготовки є необхідність дотримання графіку представлення результатів лабораторних робіт та рефератів. За затримку виконання без підтвердженої офіційно причини оцінка знижується.</p>
-----------------------	--

СТРУКТУРА ТА ЗМІСТ КУРСУ

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). (6 год.)	Лабораторна робота 1	Дослідження процесу установки Enterprise Architect та його застосування для створення базових діаграм UML (8 год.)	Самостійна робота	Детальне вивчення стандартів, що визначають поняття «архітектура ПЗ» та її опису. Приклади застосування фреймворків для опису архітектури ПЗ.
Тема 2	Вимоги до архітектури сучасного ПЗ. Сутність використання шаблонів для забезпечення сучасних вимог до ПЗ. Класифікація шаблонів проектування (Design pattern by GoF). Типовий опис шаблонів проектування. Детальний опис популярних шаблонів проектування Шаблони розподілу відповідальностей GRASP. (14 год.)	Лабораторна робота 2	Дослідження особливостей реалізації шаблонів SINGLETON та FACTORY METHOD (4 год.)		Встановлення Enterprise Architect Створення проектів та моделей у Enterprise Architect. Панелі інструментів Enterprise Architect. Створення діаграм класів, послідовності та активності у Enterprise Architect.
Тема 3	Архітектурні стилі. Визначення архітектурного стилю. Огляд архітектурних стилів сучасності. (8 год.)	Лабораторна робота 3	Дослідження особливостей реалізації шаблону TEMPLATE METHOD (4 год.)		Призначення шаблонів проектування. Типові ситуації, переваги та недоліки використання шаблонів проектування. Принципи застосування шаблонів розподілу відповідальностей GRASP. Огляд реалізацій застосування шаблонів розподілу відповідальностей.
Тема 4	Парадигми програмування (4 год.)	Лабораторна робота 4	Дослідження особливостей реалізації шаблону PROXY (4 год.)		Детальний опис архітектурних стилів. Аналіз реалізації стилю у відомих програмних продуктах.
		Лабораторна робота 5	Дослідження особливостей реалізації шаблонів COMMAND та MEMENTO (4 год.)		Сучасні погляди на підходи до проектування програмного забезпечення
		Лабораторна робота 6	Дослідження особливостей реалізації шаблону проектування COMPOSITE (4 год.)		
		Лабораторна робота 7	Дослідження особливостей реалізації шаблону CHAIN OF		

ЛІТЕРАТУРА

Обов'язкова

- 1 Sommerville Ian. Software Engineering (6th Edition) Retrieved fom: https://www.academia.edu/6826193/Ian_Sommerville_Software_Engineering_6th_Edition.
- 2 Guide to Software Engineering Body of Knowledge (2014) (SWEBOOK), version 3.0. – IEEE Computer Society,
- 3 Лаврищева К. М. (2013) Software Engineering комп'ютерних систем. Парадигми, технології та CASE засоби програмування Київ: Наук. думка,
- 4 Bass Len, (2012) Clements Paul, Kazman Rick. Software Architecture in Practice (SEI Series in Software Engineering), 3rd Edition. - Addison-Wesley Professional,
- 5 Табунщик. Г.В., Петрова О.А., Каплієнко, Т.І. (2016) Проектування та моделювання програмного забезпечення сучасних інформаційних систем: Запоріжжя
- 6 Freeman Eric, Robson Elisabeth (2020) Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software 2nd Edition. – O'Reilly Media,
- 7 Larman Craig. (2005) Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition. – Pearson,
- 8 Clements Paul, Bachmann Felix, Bass Len, Garlan David, Ivers James, Little Reed, Merson Paulo, Nord Robert, Stafford Judith. Clements Paul. Documenting Software Architectures: Views and Beyond 2nd Edition (2010) Addison-Wesley Professional,

Додаткова

Стандарти

- 9 ISO/IEC/IEEE 12207 Systems and software engineering — Software life cycle processes.
- 10 IEEE 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology.
- 11 ISO/IEC 9126-1. Software engineering – Product quality – Part 1: Quality model.
- 12 IEEE 1471. Recommended Practice for Architectural Description of Software-Intensive Systems.
- 13 ISO/IEC/IEEE 42010. Systems and software engineering — Architecture description.

Ресурси інтернет

- 14 <https://refactoring.guru/ua/design-patterns>.
- 15 <https://sparxsystems.com>

НОРМИ АКАДЕМІЧНОЇ ЕТИКИ

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.