

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ЧАСТИНА 2)

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 2)» є навчальною дисципліною з блоку обов'язкових дисциплін спеціальної (фахової) підготовки студента рівня підготовки «бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення». Вона викладається у шостому семестрі в обсязі 90 год. (3 кредити ECTS), зокрема: лекції – 32 год., лабораторні – 32 год., самостійна робота – 26 год. Вивчення дисципліни передбачає підготовку та захист курсового проекту, в ході якого студенти здійснюють проектування програмного забезпечення. Підсумкова звітність з дисципліни – екзамен. Викладення дисципліни забезпечує розуміння студентами змісту процесу створення програмного забезпечення у IT-компаніях, а також відпрацювання базових навиків з проектування програмного забезпечення та сумісної роботи із кодом проекту, необхідних для роботи у команді розробників програмного забезпечення.
Цілі курсу	Викладення дисципліни має на меті поглиблення системи знань меті поглиблення знань про життєвий цикл програмного забезпечення та стандарти, що регламентують реалізацію його етапів, про принципи організації розробки програмного забезпечення на сучасному етапі розвитку IT-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 проекту програмного забезпечення на основі визначеного переліку вимог до програмного забезпечення та навиків застосування системи контролю версій git та репозиторію github під час створення ПЗ. Засвоєння дисципліни є важливим елементом для виконання дипломної роботи.						
Формат	Лекції, лабораторні роботи, консультації, курсовий проект. Підсумковий контроль — екзамен.						
Семестр	6						
Обсяг (кредити) / Тип курсу	3 / Обов'язкова	Лекції (години)	32	Лабораторні заняття (години)	32	Самостійна робота (години)	26
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>ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки</p> <p>ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПР06. Уміння вибирати та використовувати методологію створення програмного забезпечення відповідну до задачі.</p> <p>ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p> <p>ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПР16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.</p> <p>ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.</p>						

	<p>ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p>
Методи викладання та навчання	<p>Основним методом навчання в ході лекцій є пояснювально-ілюстративний метод. Для активізації пізнавальної діяльності передбачені виступи студентів та організація дискусій за окремими питаннями лекцій.</p> <p>Під час лабораторних робіт відбувається 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 семестру. Предметна область для проектування є індивідуальною. При розробці курсового проекту акцентується увага на те, що отримані навички важливі для дипломного проектування.</p>
Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)	<p>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).</p> <p>Перевірка рівня практичних навиків перевіряється на лабораторних роботах, що виконуються за індивідуальними варіантами (CAS).</p> <p>Підсумковий контроль здійснюється на захисті курсового проекту та під час семестрового екзамену (FAS). На екзамені передбачається письмовий опит з теоретичних питань та створення визначеної моделі за індивідуальним завданням у обмежений час.</p>

СИСТЕМА ОЦІНЮВАННЯ

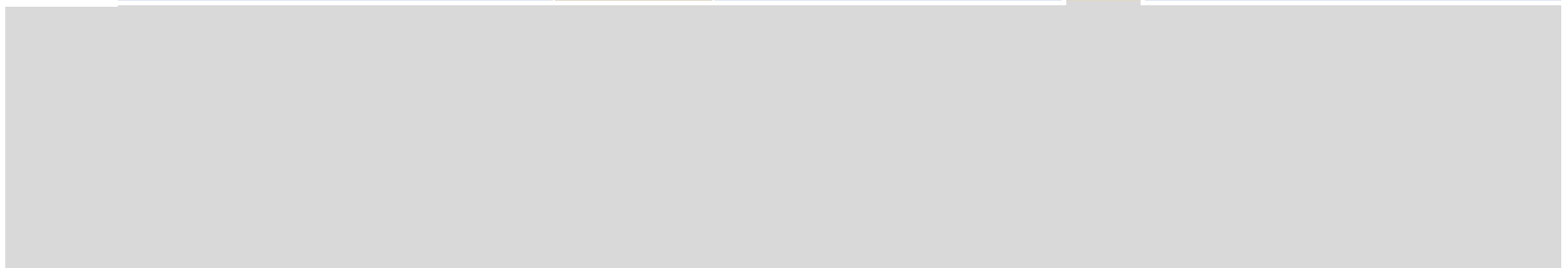
Розподіл балів для оцінювання успішності	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	Засвоєння теорії	15 балів
	90-100	A	відмінно		Практика Git	15 балів
	82-89	B	добре			
	74-81	C	задовільно		Практика з проектування ПЗ та захист курсової роботи	50 балів
	64-73	D				
	60-63	E				
	35-59	FX	незадовільно з можливістю повторного складання		Екзамен	20 балів
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		Разом за дисципліну	100 балів

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	<p>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 (6 год.)</p>	Лабораторна робота 1	<p>Встановлення та налаштування системи git. Робота в git-bash (4 год.)</p>	Самостійна робота	<p>Курсовий проект. Вибір задачі для автоматизації та визначення предметної області. Опис бізнес-процесів предметної області. Визначення функціональних та нефункціональних вимог до проектуемого програмного забезпечення. Розробка SRS для створюваного програмного забезпечення. Проектування доменної моделі об'єктів предметної області.</p>
Тема 2	<p>Проектування як складова життєвого циклу розробки програмного забезпечення. Основи визначення та нормативні документи. Нотації та представлення результатів проектування. (6 год.)</p>	Лабораторна робота 2	<p>Управління гілками з використанням git (4 год.)</p>		<p>Курсовий проект. Розробка use case діаграм та їх уточнення шляхом розробки сценаріїв або діаграм взаємодії.</p>
Тема 3	<p>UML як нотація для представлення результатів проектування. Класифікація діаграм UML. Застосування діаграм UML для опису поведінки та структури програмного забезпечення. Програмне забезпечення для створення діаграм UML. (6 год.)</p>	Лабораторна робота 3	<p>Використання репозиторію github (2 год.)</p>		<p>Курсовий проект. Проектування Entity-Relation diagram. Обґрунтований вибір системи керування БД.</p>
Тема 4	<p>Стратегії та методології розробки програмного забезпечення (10 год.)</p>	Лабораторна робота 4	<p>Формалізоване представлення бізнес-процесів та вимог до програмного забезпечення із застосуванням CASE-систем (6 год.)</p>		<p>Курсовий проект. Обґрунтований вибір архітектури програмного забезпечення. Розробка та опис структури компонентів програмного забезпечення та схеми його розгортання. Розробка components and deployment diagram.</p>

Тема 5	Проектування інтерфейсу користувачів (4 год.)	Лабораторна робота 5	Проектування поведінки системи для різних категорій користувачів на основі діаграм UML (6 год.)	
		Лабораторна робота 6	Проектування схеми для зберігання даних (6 год.)	
		Лабораторна робота 7	Проектування структури компонентів програмного забезпечення та схеми його розгортання (4 год.)	



ЛІТЕРАТУРА

Обов'язкова

- 1 Sommerville Ian. Software Engineering (6th Edition) Retrieved from: https://www.academia.edu/6826193/Ian_Sommerville_Software_Engineering_6th_Edition.
- 2 Guide to Software Engineering Body of Knowledge (SWEBOOK), version 3.0. (2014) IEEE Computer Society.
- 3 Лаврищева К. М. (2013) Software Engineering комп'ютерних систем. Парадигми, технології та CASE засоби програмування Київ: Наук. думка
- 4 Git - Documentation (git-scm.com) Retrieved from // <https://git-scm.com/doc>.
- 5 Табунцік., Г.В. Каплієнко, Т.І., Петрова. О.А (2016) Проектування та моделювання програмного забезпечення сучасних інформаційних систем: Запоріжжя.
- 6 Петрик М.Р., Петрик. О.Ю.(2015) Моделювання програмного забезпечення Тернопіль: Вид-во ТНТУ,
- 7 Rumpe Bernhard. (2016) Modeling with UML. Language, Concepts, Methods. –Springer, Retrieved from // https://nibmehub.com/opac-service/pdf/read/Modeling%20with%20UML_%20Language-%20Concepts-%20Methods.pdf .
- 8 Barry W.(2014) Boehm, Jo Ann Lane, Supannika Koolmanoijwong, Richard Turner. The Incremental Commitment Spiral Model: Principles and Practices for Successful Systems and Software. Addison-Wesley,
- 9 Kruchten Philippe.(2004) The Rational Unified Process: An Introduction: 3rd edition. Pearson.
- 10 Wiegers Karl (2013) Beatty Joy. Software Requirements, 3rd ed. Microsoft Press.
- 11 Грицюк Ю. І.(2018) Аналіз вимог до програмного забезпечення.
- 12 Піхлер Р.(2019) Agile продукт-менеджмент за допомогою Scrum. Фабула
- 13 Мартін Роберт. (2019) Чистий код. Фабула.
- 14 Clements Paul. Bachmann Felix, Bass Len, Garlan David, (2010) Documenting Software Architectures: Views and Beyond 2nd Edition. Addison-Wesley Professional.

Додаткова

Стандарти

- 15 ISO/IEC/IEEE 12207 Systems and software engineering Software life cycle processes.
- 16 IEEE 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology.
- 17 ISO/IEC 9126-1. Software engineering Product quality Part 1: Quality model.
- 18 IEEE 1471. Recommended Practice for Architectural Description of Software-Intensive Systems.
- 19 ISO/IEC/IEEE 42010. Systems and software engineering Architecture description.
- 20 IEEE Std 830. IEEE Recommended Practice for Software Requirements Specifications.
- 21 IEEE 1016. IEEE Recommended Practice for Software Design Descriptions.

Ресурси інтернет

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

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.