

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

ЗАТВЕРДЖУЮ
Тетяна ГОВОРУЩЕНКО
05 09 2024 р.



СИЛАБУС

Навчальна дисципліна **Основи інженерії програмного забезпечення**

Освітньо-професійна програма **Інженерія програмного забезпечення**

Рівень вищої освіти **перший (бакалаврський)**

Загальна інформація

Позиція	Зміст інформації
Викладач	Дьоміна Анастасія Іванівна
Профайл викладача	https://ipz.khmn.edu.ua/domina-a-i/
Е-mail викладача	DiominaAI@khmn.edu.ua
Контактний телефон	За домовленістю
Сторінка дисципліни в ІСУ	https://msn.khmn.edu.ua/course/view.php?id=3403
Навчальний рік	2024-2025
Консультації	Очні: згідно графіка, 1-201, 1-204; онлайн: за необхідністю та попередньою домовленістю

Характеристика дисципліни

Статус дисципліни	Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни		Кількість годин							Форма семестрового контролю		
				Кредити ЄКТС	Години	Аудиторні заняття				Семінарські заняття	Самостійна робота, у т.ч. ІРС	Курсовий проект	Курсова робота	Залік	Іспит
						Разом	Лекції	Лабораторні роботи	Практичні заняття						
О	Очна (денна)	1	1	5	150	72	36	36			78			+	

Анотація дисципліни

Дисципліна «Основи інженерії програмного забезпечення», як одна з фахових дисциплін, займає провідне місце у підготовці фахівців освітнього рівня «бакалавр» зі спеціальності 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення» і є теоретичним та практичним фундаментом для подальшого вивчення студентами циклу професійно-орієнтованих дисциплін. При викладанні дисципліни використовуються активні і творчі форми проведення занять, зокрема: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні заняття (з використанням методів інформаційних технологій та сучасних Case-засобів моделювання і проектування, майстер-класів, практикумів), самостійна робота студентів.

Пререквізити – офісне програмне забезпечення, основи програмування; **кореквізити** – об'єктно-орієнтоване програмування; аналіз вимог до ПЗ; моделювання та аналіз ПЗ.

Мета і завдання дисципліни

Мета дисципліни. Метою дисципліни є: навчити студентів фундаментальним основам інженерії програмного забезпечення (ПЗ) відповідно до змісту міжнародного професійного стандарту Guide to the Software Engineering Body of Knowledge (SWEBOK); сформувати в них загальне уявлення про задачі, процеси, методології, методи та засоби інформаційних технологій, які використовуються в інженерії ПЗ; ознайомити їх з основними положеннями технологій розробки ПЗ, включаючи розгляд питань, пов'язаних з аналізом, проектуванням, реалізацією, тестуванням, документуванням та супроводом програмних продуктів.

Завдання дисципліни. Формування у здобувачів вищої освіти базових теоретичних знань та практичних навичок у галузі інженерії ПЗ на всіх етапах життєвого циклу ПЗ, опанування ними необхідного апарату для подальшого вивчення циклу професійно-орієнтованих дисциплін.

Очікувані результати навчання

Відповідно до Стандарту вищої освіти із зазначеної спеціальності та освітньої програми дисципліна має забезпечити:

– **компетентності:** здатність брати участь у проектуванні ПЗ, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність формулювати та забезпечувати вимоги щодо якості ПЗ у відповідності з вимогами замовника, технічним завданням та стандартами; здатність дотримуватися специфікацій, стандартів, правил і рекомендацій у професійній галузі при реалізації процесів життєвого циклу; здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення; здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження ПЗ та визнання важливості навчання протягом всього життя; здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки ПЗ; здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності ПЗ.

– **програмні результати навчання:** аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії ПЗ і дотримуватись їх у професійній діяльності; знати основні процеси, фази та ітерації життєвого циклу ПЗ; знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії ПЗ; проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; застосовувати на практиці ефективні підходи щодо проектування ПЗ; застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування ПЗ; мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; вміти документувати та презентувати результати розробки ПЗ.

Результати навчання. Після вивчення дисципліни студент має: володіти професійною термінологією та основними поняттями з інженерії ПЗ; описувати області знань інженерії ПЗ за SWEBOOK; характеризувати моделі та процеси життєвого циклу ПЗ на основі чинних стандартів; орієнтуватись у сучасних методологіях та технологіях розробки ПЗ; використовувати методи, нотації та інструментальні засоби моделювання і проектування ПЗ; будувати структурно-функціональні, інфологічні та об'єктно-орієнтовані моделі ПЗ; оцінювати ступінь адекватності розроблених моделей; документувати прийняті рішення, виходячи безпосередньо з моделей; демонструвати культуру мислення при розробці моделей; аналізувати предметну область та визначати функціональні вимоги до ПЗ; аналізувати проблеми і тенденції розвитку інженерії ПЗ; застосовувати методи та засоби інженерії ПЗ у вивченні циклу професійно-орієнтованих дисциплін.

Тематичний і календарний план вивчення дисципліни

№ тижня	Тема лекції*	Тема лабораторного заняття*	Самостійна робота студентів		
			Зміст	Год.	Література
1	2	3	4	5	6
1	Вступ до інженерії програмного забезпечення		Опрацювання теоретичного матеріалу. Підготовка до виконання ЛР1.	4	[1], [2], [6]
2	Життєвий цикл програмного забезпечення	Моделювання на основі електронних таблиць. Аналіз моделей	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР1. Підготовка до виконання ЛР2.	5	[1], [2], [5], [6]
3	Моделі життєвого циклу в сучасних методологіях розробки програмного забезпечення		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР2. Підготовка до виконання ЛР3.	4	[1], [2], [5]
4	Структурно-функціональний аналіз і моделювання. Методологія SADT	Структурно-функціональне моделювання за методологією SADT (IDEF0)	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР2. Підготовка до виконання ЛР3.	5	[3]
5	Аналіз і моделювання потоків даних		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР3. Підготовка до виконання ЛР4.	4	[3], [5], [6]
6	Інфологічне моделювання	Аналіз і моделювання потоків даних. Декомпозиція процесів у нотації IDEF	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР3. Підготовка до виконання ЛР4.	4	[3], [6]
7	Уніфікована мова моделювання UML. Діаграми варіантів використання		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5. Підготовка до тестового контролю з тем 1-2.	5	[4 – 6]
8	Уніфікована мова моделювання UML. Деталізація варіантів використання	Інфологічне моделювання. Побудова діаграм «сутність-зв'язок»	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР4. Підготовка до виконання ЛР5. Підготовка до тестового контролю з тем 1-2.	5	[4 – 6]

1	2	3	4	5	6
9	Уніфікована мова моделювання UML. Документування сценаріїв використання за допомогою діаграм взаємодії. Діаграми класів		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР5. Підготовка до виконання ЛР6.	4	[4 – 6]
10	Процес розробки вимог до програмного забезпечення	Аналіз вимог до ПЗ. Побудова UML-діаграм варіантів використання	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР5. Підготовка до виконання ЛР6.	4	[1], [2], [5]
11	Джерела та стратегії виявлення вимог до програмного забезпечення		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР6. Підготовка до виконання ЛР7.	4	[1], [2], [5]
12	Аналіз, систематизація та специфікація вимог до програмного забезпечення	Аналіз вимог до ПЗ. Побудова UML-діаграм станів та діяльності	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР6. Підготовка до виконання ЛР7.	4	[1], [2], [5]
13	Основи проектування програмного забезпечення		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8.	4	[1], [2], [5], [6]
14	Основи конструювання програмного забезпечення	Аналіз вимог до ПЗ. Побудова UML-діаграм взаємодії (послідовності та кооперації)	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР7. Підготовка до виконання ЛР8.	4	[1], [2], [5]
15	Основи тестування та якості програмного забезпечення		Опрацювання теоретичного матеріалу. Підготовка до тестового контролю з тем 3-5.	5	[1], [2], [5]
16	Документування програмного забезпечення	Проектування класів. Розробка UML-діаграм реалізації.	Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР8. Підготовка до тестового контролю з тем 3-5.	5	[1],
17	Управління конфігурацією в життєвому циклі ПЗ. Впровадження, супровід та моніторинг ПЗ		Опрацювання теоретичного матеріалу. Підготовка до захисту ЛР8.	4	[1], [5]
18	Етика професійної діяльності розробника програмного забезпечення	Проектування класів. Розробка UML-діаграм реалізації.	Опрацювання теоретичного матеріалу.	4	[2], [5]

* Лекції проводяться щотижня по дві години; лабораторні заняття – один раз у два тижні по чотири години (для підгрупи). Послідовність проведення занять визначається розкладом (може не відповідати нумерованим тижням).

Скорочення: ЛР – лабораторна робота.

Політика дисципліни

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати всі заняття згідно з розкладом, не запізнюватися на заняття, всі види робіт виконувати якісно і відповідно до графіка. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи

занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати не пізніше ніж за два тижні до кінця теоретичних занять у семестрі.

Лабораторні роботи виконуються індивідуально, згідно з варіантами, що представлені у методичних вказівках до лабораторних робіт. Під час роботи над індивідуальними завданнями недопустимі порушення правил академічної доброчесності. У разі наявності плагіату (спроба представити до захисту лабораторну роботу іншого варіанту) здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати лабораторну роботу згідно з його варіантом.

Здобувачі вищої освіти при вивченні дисципліни можуть користуватись як наявним в аудиторіях кафедри комп'ютерним обладнанням, так і власними пристроями (ноутбуками, планшетами, смартфонами).

Набутті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ:

<https://khmnu.edu.ua/wp-content/uploads/normatyvni-dokumenty/polozhennya/pro-poryadok-vyznannya-ta-perezarahuvannya-rezultativ-navchannya.pdf>

Критерії оцінювання результатів навчання

Оцінювання академічних досягнень студента здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». Кожний вид роботи з дисципліни оцінюється за **чотирибальною** шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих **позитивно** з урахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих видів її робіт.

Поточний контроль здійснюється під час лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком навчального процесу. Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання роботи; знання теоретичного матеріалу з теми; якість оформлення звіту з лабораторної роботи; вільне володіння студентом спеціальною термінологією і уміння професійно обґрунтувати прийняті рішення; своєчасний захист лабораторної роботи.

При оцінюванні знань студентів використовуються різні методи контролю: усне опитування перед допуском до лабораторного заняття; захист лабораторних робіт; тестовий контроль теоретичного матеріалу з теми; презентація індивідуальних завдань.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів у семестрі за ваговими коефіцієнтами

Аудиторна робота	Самостійна робота		Семестровий контроль
Лабораторні роботи	Тестовий контроль		
№ 1 – 8	ТК1 (Т1-2)	ТК2 (Т3-5)	За рейтингом
ВК: 0,5	0,5		

Умовні позначення: ВК – ваговий коефіцієнт, ТК – тестовий контроль, Т – тема дисципліни.

Оцінювання тестових завдань

Тематичний тест для кожного студента складається з 20 тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20. Тестові завдання для кожного студента генеруються випадковим чином із загального банку питань у

середовищі для навчання Moodle. Оцінювання відповідей студента здійснюється в автоматичному режимі. Оцінювання здійснюється за чотирибальною шкалою. Сума балів пропорційна кількості правильних відповідей. Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестові завдання	1 – 11	12 – 14	15 – 18	19 – 20
Оцінка за 4-бальною шкалою	2	3	4	5

На тестування відводиться 25 хвилин.

При отриманні негативної оцінки тест слід перездати до терміну наступного контролю.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені у таблиці.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Вітчизняна оцінка, критерії	
A	4,75-5,00	Зараховано	ВІДМІННО – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25-4,74		ДОБРЕ – повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24		ДОБРЕ – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25-3,74		ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24		ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00 -2,99	Незараховано	НЕЗАДОВІЛЬНО – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1, 99		НЕЗАДОВІЛЬНО – необхідна серйозна подальша робота і повторне вивчення дисципліни

Питання для підсумкового контролю з дисципліни

1. Поняття програми, програмного засобу, програмного забезпечення. Види ПЗ за типами ліцензій.
2. Причини та передумови виникнення дисципліни «Інженерія ПЗ». Визначення інженерії ПЗ.
3. Інженерія ПЗ як інженерна та наукова дисципліна.
4. Стандартизація в інженерії ПЗ.
5. Основні області знань SWEBOOK.
6. Життєвий цикл програмного забезпечення.
7. Загальна характеристика каскадної моделі життєвого циклу ПЗ; її переваги і недоліки.
8. Загальна характеристика спіральної моделі життєвого циклу ПЗ та її переваги і недоліки.
9. Загальна характеристика ітераційної моделі життєвого циклу ПЗ та її переваги і недоліки.
10. Загальна характеристика інкрементної (покрокової) моделі життєвого циклу ПЗ та її переваги і недоліки.

11. Загальна характеристика методології швидкої розробки додатків RAD (Rapid Application Development) та її переваги і недоліки.
12. Методологія RUP (Rational Unified Process). Загальна характеристика процесу розробки ПЗ.
13. Модель проектної групи MSF (Microsoft Solutions Framework).
14. Модель процесів MSF (Microsoft Solutions Framework).
15. Гучкі методології розробки ПЗ. Загальна характеристика.
16. Екстремальне програмування (XP).
17. Методологія Scrum.
18. Міжнародні та національні стандарти життєвого циклу ПЗ.
19. Сутність та методи структурного аналізу і моделювання.
20. Загальна характеристика методологій структурного аналізу і моделювання.
21. Основні принципи структурного аналізу і моделювання.
22. Нотація IDEF0 для графічного представлення SADT-моделі.
23. Синтаксис і семантика DFD-діаграм.
24. Базові поняття інформаційного моделювання «сутність-зв'язок» (ERD).
25. Огляд нотацій, що використовуються при побудові ER-діаграм.
26. Методика побудови ER-діаграм.
27. Сутність об'єктно-орієнтованого аналізу і моделювання; його переваги і недоліки.
28. Основні принципи об'єктно-орієнтованого аналізу і моделювання.
29. Загальна характеристика мови UML. Діаграми UML.
30. Загальний процес розробки вимог до ПЗ.
31. Властивості вимог до ПЗ.
32. Джерела та стратегії виявлення вимог до ПЗ.
33. Аналіз, систематизація та перевірка вимог до ПЗ.
34. Специфікація та формалізація вимог до ПЗ.
35. Технічне завдання на ПЗ.
36. Поняття архітектури ПЗ. Основні класи архітектур.
37. Основні принципи проектування архітектури ПЗ.
38. Модульна структура програмного продукту. Основні характеристики модулів.
39. Типи програмних модулів та їх структура.
40. Низхідне («зверху-донизу») та висхідне («знизу-доверху») проектування ПЗ.
41. Розробка структури ПЗ при об'єктному підході.
42. Принципи, правила та етапи проектування інтерфейсу ПЗ.
43. Парадигми програмування.
44. Поняття мови програмування. Етапи розвитку мов програмування.
45. Класифікація мов програмування.
46. Інструментальні засоби та інтегровані середовища програмування. Транслятори.
47. Основи конструювання ПЗ. Стандарти в конструюванні. Управління конструюванням.
48. Класифікація програмних помилок. Методи відладки ПЗ. Загальна методика відладки ПЗ.
49. Основні види тестування.
50. Рівні тестування (модульне, інтеграційне та системне тестування).
51. Тести. Класифікація тестів. Валідаційні тести. Вимірювання результатів тестування.
52. Особливості об'єктно-орієнтованого тестування.
53. Особливості тестування об'єктно-орієнтованих модулів.
54. Верифікація та атестація ПЗ.
55. Загальні принципи складання та оформлення документації ПЗ.
56. Основні види програмної документації.
57. Характеристика документації, призначеної для користувача ПЗ.
58. Електронна довідкова система.
59. Процеси управління конфігурацією програмних засобів.
60. Етапи і процедури при управлінні конфігурацією ПЗ.
61. Технологічне забезпечення при управлінні конфігурацією ПЗ.

62. Основні етапи готовності програмного продукту. Альфа-версія. Бета-версія. Реліз.
63. Організація і методи супроводу ПЗ. Етапи і процедури при супроводі ПЗ.
64. Техніки супроводу. Реінжинірінг, зворотній реінжинірінг.
65. Кодекс етичних норм професіонала в області інженерії ПЗ (IEEE-CS/ACM).
66. Корпоративна етика.

Методичне забезпечення

Навчальний процес з дисципліни повністю і в достатній кількості забезпечений необхідними навчально-методичними розробками у Модульному навчальному середовищі. Доступ до ресурсу: <https://msn.khmnmu.edu.ua/course/view.php?id=3403>.

Рекомендована література

1. Бородкіна І. Інженерія програмного забезпечення. Посіб. для студентів вищих навч. закладів / І. Бородкіна, Г. Бородкін. – К: Вид-тво "Центр навчальної літератури", 2018. – 204 с.
2. Левус Є. Вступ до інженерії програмного забезпечення / Є. Левус, Н. Мельник. – Л.: Львівська політехніка, 2018. – 248 с.
3. Крижановський Є. М. Моделювання бізнес-процесів та управління ІТ-проектами: навч. посіб. / Є. М. Крижановський, А. Р. Яшолт, С. О. Жуков, О. М. Козачко – Вінниця: ВНТУ, 2018. – 91 с.
4. Постіл С. Д. UML. Уніфікована мова моделювання інформаційних систем: навч. посіб. / С. Д. Постіл. – Ірпінь: УДФСУ, 2019. – 322 с.
5. Rod Stephens. Beginning Software Engineering: Second Edition. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada and the United Kingdom. 2022. 685 P.
6. Elvis C. Foster. Software Engineering. A Methodical Approach: Second Edition. Published 2022 by CRC Press, Taylor & Francis Group. Boca Raton-London-New York. 579 P.

Розробник



асистент Дьоміна А. І.

Погоджено:

Гарант ОПП «ІПЗ»



д-р фіз.-мат. наук, проф. Бедратюк П. П.

Зав. кафедри ІПЗ



д-р фіз.-мат. наук, проф. Бедратюк П. П.