

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій  
Кафедри інженерії програмного забезпечення

Декан ФІТ \_\_\_\_\_  
ЗАТВЕРДЖУЮ  
Говорунченко Т.О.  
1 вересня 2024 р.



## СИЛАБУС

Навчальна дисципліна **Методології та технології розробки програмних систем**  
Освітньо-наукова програма **Інженерія програмного забезпечення**  
Рівень вищої освіти **Другий (магістратура)**

Позиція	Зміст інформації
Викладач(і)	Праворська Наталія Іванівна
Профайл викладача	<a href="http://ipz.khnu.km.ua/праворська-н-і/">http://ipz.khnu.km.ua/праворська-н-і/</a>
E-mail викладача	<a href="mailto:margana2000007@gmail.com">margana2000007@gmail.com</a>
Контактний телефон	Заповнюється за домовленістю
Сторінка дисципліни в ІСУ	<a href="https://msn.khnu.km.ua/course/view.php?id=8110">https://msn.khnu.km.ua/course/view.php?id=8110</a>
Навчальний рік	2024-2025
Консультації	Очні: вівторок 6 пара, 1-210 Он-лайн: за необхідністю і попередньою домовленістю

### Загальна характеристика дисципліни

Статус дисципліни	Форма навчання	Курс	Семестр	Загальний обсяг		Кількість годин				Курсовий проект	Курсова робота	Форма семетрового контролю	
				Європ. кредит	Години	Аудиторні заняття			Самостійна робота			Залік	Іспит
						Всього	Лекції	Лабораторні роботи					
3	Д	1	1	5	150	51	17	34	99	х			х

### Анотація навчальної дисципліни

Дисципліна “Методології та технології розробки програмних систем” є дисципліною вільного вибору студента в галузі інженерії програмного забезпечення. Студент, який успішно завершив вивчення дисципліни, має: досконало володіти професійною термінологією та основними поняттями ПС; розуміти основні етапи попередніх досліджень, визначення цілей продукту, що розробляється, зарубіжні та вітчизняні стандарти оформлення специфікації до ПС, мови специфікації вимог, проектувати ІТ-продукти, знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення, виявляти інформаційні потреби і класифікувати дані для проектування програмних систем, розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати

варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів.

Дисципліна викладається для здобувачів другого (магістерського) рівня вищої освіти денної форми навчання спеціальностей галузі інформаційних технологій. При викладанні дисципліни використовуються активні і творчі форми проведення занять, зокрема, методи проблемного навчання.

### *Мета і завдання дисципліни*

**Мета дисципліни.** Вивчення основних шляхів організації і виконання успішних проектів в області розробки програмного забезпечення. В навчальному курсі розглянуто технологічні основи процесу розробки програмного забезпечення. Галузь розробки програмного забезпечення – одна з наймолодших і перспективних галузей людської діяльності. В даний час складність завдань, що стоять перед галуззю, безперервно зростає, а це вимагає застосування спеціальних технологій організації і проведення процесу розробки програмного забезпечення. Теоретичне вивчення подібних технологій, суміщене з їх практичним застосуванням, є важливою частиною підготовки фахівця в області розробки ПП.

**Завдання дисципліни.** Отримання студентами знань, умінь і навичок в області застосування програмних систем, оволодіння дослідницькими навичками, уміння працювати самостійно, здатністю породжувати нові ідеї (креативністю), оволодіння основними методами, способами й засобами одержання, зберігання, переробки інформації з використанням комп'ютерної техніки, уміннями аналізувати й оцінювати зібрані дані.

У рамках дисципліни “Методології та технології розробки програмних систем” вивчається процес конструювання програмних систем, розповсюдженні методики і практики побудови надійного програмного забезпечення.

### *Очікувані результати навчання*

Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити: **компетентності:** здатність знаходити, обробляти та аналізувати інформацію з різних джерел; здатність до проектування ІТ-продуктів та методик професійного навчання; здатність до застосування фундаментальних знань і вмінь до розв'язання професійних задач.

**Програмні результати навчання:** застосовувати відповідне програмне забезпечення виробничого та освітнього призначення; обирати технології для створення ІТ-продуктів та методик професійного навчання; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів; аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення.

Студент, який успішно завершив вивчення дисципліни, повинен: знати і вміло застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення; оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу; будувати і досліджувати моделі інформаційних процесів у прикладній області; виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення; розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення; розробляти і оцінювати стратегії проектування програмних засобів; розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника програмного забезпечення; конфігурувати програмне забезпечення, керувати його змінами та

розробленням програмної документації на всіх етапах життєвого циклу; здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника.

**Тематичний і календарний план вивчення дисципліни**

№ тижня	Тема лекції	Тема лабораторного заняття	Самостійна робота студента		
			Зміст	Год.	Література
1	2	3	4	5	6
1	<b>Тема 1.</b> Процес розробки програмних систем Бізнес і IT-проекти. Ринок ПЗ.	Планування процесу розробки ПС	Бізнес і IT-проекти. Ринок ПЗ.	7	1; 12; 20
2			Причини невдач IT-проектів. Виникнення Технології розробки ПЗ.	8	19; 22
3	<b>Тема 2.</b> Міжнародні та національні стандарти розробки складних програмних продуктів	Організація розробки ПС	Міжнародний стандарт ISO / ІЕС 12207: 1995-08-01. Стандарти комплексу ГОСТ 1934. Стандарти комплексу ГОСТ 19.	8	2; 4-6;
4			Забезпечення надійності розробки ПЗ. Складність програмної системи.	8	10-11
5	<b>Тема 3.</b> Організація технологічного процесу розробки ПЗ. Підстави аналізу вимог. Встановлення вимог Принципи специфікації вимог.	Аналіз вимог та моделювання Розробка та специфікація вимог	Системні послуги. Системні обмеження. Специфікація вимог. Специфікації станів. Моделювання класів. Виявлення класів.	8	3; 7; 27
6			Підхід з урахуванням використання іменних груп. Підхід з урахуванням використання загальних шаблонів класів. Підхід з урахуванням використання прецедентів. Комплексний підхід	8	8; 16-18
7	<b>Тема 4.</b> Оцінка програмного проекту по СОСОМО	Об'єктно-орієнтований аналіз систем	Розмірно-орієнтовані метрики. Функціонально-орієнтовані метрики.	10	10-11; 28
8			Оцінка проекту на основі LOC- і FP-метрик.	10	9; 22; 25

1	2	3	4	5	6
9	<b>Тема 5.</b> Системне проектування. Проектування баз даних.	Побудова тимчасової діаграми етапів робіт для заданого процесу	Відображення об'єктів у базі даних. Модель об'єктної бази даних. Об'єктно-реляційна модель бази даних	10	5-6; 26
10			Модель реляційної бази даних. Елементарні типи моделі РБД. Реляційні таблиці.	10	20; 22
11	<b>Тема 6.</b> Проектування програмної системи	Побудова тимчасової діаграми етапів робіт для заданого процесу	Закон Деметра. Методи відкриття доступу. Динамічна класифікація та зв'язність класів зі змішаними екземплярами. Проектування клієнт-серверних кооперативних взаємодій.	10	22; 24
12			Збереження процедур. Тригери. Проектування транзакцій. Короткі транзакції. Рівень ізольованості. Автоматичне відновлення. Програмоване відновлення. Точка збереження. Тригерний відкат.	10	21; 28; 30
13	<b>Тема 7.</b> Побудова архітектури ПП. Розробка програм, модульне програмування	Розробка проектної моделі	Взаємодія між підсистемами і архітектурні функції.	10	25; 29
14			Контроль архітектури програмних засобів.	10	26; 29
15	<b>Тема 8.</b> Тестування і відлагодження ПС. Тестування та управління змінами. Види тестування	Розробка проектної моделі	Тестування стосовно програмного коду. Тестування системних обмежень.	10	24; 27
16			Тестування інтерфейсу користувача. Тестування бази даних. Тестування авторизації.	10	25; 29
17	Підготовка до іспиту		Тестування інших обмежень. Документація з тестування та управління змінами.	10	19; 21-22; 29

**Примітка:** \*лекції, лабораторні роботи проводяться раз в 2 тижні або по дві або чотири години відповідно.

### ***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 лекції, лабораторні заняття згідно з розкладом, не запізнюватися на заняття, завдання виконувати відповідно до графіка. Пропущене лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До лабораторних занять студент має підготуватися за відповідною темою і проявляти активність. Набуті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (<http://khnu.km.ua/root/files/01/06/03/006.pdf>).

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене лабораторне заняття студент зобов'язаний відпрацювати в лабораторіях кафедри у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Здобувачі вищої освіти при вивченні дисципліни можуть користуватись як наявним в аудиторіях кафедри комп'ютерним обладнанням, так і власними пристроями (ноутбуками, планшетами, смартфонами). Власними пристроями можна користуватися як для роботи в системі Moodle, так і для доступу до зовнішніх інформаційних ресурсів, які необхідні для виконання лабораторних робіт та пов'язаних із ними, власних завдань магістерської роботи.

Лабораторні роботи виконуються індивідуально або групами, згідно з варіантами, що представлені у методичних вказівках до лабораторних робіт. Під час роботи над індивідуальними завданнями не допустим і 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 із його варіантом.

### ***Критерії оцінювання результатів навчання.***

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 лекційних та 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 лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення протоколу; вільне володіння студентом спеціальною термінологією і уміння професійно обґрунтувати прийняті рішення; своєчасний захист лабораторної роботи.

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

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контрольного заходу, який проводиться тестуванням з усього матеріалу дисципліни. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається невстигаючим. Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється тестуванням.

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

Аудиторна робота		Семестровий контроль, іспит (І)
Лабораторні роботи (ЛР) №	Тест (Т)	Підсумковий контрольний захід
ВК*	0,3	0,4

**Умовні позначення:** ВК – ваговий коефіцієнт, ЛР – лабораторна робота, Т – тест, І – іспит.

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у таблиці.

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20. На тестування відводиться 20 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

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

**Розподіл балів, що присвоюються студентам, з курсової роботи:**

Вагові коефіцієнти	Пояснювальна записка	Доповідь	Програмний продукт
	<b>0,3</b>	<b>0,3</b>	<b>0,4</b>

**Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС**

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Вітчизняна оцінка, критерії		Зараховано
A	4,75–5,00	5	<b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків	
B	4,25–4,74	4	<b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками	
C	3,75–4,24	4	<b>Добре</b> – в загальному правильна відповідь з двома-трьома суттєвими помилками	
D	3,25–3,74	3	<b>Задовільно</b> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією	
E	3,00–3,24	3	<b>Задовільно</b> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання	

FX	2,00–2,99	2	<i>Незадовільно</i> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни	Незараховано
F	0,00–1,99	2	<i>Незадовільно</i> – необхідна серйозна подальша робота і повторне вивчення дисципліни	

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

### **Питання для підсумкового контролю з дисципліни “ Методології та технології розробки програмних систем ”**

1. Планування розробки системи
2. Планування проекту протягом життєвого циклу ПЗ
3. Підхід SWOT
4. Проектування клієнт-серверних кооперативних взаємодій
5. Проектування транзакцій
6. Програмування баз даних
7. Підхід на основі використання прецедентів
8. Підхід на основі використання іменних груп
9. Підхід на основі використання загальних шаблонів для класів.
10. Прототипування
11. Принципи встановлення вимог
12. Поясніть різницю між етапами визначення вимог і розробки специфікації.
13. Поясніть взаємозв'язок двох етапів проектування (архітектурне проектування і деталізоване проектування) з першими двома етапами життєвого циклу – етапом визначення вимог і етапом розробки специфікації.
14. Поясніть, у чому складається розходження між розподіленою системою обробки і розподіленою системою баз даних.
15. Поясніть, у чому полягають основні розходження чотирьох підходів до виявлення класів.
16. CASE-засоби та удосконалювання процесу
17. Системи для трьох рівнів керування
18. Стратегія повторного використання.
19. Специфікації вимог
20. Специфікації станів
21. Спільна розробка додатків (JAD-метод).
22. Системні сервіси
23. Суб'єкти
24. Етапи життєвого циклу програмного забезпечення.
25. Етап встановлення вимог
26. Етап специфікації вимог.
27. Етап проектування архітектури
28. Етап деталізованого проектування.
29. Етап реалізації
30. Елементарні типи моделі РБД.
31. Основи об'єктної технології
32. Об'єктно-реляційна модель бази даних
33. Опишіть види блокувань при песимістичному керуванні паралельністю.
34. Моделювання аналізу
35. Модель об'єктної бази даних.
36. Моделі даних
37. Моделювання класів.
38. Моделювання прецедентів

39. Моделювання видів діяльності
40. Динамічна класифікація і зв'язаність класів зі змішаними екземплярами
41. Діаграми прецедентів та документування прецедентів
42. Діаграма видів діяльності
43. Динамічна класифікація і зв'язаність класів зі змішаними екземплярами
44. Зв'язність і ув'язування класів
45. Збережені процедури.
46. Закон Деметра.
47. Реляційні таблиці
48. Рівень постійних об'єктів бази даних
49. Реалізація Web-додатків
50. Рівні BCED
51. Реінжиніринг бізнес-процесів (BPR) проводить ясне розходження між бізнес-процесом і бізнес-функцією. У чому полягає це розходження?
52. Виявлення вимог
53. Відображення об'єктів у базу даних
54. Тестування відносно специфікації
55. Триланкова архітектура
56. Традиційні методи виявлення вимог
57. Комплексний підхід
58. Коротко опишіть види блокувань при песимістичному керуванні паралельністю.
59. Коротко опишіть п'ять рівнів SQL-інтерфейсів.
60. Архітектура програмного забезпечення
61. Що таке точка збереження? Як її можна використовувати при проектуванні програми?
62. Що таке відношення з'єднання?
63. Що таке триланкова архітектура? У чому її переваги і недоліки?
64. Що таке доміантний клас?
65. Які дії можливі у відповідь на відправлений запит на зміни?
66. Які об'єкти можуть виступати як цільові об'єкти для повідомлень відповідно до закону Деметра?
67. Який вплив на проектування роблять принципи, зв'язані зі зв'язаністю та ув'язуванням?
68. Які розходження між об'єктною і реляційною моделлю БД?
69. Які основні причини зрушення від структурного підходу до проектування об'єктно-орієнтованого?
70. Які принципи встановлення вимог?
71. У чому перевага виклику з клієнтської програми збереженої процедури в порівнянні з SQL-запитом, що пересилається базі даних? Чи існують ситуації, при яких ми змушені використовувати SQL-запит замість виклику вилученої процедури?
72. У чому полягає сутність підходу на основі використання загальних шаблонів для класів?
73. Чому розуміння методу ISA (архітектура інформаційної системи) важливо для системної розробки?

### **Приблизна тематика курсових проектів з дисципліни “ Методології та технології розробки програмних систем ”**

1. Проект програмної системи СУБД із захистом даних методом маскування
2. Проект програмної системи API графічного користувальницького інтерфейсу Windows. Розробка базових примітивів і компонентів
3. Проект програмної системи драйвера SQL-запитів до СУБД
4. Проект програмної системи API графічного користувальницького інтерфейсу Windows. Розробка елементів керування
5. Проект програмної системи Internet API для PIC – мікроконтролерів
6. Проект програмної системи інтерпретатора мови C (C#, Java, JavaScript)
7. Проект програмної системи безпеки котеджу
8. Проект програмної системи керування відеоспостереженням
9. Проект програмної системи керування даними в локальному кластері



10. Проект програмної системи дистанційного моніторингу температурного режиму віддалених об'єктів

### **Методичне забезпечення**

Навчальний процес з дисципліни забезпечений необхідними навчально-методичними розробками в модульному середовищі.

### **РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

#### **ОСНОВНА:**

1. Бородкіна І., Бородкін Г.. Інженерія програмного забезпечення. – Центр навчальної літератури. – 2018.- 204 с.
2. Матвеева Л.С. Процес розробки програмного забезпечення. Від теорії до практики / Л.С.Матвеева, В.А. Волков. –К. ТОВ «Інформаційні програмні системи»,2012 – 117с.
3. Перевозчикова О.Л. Інформаційні системи і структури даних. – Києво-Могилянська академія. – 2011. -287 с.
4. Grady Booch. Object-Oriented Analysis and Design with Applications. Third Edition – Addison Wesley – 2007 – 717р.
5. Van Veendabl E. Standard glossary of term used in Software testing / E. Van Veendabl. - ISTQB. - 2007. Vol. 1,2. - June. - 30 p.
6. Creting a software engineering culture / K. Wiegers // Dorset House Publishing. - New York, 2003. - 358 p.
7. Джефф Сазерленд. Scrum. Навчись робити вдвічі більше за менший час. – Клуб Сімейного Дозвілля. 2022 – 280 р.
8. G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Reference Manual - Second Edition, Addison-Wesley, 2004.
10. ДСТУ ISO 9000:2007. Системи управління якістю. Основні положення та словник термінів. – К.: Держспоживстандарт, 2008. – [Чинний від 2008-01-01] – 35 с.– (Державний стандарт).
11. ДСТУ ISO 9001:2009. Системи управління якістю. Вимоги. [Текст]: – К.: Держспоживстандарт, 2009. – [Чинний від 2009-06-22] – 80 с.– (Державний стандарт).
12. Sidorov N.A. Software Stylistics / N. A. Sidorov // Proceedings of NAU. - 2015. - 2(24), - P. 98 - 103.
13. Van Veendabl E. Standard glossary of term used in Software testing / E. Van Veendabl. - ISTQB. - 2010. Vol. 1,2. - June. - 30 p.
14. 1846A: Microsoft Solutions Framework Essentials. Microsoft Official Course, 2002
15. 2710B: Analyzing Requirements and Defining Microsoft .NET Solutions Architecture. Microsoft Official Course, 2003
16. MSF Process Model. White paper, 2002 Microsoft Corporation.
17. MSF Risk Management Discipline. White paper, 2002 Microsoft Corporation.
18. MSF Team Model. White paper, 2002 Microsoft Corporation.

#### **ДОПОМІЖНА:**

19. Steve Resnick, Aaron Bjork, Michael de la Maza. Professional Scrum with Team Foundation Server 2010. – Wrox- 2011 - 211 p.
20. Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2011. - 108 с.
21. J.D. Meier, Jason Taylor, Alex Mackman, Prashant Bansode, Kevin Jones. Team Development with Visual Studio Team Foundation Server. - Microsoft Corporation.-2014.-496 p.
22. Len Bass, Paul Clements, Rick Kazman. Software Architecture in Practice (SEI Series in Software Engineering) 3rd Edition. – 2013
23. Microsoft Visual Studio Licensing. 2019. - Microsoft Corporation–34 p.
24. Charles Petzold. Code: The Hidden Language of Computer Hardware and Software (Dv- Undefined) – 2000-
25. Michael C. Feathers, Prentice Hall. Working Effectively with Legacy Code – 2013.
26. Braude Eric J. Software Design: From Programming to Architecture Gebundene Ausgabe – Illustriert, 14. Februar 2013

27. Computing Curricula 2005. The Overview Report. [Електронний ресурс]. – Режим доступу: [www.acm.org/education/curricula.html](http://www.acm.org/education/curricula.html).
28. Systems and software engineering – Software Life Cycle Processes. ISO 12207:2008. – [Чинний від 2008-02-01] – II, 122 с.– (Міжнародний стандарт).
29. IEEE Standard Glossary of Software Engineering Terminology, Глосарій. IEEE Std 610.12-1990. – (Галузевий стандарт).

#### ІНФОРМАЦІЙНІ РЕСУРСИ

30. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khnu.km.ua>.
31. Електронна бібліотека університету . Доступ до ресурсу: [http://lib.khnu.km.ua/asp/php\\_f/p1age\\_lib.php](http://lib.khnu.km.ua/asp/php_f/p1age_lib.php).
32. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khnu.km.ua/jspui/?locale=uk>.

Розробник: Праворська Н.І.

Погоджено:  
Зав. каф. ПЗ Бедратюк Л.П.

Гарант ОНП Яшина О.М.