



МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ УНІВЕРСИТЕТ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ЄВРОПЕЙСЬКА ШКОЛА БІЗНЕСУ»

ЗАТВЕРДЖЕНО

Директор ННІ

«Європейська школа бізнесу»

Юлія РЕМИГА

від «13» 12 2023 р.

М.П.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ**

Рівень вищої освіти

перший (бакалаврський)

(назва)

Спеціальність

121 Інженерія програмного забезпечення

(код і назва спеціальності)

Освітня програма

«Інженерія програмного забезпечення»

(назва освітньої програми)

Робоча програма навчальної дисципліни **«Моделювання та аналіз програмного забезпечення»** складена на основі освітньо-професійної програми «Інженерія програмного забезпечення» для першого (бакалаврського рівня спеціальності 121 «Інженерія програмного забезпечення»), затвердженої Вченою радою Університету «30» травня 2023 року, протокол №4.

Укладач програми: Шевчук Борис Вікторович – кандидат педагогічних наук, доцент

Рецензент: к.ф.м. наук, доцент Шерман З.О.

Гарант освітньої програми:  Олександр НЕСТЕРЕНКО

Робочу програму навчальної дисципліни розглянуто та схвалено на засіданні кафедри інформаційних технологій, протокол №5 від «12 грудня 2023 р.

Розглянуто і схвалено Вченою радою Навчально-наукового інституту «Європейська школа бізнесу», протокол № 3 від «13» грудня 2023 р.

ВСТУП

Програма вивчення навчальної дисципліни «Моделювання та аналіз програмного забезпечення» складена відповідно до Стандарту вищої освіти України (далі – Стандарт) галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення».

Опис навчальної дисципліни (анотація). Дана навчальна дисципліна є однією з дисциплін фахової підготовки майбутніх розробників програмного забезпечення.

Таблиця 1

Найменування показників	Галузь знань, напрям підготовки, освітній рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 4	Галузь знань, 12 «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»	Нормативна	
Розділів – 1	Спеціальність: 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	Рік підготовки	
Змістових розділів – 2		2024	
Індивідуальне науково-дослідне завдання:		Семестр	
		6	6
Загальна кількість годин – 120		Лекції	
		12	4
Тижневе навантаження: аудиторних – 4 самостійної роботи студента – 6	Практично-лабораторні		
	20	4	
	Самостійна робота		
	118	142	
	Освітній рівень: бакалавр	Вид контролю:	
		екзамен	екзамен

Предметом вивчення навчальної дисципліни «Моделювання та аналіз програмного забезпечення» є здобування теоретичних знань та практичних навичок, щодо методології розроблення програмного забезпечення та об'єктно-орієнтованого моделювання.

Міждисциплінарні зв'язки: «Архітектура та проектування програмного забезпечення», «Основи програмування», «Об'єктно-орієнтоване програмування».

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** навчальної дисципліни «Моделювання та аналіз програмного забезпечення» є дослідження методик об'єктно-орієнтованого аналізу і графічних засобів мови UML та набуття практичних навичок щодо призначення і побудови основних видів канонічних діаграм.

1.2. **Основними завданнями** вивчення навчальної дисципліни «Моделювання та аналіз програмного забезпечення» є: отримання студентом компетенцій об'єктно-орієнтованого аналізу та моделювання на візуальній мові

UML для того, щоб приймати участь у сумісній розробці, супроводженні та проектуванні, документуванні об'єктно-орієнтованих програмних систем. Даний курс має, своєю основною метою, навчити студентів використовувати базові діаграми UML та за допомогою CASE-засобів навчитися застосовувати ці знання на всіх етапах концептуального, логічного та фізичного проектування архітектури програмних додатків.

1.3. **Компетентності та результати навчання**, формуванню яких сприяє дисципліна (взаємозв'язок з нормативним змістом підготовки здобувачів вищої освіти, сформульованим у термінах результатів навчання у Стандарті).

Згідно з вимогами стандарту дисципліна забезпечує набуття студентами **компетентностей**:

Таблиця 2

Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов у сфері розробки програмного забезпечення.
Спеціальні (фахові) компетентності	СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. СК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. СК14. Здатність до алгоритмічного та логічного мислення.

Деталізація компетентностей відповідно до дескрипторів НРК у формі «Матриці компетентностей» наведено у табл. 3.

Таблиця 3

Матриця компетентностей

	Компетентність	Знання	Уміння / навички	Комунікація	Автономія та відповідальність
Інтегральна компетентність					
1.	Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми, які характеризуються комплексністю і невизначеністю умов, у сфері розробки програмного забезпечення.	змісту базових понять, предмету та методів курсу і основних шаблонів моделювання	проводити аналіз вимог до ПЗ, що розробляється та постановку вимог до програмного забезпечення	Програмна взаємодія	Самостійне проектування і апробація на виробництві
Спеціальні (фахові, предметні) компетентності					

3.	<p>СК2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>СК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.</p> <p>СК14. Здатність до алгоритмічного та логічного мислення.</p>	<p>методології та технології збору основних вимог клієнтів та методи їх аналізу, основних етапів, принципів та стилів побудови складних програмних систем</p>	<p>проектувати та моделювати архітектуру ПЗ з використанням засобів візуального моделювання та розробляти інтерфейс програмних модулів, проводити декомпозицію</p>	<p>Застосовування технологій паралельної роботи</p>	<p>Опис методологій</p>
----	---	---	--	---	-------------------------

Інтегративні кінцеві програмні результати навчання, формуванню яких сприяє навчальна дисципліна:

<p>Програмні результати навчання</p>	<p>ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p>
---	--

Результати навчання:

Після опанування дисципліни студент повинен

знати:

- зміст базових понять, предмету та методів курсу;
- основні шаблони проектування;
- історію створення та розвитку універсальної мови моделювання (UML);
- можливості, переваги і недоліки різних методик об'єктно-орієнтованого аналізу і графічних нотацій;
- вигляд та призначення основних графічних засобів мови UML, основні

- види канонічних діаграм;
- аналізувати предметну область і описувати її з використанням мови UML;
- основні етапи, принципи та стилі побудови складних програмних систем;
- життєвий цикл програмного забезпечення, стадії у різних моделях життєвого циклу;

уміти:

- проводити аналіз вимог до ПЗ, що розробляється;
- оцінювати трудомісткість і вибирати адекватні підходи до розробки ПЗ;
- проектувати архітектуру ПЗ з використанням засобів візуального моделювання;
- оформляти програмну документацію;
- застосовувати інструментальні засоби об'єктно-орієнтованого аналізу і графічного представлення на мові UML (IBM Rational Rose);
- проектувати компоненти архітектурного рішення;
- проводити постановку вимог до програмного забезпечення;
- проводити аналіз архітектури програмного забезпечення на предмет відповідності атрибутам якості.

2. ІНФОРМАЦІЙНИХ ОБСЯГ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

На вивчення навчальної дисципліни «*Моделювання та аналіз програмного забезпечення*» відводиться 120 годин 4 кредитів ЄКТС.

МОДУЛЬ I. ТЕОРЕТИЧНІ І ПРАКТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ ТА АНАЛІЗУ

Змістовий модуль I. Вступ до дисципліни «Моделювання та аналіз ПЗ»

Тема 1. Вступ до дисципліни «Моделювання та аналіз ПЗ». Загальні підходи до моделювання програмного забезпечення

Змістовий модуль II. Основи моделювання та аналізу ПЗ

Тема 2. Основи структурного моделювання

Тема 3. Основи моделювання поведінки ПЗ

Тема 4. Основи моделювання подій

Тема 5. Основи моделювання архітектури ПЗ

3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Назви модулів і тем	Кількість годин (денна форма навчання)				
		Всього	у тому числі			
			Лекції	Практичні (семінарські)	Лабораторні	СРС
МОДУЛЬ I. ТЕОРЕТИЧНІ І ПРАКТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ ТА АНАЛІЗУ						
Змістовий модуль I. Вступ до дисципліни «Моделювання та аналіз ПЗ»						
1.	<u>Тема 1.</u> Вступ до дисципліни «Моделювання та аналіз ПЗ». Загальні підходи до моделювання програмного забезпечення	18	2	4	-	12
<i>Всього за змістовий модуль I:</i>		<i>18</i>	<i>2</i>	<i>4</i>	<i>-</i>	<i>12</i>
Змістовий модуль II. Основи моделювання та аналізу ПЗ						
2.	<u>Тема 2.</u> Основи структурного моделювання	30	6	8	-	16
3.	<u>Тема 3.</u> Основи моделювання поведінки ПЗ	28	4	8	-	16
4.	<u>Тема 4.</u> Основи моделювання подій	22	2	6	-	14
5.	<u>Тема 5.</u> Основи моделювання архітектури ПЗ	22	2	6	-	14
<i>Всього за змістовий модуль II:</i>		<i>102</i>	<i>14</i>	<i>28</i>	<i>-</i>	<i>60</i>
<i>Всього годин за курс:</i>		<i>120</i>	<i>16</i>	<i>32</i>	<i>-</i>	<i>72</i>

4. ТЕМИ ЛЕКЦІЙ

№ з/п	Назва модулів, тем та їх зміст
МОДУЛЬ I. ТЕОРЕТИЧНІ І ПРАКТИЧНІ ОСНОВИ МОДЕЛЮВАННЯ ТА АНАЛІЗУ	
Змістовний модуль I. Вступ до дисципліни «Моделювання та аналіз ПЗ»	
1.1.	<u>Лекція 1. Вступ до дисципліни «Моделювання та аналіз ПЗ». Загальні підходи до моделювання програмного забезпечення</u> Зміст теми: Основні підходи та концепції моделювання ПЗ. Три різні точки зору до розроблення ПЗ. Написання правильного коду мінімального обсягу. Що таке модель системи? Мета моделювання ПЗ. Неформальне і формальне моделювання. Принципи моделювання. Об'єкти моделювання. Введення в UML. Концептуальна модель UML. Будівельні блоки UML. Зв'язки UML. Діаграми UML. Правила UML. Загальні механізми UML. Специфікації (Specifications). Візуалізація системи та опис деталей. Доповнення (Adornments). Загальноприйняті поділи (Common divisions). Механізм розширення (Extensibility mechanisms). Початки роботи. Ключові абстракції. Пакетування. Механізми. Артефакти
Змістовний модуль II. Основи моделювання та аналізу ПЗ	
1.2.	<u>Лекція 2. Основи структурного моделювання. Моделювання класів, об'єктів та екземплярів</u> Зміст теми:

	<p>Класи. Вимоги до проектування класів. Організація атрибутів і операцій. Атрибути й операції як засоби виконання обов'язків. Формулювання обов'язків класу. Інші характеристики. Специфікація та реалізація класу. Кооперації класів. Активні класи, класифікатори, вузли. Моделювання словника системи. Моделювання розподілу обов'язків у системі. Основні механізми доповнення і розширення будівельних блоків UML. Примітка (note). Додаткові розділи. Моделювання коментарів. Практичне використання розширень. Розширені класи. Класифікатори. Розширені властивості класів. Види класифікаторів. Область дії екземпляра й статична область дії. Статичні операції. Абстрактні, листові й поліморфні елементи (класи) та операції. Розширення для атрибутів. Константні атрибути (readably). Операції. Синтаксис параметрів. Шаблонні класи. Створення екземпляра шаблонного класу. Явне і неявне зв'язування. Стандартні стереотипи розширення класифікаторів. Моделювання семантики класу. Екземпляри. Абстракція та екземпляр. Прямі й непрямі екземпляри. Аналогія абстрактних класів та інтерфейсів. Ім'я екземпляра. Зміна стану об'єкта. Активні й пасивні елементи</p>
1.3.	<p><u>Лекція 3. Основи структурного моделювання. Моделювання зв'язків та відношень</u> <i>Зміст теми:</i> Типові зв'язки. Детальніше про відношення і зв'язки. Розширення асоціації. Абстрактні класи. Множинне успадкування. Асиметричність зв'язків залежності і узагальнення. Асоціація як структурний шлях взаємодії об'єктів двох класів. Моделювання структурних зв'язків. Розширені зв'язки. Моделювання семантики зв'язків різних рівнів. Стереотипи зв'язків узагальнення між класами. Асоціації. Навігація (navigation). Рівні видимості кінців асоціації (ролі). Кваліфікація (qualification). Композиція (composition). Асоціації-класи (association class). Обмеження асоціації. Практичні рекомендації для моделювання систем зв'язків. Інтерфейси, типи, ролі. З'єднання в системі. Інтерфейс як клас зі стереотипом. Операції. Оголошення операцій інтерфейсу. Розуміння інтерфейсів. Практичні підходи моделювання інтерфейсів. Моделювання статичних і динамічних типів</p>
1.4.	<p><u>Лекція 4. Основи структурного моделювання. Діаграми для моделювання статичних характеристик системи</u> <i>Зміст теми:</i> Основні моделі статичних представлень системи. Основні базові поняття моделювання. Практичні підходи моделювання діаграм. Одна модель – різні рівні абстракції. Моделювання систем на різних рівнях абстракції. Створення діаграм. Діаграми класів. Основні властивості. Стандартне використання. Моделювання кооперацій. Діаграма класів описування реалізації автономного робота. Відмінність діаграм класів UML і E-R діаграми БД. Зразок ланцюжка обов'язків (responsibility pattern). Діаграми об'єктів. Загальні властивості. Статичний кадр у динамічному сценарії. Моделювання структури об'єктів. Компоненти та інтерфейси. Діаграми компонентів. Замінність. Організація компонентів. Внутрішня структура. Моделювання структурованих класів. Моделювання програмного інтерфейсу API</p>
1.5.	<p><u>Лекція 5. Основи моделювання поведінки ПЗ. Моделювання варіантів використання</u> <i>Зміст теми:</i> Стратегія розробки варіантів використання як кооперації класів. Поділ, реалізація і специфікація поведінки. Спочатку найпростіші проблеми – подробиці далі. Варіанти використання і дійові особи (актанти). Опис варіантів використання як потоку подій. Варіанти використання і сценарії (екземпляри) та діаграми взаємодій. Варіанти використання і кооперації. Кооперація класів</p>

	<p>Від варіантів використання до варіантів тестування. Організація варіантів використання. Варіанти використання у моделюванні поведінки системи. Приклад моделі поведінки системи. Діаграми варіантів використання. Зміст діаграми варіантів використання. Стандартне використання. Діаграми варіантів використання у задачах проектування. Моделювання контексту системи (context). Моделювання вимог до програмної системи. Діаграми варіантів використання керують вимогами. Варіанти використання тестування</p>
1.6.	<p><u>Лекція 6. Основи моделювання поведінки ПЗ. Моделювання взаємодії об'єктів</u> <i>Зміст теми:</i> Стратегія моделювання взаємодії як обміну поведінками між об'єктами. Контекст взаємодії. Об'єкти і ролі у взаємодії. Посилання та коннектори. Необхідна умова для обміну повідомлень між об'єктами. Прототипні об'єкти (ролі) та прототипні зв'язки або посилання (коннектори). Типи та екземпляри повідомлень. Посилання сигналів. Потoki та послідовності (sequencing) повідомлень. Процедурний потік керування. Створення, модифікація та знищення об'єктів і посилань. Моделювання потоку керування. Діаграми взаємодії. Стратегії розкладування сценаріїв. Структуроване високорівневе керування на діаграмах послідовності. Вкладені діаграми послідовності й діяльності (ref). Діаграми комунікацій. Семантична еквівалентність та відмінність моделей взаємодії. Прикладні аспекти застосування. Моделювання потоків керування, упорядкованих за часом. Пряме і зворотне проектування. Діаграми діяльності. Від одного кроку процесу до іншого. Вміст діаграми діяльності. Дії і вузли діяльності. Потoki керування. Розгалуження (branching). Поділ і з'єднання. Моделювання паралельних потоків у системі. «Плаваючі доріжки». Области розширення. Моделювання потоку робіт. Моделювання операцій</p>
1.7.	<p><u>Лекція 7. Основи моделювання подій. Моделювання подій сигналів</u> <i>Зміст теми:</i> Моделювання подій сигналів. Настання події – зміна станів. Види подій. Моделювання сигналів класами. Стан «рандеву». Моделювання події виклику. Поліморфні події. Кінцеві автомати. Моделювання поведінки спільно об'єктів. Події, стани, переходи й ефекти в ЖЦ об'єктів. Розширені UML – засоби моделювання поведінки. Стан Tracking (Переслідування). Моделювання поведінки банкомата. Неортогональні (nonorthogonal) стани. Історичний стан (history state). Процеси і потоки керування. Потік керування. Способи комунікації між об'єктами в потоках керування. Моделювання систем реального часу. Діаграми станів</p>
1.8.	<p><u>Лекція 8. Основи моделювання архітектури ПЗ</u> <i>Зміст теми:</i> Поняття про архітектуру та раціональний уніфікований процес. Сліпі і слон. Архітектурні вигляди. Життєвий цикл розроблення ПЗ. Моделювання архітектурних зразків. Механізми і каркаси як основа архітектури системи. Механізми як зразки проектування співтовариства класів. Шаблони й параметризовані кооперації. Моделювання кооперації. Відмінність кооперації і класів. Моделювання пакетів як способів організації елементів моделі. Організуючі сутності. Ім'я пакета. Елемент, що належить пакету. Простір імен пакета. Видимість елементів пакета (відкриті, захищені і закриті). Інтерфейс пакета. Імпорт і експорт. Моделювання груп елементів. Відмінність між класами і пакетами. Моделювання архітектурних виглядів</p>

5. ТЕМИ ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	<i>Теми лабораторних занять</i>	<i>К-сть годин</i>
1	Аналіз вимог та побудова цільової моделі предметної області	4
2	Структурне моделювання ПЗ. Методології ER.	4
3	Структурне моделювання ПЗ. Нотація Бахмана	4
4	Структурне моделювання ПЗ. Методології функціонального моделювання IDEF0	2
5	Структурне моделювання ПЗ. Методології функціонального моделювання DFD	2
6	Побудова діаграм варіантів використання (модель прецедентів)	4
7	Побудова діаграми класів в UML. Модель предметної області	4
8	Побудова діаграм взаємодії для виявлення розподілу обов'язків між об'єктами	4
9	Побудова діаграм стану та діяльності як інструмент аналізу складних транзакцій та операцій класу при проектуванні системи	4
Всього:		32

6. САМОСТІЙНА РОБОТА

№	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>К-сть годин</i>
1.	Структурна модель предметної області	4
2.	Функціонально-орієнтовані та об'єктно-орієнтовані методології опису предметної області (IDEF0)	4
3.	Призначення мови UML та її загальна структура	4
4.	Пакети на мові UML	4
5.	Базові семантичні конструкції мови, їх опис за допомогою спеціальних позначень	4
6.	Особливості графічного зображення діаграм мови UML	4
7.	Елементи графічної нотації діаграми варіантів використання	4
8.	Формалізація функціональних вимог до системи за допомогою діаграми варіантів використання	4
9.	Розширення мови UML для побудови моделей програмного забезпечення та бізнес-систем.	4
10.	Відношення та їх графічне зображення на діаграмі класів	4
11.	Внутрішні дії стану і діяльність	4
12.	Події та їх специфікація на діаграмах станів	4
13.	Особливості моделювання поведінки об'єктів у вигляді діаграм станів	4
14.	Особливості моделювання паралельного поведінки об'єктів у формі діаграм станів	4
15.	Особливості фізичного представлення програмних систем	4
16.	Використання діаграми компонентів для проектування залежностей між компонентами	4
17.	Специфіка подання ресурсномістких вузлів і технічних пристроїв	4
18.	Варіанти графічного зображення вузлів на діаграмі розгортання	4
Всього:		72

7. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни «*Моделювання та аналіз програмного забезпечення*» застосовуються інформаційні та практичні методи навчання, а саме: виконання та захист практичних робіт, тестування та опитування, презентації результатів опрацювання лекційних занять, опанування термінологічними поняттями засобами створення тлумачного словника, а також консультування з виконання самостійної роботи студентів.

Методи навчально-пізнавальної діяльності: пояснювально-ілюстративний метод, репродуктивний метод, метод проблемного викладу, частково-пошуковий або евристичний метод, дослідницький метод.

Методи стимулювання й мотивації навчально-пізнавальної діяльності: індуктивні і дедуктивні методи навчання, методи стимулювання і мотивації навчання.

8. МЕТОДИ КОНТРОЛЮ

Відповідно до плану вивчення дисципліни «*Моделювання та аналіз програмного забезпечення*» передбачається проведення поточного та підсумкового контролю.

Основними формами проведення поточного контролю є виконання та захист лабораторних робіт. Захист лабораторних робіт представляє собою усну відповідь на запитання викладача в межах теми лабораторної роботи та надання звіту про виконану роботу. При оцінці усної відповіді враховуються: знання теоретичного матеріалу з відповідної теми; цілісність та повнота відповіді на поставлені запитання; оперування науковими визначеннями та поняттями; термінологічна та технічна грамотність відповіді; логічність та лаконічність викладу матеріалу; уміння довести свою думку; уміння супроводжувати відповідь графічними засобами.

Перевірка результатів опрацювання теоретичних питань, проводиться у формі тестових завдань або формування реферату, презентації, розробки проекту.

Контроль самостійної роботи студентів спрямований на виявлення рівня розвитку пізнавальних здібностей та творчої ініціативи студентів, самостійності, відповідальності та організованості; рівня сформованості мислення, здібностей до саморозвитку, самовдосконалення та самореалізації та опанування студентами елементів методики наукових досліджень.

9. ФОРМА ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ НАВЧАННЯ

Формою підсумкового контролю є *екзамен*, який складається очно (при особливій ситуації – у формі комп'ютерного тесту) в період призначений деканатом або за індивідуальним графіком, який затверджується навчальним планом.

10. СХЕМА НАРАХУВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Накопичення балів протягом семестру

№ з/п	Вид діяльності	Кількість балів за дидактичну одиницю	Кількість	Загальна кількість балів
1	Практична частина (виконання практичних робіт із захистом звіту)	2	14	28
2	Теоретична частина (контроль на лекціях та тестування)	14	1	14
3	Самостійна робота (створення тлумачного словника)	18	1	18
Максимальна кількість балів:				60

Загальна оцінка знань студентів за поточним контролем

Результати поточного контролю знань студентів в цілому оцінюються в діапазоні від **0** до **60** балів.

Студент допускається до підсумкового контролю за умови виконання вимог навчальної програми та у разі, якщо за поточну навчальну діяльність він набрав не менше **36** балів.

Підсумкове оцінювання знань студентів

Підсумкове оцінювання знань студентів проводиться у формі *екзамену*.

Розподіл балів оцінювання при підсумковому контролі з навчальної дисципліни

Оцінка в балах за поточне оцінювання	Оцінка за національною шкалою
35-40	Відмінно
21-34	Добре
10-20	Задовільно
менше 10	Незадовільно

Під час оцінювання відповіді на окреме питання під час екзамену додатково враховуються допущені недоліки та помилки, якими вважаються:

- неохайне оформлення роботи (не загальноприйняті скорочення, незрозумілий почерк, використання олівців замість чітких чорнил) (мінус **2** бали);
- неточності в назвах окремих термінів та понять (мінус **4** бали).

Критерії оцінювання відповіді на теоретичні питання:

1. Повна відповідь на питання, яка оцінюється «*відмінно*», повинна відповідати таким вимогам:

- розгорнутий, вичерпний виклад змісту даної у питанні проблеми;

- повний перелік необхідних для розкриття змісту питання фізичних термінів та законів;
- здатність здійснювати порівняльний аналіз різних теорій, концепцій, підходів та самостійно робити логічні висновки й узагальнення;
- уміння користуватись методами наукового аналізу фізичних явищ, процесів і характеризувати їхні риси та форми виявлення;
- демонстрація здатності висловлення та аргументування власного ставлення до альтернативних поглядів на дане питання;

2. Відповідь на питання оцінюється **«добре»**, якщо:

- відносно відповіді на найвищий бал не зроблено розкриття хоча б одного з пунктів, вказаних вище (якщо він явно потрібний для вичерпного розкриття питання) або, якщо:
- при розкритті змісту питання в цілому правильно за зазначеними вимогами зроблені окремі помилки під час: використання формул.

3. Відповідь на питання оцінюється **«задовільно»**, якщо:

- відносно відповіді на найвищий бал не зроблено розкриття чотирьох чи більше пунктів, зазначених у вимогах до нього (якщо вони явно потрібні для вичерпного розкриття питання);
- одночасно присутні чотири чи більше типів недоліків, які окремо характеризують критерій оцінки питання;
- висновки, зроблені під час відповіді, не відповідають правильним чи загально визначеним при відсутності доказів супротивного аргументами, зазначеними у відповіді;
- характер відповіді дає підставу стверджувати, що особа, яка складає іспит, не зовсім правильно зрозуміла зміст питання чи не знає правильної відповіді і тому не відповіла на нього по суті, допустивши грубі помилки у змісті відповіді.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
66-73	D	задовільно	
60-65	E		
30-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-29	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Загальна підсумкова оцінка в балах, за національною шкалою та за шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

11. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- робоча навчальна програма дисципліни;
- електронний курс на платформі дистанційного навчання;
- плани лекцій, практичних занять та самостійної роботи студентів;
- тези лекцій з дисципліни;
- тестові завдання до тем лекцій;
- перелік питань до екзамену.

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Петрик М.Р., Петрик О.Ю. Моделювання програмного забезпечення : Науково-методичний посібник. Тернопіль : Вид-во ТНТУ імені Івана Палія, 2015. 200 с.
2. Лавріщева К.М. Програмна інженерія.–Київ.– 2008.–319 с.
3. Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолтНТУ, 2017. – 218 с.
4. К.Є. Золотько, Д.В. Красношарпа, С.Ф.Сірик, Методичні рекомендації до виконання лабораторних робіт з дисципліни «Проектування програмних систем». – Дніпро, 2018. – 27с.
5. Методичні вказівки до лабораторного практикуму з дисципліни „Конструювання програмного забезпечення” для студентів напряму підготовки 6.050103 - “Програмна інженерія” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: ЧПБіП, 2016. — 34с.
6. Технологія створення програмних продуктів [Електронний ресурс]: лабораторний практикум для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / уклад. В.А. Литвинов, М.В. Гладка, О.А. Хлобистова: НУХТ, 2014.– 86 с
7. Алістер Коберн. Сучасні методи опису функціональних вимог до систем
8. Фаулер Скотт До. UML в короткому викладі. Застосування стандартної мови об'єктного моделювання: Пер. з англ. – М.:Мир, 1999. – 191 с.
9. Смірнов В.В. Технологія проектування програмних систем. Лабораторні роботи / В.В. Смірнов, Н.В. Смірнова. – Кіровоград: КНТУ, 2012. – 53 с.
10. Смірнов В.В. Технологія проектування програмних систем. Лекції / В.В. Смірнов, Н.В. Смірнова. – Кіровоград: КНТУ, 2012. – 95 с.

Інформаційні ресурси:

1. Визначення архітектури додатків за допомогою Rational Software Architect. – Режим доступу: <https://www.ibm.com/developerworks/ru/library/r-define-application-architecture-rational-software-architect-2/index.html>.
2. Система дистанційного навчання Міжнародного Європейського Університету. Курс «Моделювання та аналіз програмного забезпечення». Режим доступу: <https://dist.ieu.edu.ua/course/view.php?id=635>
3. C++ Tutorial. – Режим доступу: <https://www.tutorialspoint.com/cplusplus/index.htm>.
4. Java Tutorial. – Режим доступу: <https://www.tutorialspoint.com/java/index.htm>.
5. Matplotlib – офіційна сторінка бібліотеки. – Режим доступу: <https://matplotlib.org>.
6. UML Tutorial. – Режим доступу: <https://www.tutorialspoint.com/uml/index.htm>.