

“ЗАТВЕРДЖУЮ”

Завідувач кафедри

Білоус Ірина Володимирівна

“ _____ ” _____ 20__ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Проектування програмного забезпечення (ОК20)

Освітня програма «Інженерія програмного забезпечення»

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 121 – Інженерія програмного забезпечення

Мова навчання: українська

Статус дисципліни: обов'язкова

Форма навчан.	Рік навч.	Сем.	Розподіл годин					Разом	За тиждень		ІНДЗ	Контр.
			Всього ауд.	Лек	Прак	Лаб.	СРС		Ауд.	СРС		
Денна	3	6	30	16		14	90	120	2,1	8,5	РГР	І
	4	7	30	16		14	60	90	3	9	КП	3
	Разом		60	32		28	150	210	5,1	17,5		

Робоча програма Проектування програмного забезпечення для здобувачів вищої освіти галузі знань 12 – «Інформаційні технології» спеціальності 121 – «Інженерія програмного забезпечення»

Розробник робочої навчальної програми:

старший викладач кафедри інформаційних технологій та програмної інженерії,

_____ (*В.В. Нехай*)
(підпис) (прізвище та ініціали)

Робоча програма схвалено на засіданні кафедри *інформаційних технологій та програмної інженерії*

Протокол від “02” вересня 2020 року №1

Завідувач кафедри *інформаційних технологій та програмної інженерії*

_____ (*І.В. Білоус*)
(підпис) (прізвище та ініціали)

Abstract

ESIEIT/SE OK20 Software Design 2020/2021 Sem. 2, 2021/2022 Sem. 1 Course Description

Software Design is a systematic and disciplined approach to developing software. It applies both computer science and engineering principles and practices to the creation, operation, and maintenance of software systems.

Software Design is concerned with developing and maintaining software systems that behave reliably and efficiently, are affordable to develop and maintain, and satisfy all the requirements that customers have defined for them. It is important because of the impact of large, expensive software systems and the role of software in safety-critical applications. It integrates significant mathematics, computer science and practices whose origins are in engineering.

Contents: existing tools for building software models, trends in the development of software tools.

1 Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	
Кількість кредитів – 7	Галузь знань 12 Інформаційні технології	Обов'язкова	
Модулів – 4	Спеціальність: <i>121 – Інженерія програмного забезпечення</i> Освітньо-професійна програма: <i>Інженерія програмного забезпечення</i>	Рік підготовки:	
Змістових модулів – 4		3,4-й	
Індивідуальне науково-дослідне завдання – розрахунково-графічна курсова робота, курсовий проект		Семестр	
Загальна кількість годин – 210		6-й	7-й
Тижневих годин: аудиторних – 2,1 (6-й семестр); 3 (7-й семестр) самостійної роботи і індивідуальної студента – 8,5 (6-й семестр) та 9 (7-й семестр)	Освітньо-кваліфікаційний рівень: бакалавр	Лекції	
		1,1 год.	1,6 год.
		Лабораторні	
		1 год.	1,4 год.
		Самостійна робота	
		8,5 год.	9 год.
		Вид контролю:	
Іспит РГР	Залік КП		

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

у 6-ому семестрі – $30:120 = 1:4$;

у 7-ому семестрі – $30:90 = 1:3$;

Передумовою для вивчення дисципліни є успішне засвоєння дисциплін “Основи програмування”, “Системне програмування”, “Об’єктно-орієнтовне програмування”, “Бази даних”.

Дисципліна є базовою для вивчення дисциплін освітньої програми «Інженерія програмного забезпечення» другого (магістерського) рівня вищої

освіти, а також може використовуватися під час підготовки випускної кваліфікаційної роботи бакалавра за відповідною темою.

2 Мета та завдання навчальної дисципліни

Метою викладання дисципліни “*Проектування програмного забезпечення*” є закріплення та розвиток фахових компетентностей в галузі знань 12 – *Інформаційні технології* із практичним застосуванням при розробці прикладних інформаційних систем.

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК5. Здатність вчитися і оволодівати сучасними знаннями.
- ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК31. Здатність працювати в міжнародному контексті.
- ФК15. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
- ФК16. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- ФК17. Здатність розробляти архітектури, модулі та компоненти програмних систем.
- ФК18. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.
- ФК25. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
- ФК27. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.
- ФК 30. Здатність проводити комплексну оцінку варіантів ІТ-проектів.

3 Очікувані результати навчання з дисципліни

Навчальна дисципліна “*Проектування програмного забезпечення*” має допомогти сформуванню наступні програмні результати навчання.

- ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.
- ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.
- ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

- ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.
- ПР08. Вміти розробляти людино-машинний інтерфейс.
- ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.
- ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.
- ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.
- ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.
- ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.
- ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.
- ПР16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.
- ПР19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.
- ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.
- ПР22. Знати та вміти застосовувати методи та засоби управління проектами.
- ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.

Після вивчення дисципліни студенти **повинні знати:**

- поняття системи та її оточення;
- основні поняття системотехніки;
- особливості програмних систем;
- основні моделі процесів розробки програмних систем;
- методи та моделі управління процесом розробки програмних систем;
- класифікацію вимог до програмних систем;
- моделі та методи розробки вимог користувача та системних специфікацій;
- моделі систем та їх переваги та недоліки;
- технології автоматизації процесів розробки програмних систем;
- методології проектування програмних систем;
- сучасні архітектури та технології проектування програмних систем;
- типову модель життєвого циклу програмного забезпечення;
- стадії тестінга у життєвому циклі продукту;
- класифікацію типів тесту за різними ознаками;
- класифікацію дефектів та їх атрибути;

- поняття контролю якості та гарантії якості;
- принципи складання тест-циклів;

У результаті опанування навчальною дисципліною студенти **повинні вміти** :

- застосовувати на практиці основні моделі розробки програмних систем;
- розробляти робочу документацію проектування програмних систем;
- розробляти вимоги користувача та системні програмні вимоги до програмних систем;
- застосовувати на практиці моделі та методи розробки програмних систем;
- розробляти тест-плани, тест-кейси, звіт про знайдений дефект: застосовувати на практиці різноманітні типи тестінгу.

4 Критерії оцінювання результатів навчання

З тими студентами, які до проведення підсумкового семестрового контролю не встигли виконати всі обов'язкові види робіт та мають підсумкову оцінку менше 20 балів (за шкалою оцінювання), проводяться додаткові індивідуальні заняття, за результатами яких визначається, наскільки глибоко засвоєний матеріал, та чи необхідне повторне вивчення дисципліни.

Дисципліну можна вважати такою, що засвоєна, якщо студент:

1) знає:

- закономірності і основні моделі життєвого циклу ПЗ;
- сучасні технологічні підходи до створення ПЗ, їх можливості, межі застосування, переваги та недоліки; завдання і методи розробки архітектури ПЗ;
- основні підходи до вирішення завдання забезпечення заданого рівня якості ПЗ;
- методи проектування ПЗ.

2) вміє:

- проводити аналіз вимог до ПЗ, що розробляється;
- оцінювати трудомісткість і вибирати адекватні підходи до розробки ПЗ;
- проектувати архітектуру ПЗ з використанням засобів візуального моделювання;
- усвідомлено застосовувати методики випробувань і налагодження розроблюваного ПЗ; проектувати компоненти архітектурного рішення.

5 Засоби діагностики результатів навчання

Засобами оцінювання та методами демонстрування результатів навчання з дисципліни є:

- іспит;
- залік;
- розрахунково-графічна робота;
- курсовий проект;

- презентації результатів виконаних завдань;
- завдання, які виконуються в навчальній лабораторії;
- інші види індивідуальних та групових завдань.

6 Програма навчальної дисципліни

МОДУЛЬ 1 Основи моделювання

Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми розробки

ПЗ

Тема 1. Вступ

Особливості структурного підходу до проектування ПЗ. Об'єктно-орієнтована декомпозиція та абстрактні типи даних. Об'єкт як основа нової парадигми.

Тема 2 Об'єктно-орієнтовані графічні нотації

Набір нотацій UML, ER нотації.

Тема 3. Основні діаграми UML:

- діаграми класів (class diagram);
- діаграми кооперації (collaboration diagram);
- діаграми послідовностей (sequence diagram);
- діаграми прецедентів (use case diagram);
- діаграми розгортання та компонентів (deployment diagram, component diagram);
- діаграми пакетів (package diagram);
- діаграми станів (statechart);
- діаграми діяльності (activity diagram).

Тема 4 . Основні діаграми ER

Нотація Пітера Чена

Тема 5. Зв'язок між нотаціями та мовами об'єктно-орієнтованого програмування. Відображення нотацій в об'єктно-орієнтованих мовах програмування: c++, Object Pascal, Small Talk.

МОДУЛЬ 2 Планування конструювання

Змістовий модуль 2. Планування конструювання програмного забезпечення

Тема 6. Моделі життєвого циклу розробки програм

- Каскадні моделі ЖЦ
- Інкрементні моделі ЖЦ
- Спіралеподібні моделі ЖЦ

Тема 7. Основні процедури проектування на етапі аналізу вимог для

ПЗ

Поняття вимог для ПЗ, їх класифікація. Задачі управління вимогами. Трасування вимог. Стратегії реалізації вимог. Засоби UML при аналізі вимог до ПЗ.

Тема 8. Процедури етапу об'єктного аналізу

Об'єктний структурний і об'єктний поведінковий аналіз. Об'єктна декомпозиція. Структурна модель системи. Визначення поведінки об'єктів.

Тема 9. Процедури розробки архітектури ПЗ. Шаблони проектування архітектури

Поняття архітектури системи та вимоги до неї.

Процедури:

- відображення програмних пакетів на процесори;
- вибір шини та протоколів обміну;
- визначення моделі паралелізму і потоків задач.

Математичне та імітаційне моделювання. Рівні програмного забезпечення та вертикальні зрізи. Глобальна політика обробки помилок.

Тема 10. Процедури розробки механізмів ПЗ. Шаблони проектування механізмів

Основні особливості етапу проектування механізмів. Екземпляри шаблонів проектування для багатьох об'єктів, взаємодіючих спільно. Контейнери, класи та об'єкти рівня проектування. Політики середнього рівня для обробки помилок. Шаблони: прості та повторного використання. Бібліотеки шаблонів. Фреймворки.

Тема 11. Процедури детального проектування

Алгоритмічні деталі всередині об'єкта. Деталі структур даних в об'єктах (типи, діапазони та інші). Деталі функціональних членів (аргументи, внутрішня структура). Структури даних. Реалізація асоціацій. Набір операцій, визначених над даними. Видимість даних і операцій. Алгоритми, що використовуються для реалізації операцій. Підтримка виняткових ситуацій.

МОДУЛЬ 3 Планування та мови конструювання. Інтеграція

Змістовий модуль 3. Кодування та інтеграція

Тема 12 Огляд об'єктно-орієнтованих алгоритмічних мов програмування

C++, Small Talk, Ada, та ін..

Тема № 13 Інтегровані середовища підтримки управління розробкою програмного забезпечення розподіленими командами програмістів

Поняття успіху проекту. Управління вимогами. Задачі та «помилки». Управління версіями і конфігураціями. Управління тестуванням. Вимоги до вибору інтегрованого середовища.

Тема 14 Методи інтеграції ПЗ при проектуванні

Методи використання контролю версій, побудови builds при розробці програмної системи.

Тема 15 Засоби інтеграції розподілених систем

Можливості стандартів COM та CORBA.

МОДУЛЬ 4 Якість конструювання

Змістовий модуль 4. Тестування та забезпечення якості програмного забезпечення

Теми 16 Основні поняття тестування ПЗ

Поняття тестування, основні визначення, нотації і концепти. Класифікація відмов. Основні методи тестування. Принципи тестування ПЗ. Типи тестів ПЗ. Планування тестування. Розробка тестів. Виконання тестів і обробка результатів тестування. Управління тестуванням.

Тема № 17 Тестування ПЗ об'єктно-орієнтованих систем.

Огляд особливостей технології тестування FREE. Модель станів для FREE технології. Тестування класів, засноване на виконанні. Тестування класів або малих кластерів, що базується на представленні. Інтеграція великих кластерів і підсистем. Тестування системи з використанням розширених Use Cases.

Тема № 18 Особливості процесів програмування, налагодження і тестування програм для вбудованих систем і систем реального часу

Технологія створення програмного коду. Режим налагодження. Апаратні і програмні засоби налагоджувача Р&Е від компанії РЕМІСРО. Емулятори. Логічні аналізатори. Особливості компілятора і асемблера..

7 Структура навчальної дисципліни

Назви змістових модулів і тем		Кількість годин для денної форми навчання				
		Всього	У тому числі			
			Лек.	Пр.	Лаб.	Ср.
1	2	3	4	5	6	7
6-й семестр						
Модуль 1 Основи моделювання						
Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми розробки ПЗ						
1	Вступ	6	1			5
2	Об'єктно-орієнтовані графічні нотації	6	1			5
3	Основні діаграми UML	18	1		2	15
4	Основні діаграми ER	19	2		2	15
5	Зв'язок між нотаціями та мовами об'єктно-орієнтованого програмування	6	1			5
Разом за змістовим модулем 1		55	6		4	45
Модуль 2 Планування конструювання						
Змістовий модуль 2. Планування конструювання програмного забезпечення						
6	Моделі життєвого циклу розробки програм	6	1			5
7	Основні процедури проектування на етапі аналізу вимог для ПЗ	8	1		2	5
8	Процедури етапу об'єктного аналізу	14	2		2	10
9	Процедури розробки архітектури ПЗ. Шаблони проектування архітектури	14	2		2	10
10	Процедури розробки механізмів ПЗ. Шаблони проектування механізмів.	14	2		2	10
11	Процедури детального проектування	9	2		2	5
Разом за змістовим модулем 2		65	10		10	45
Разом за 6-й семестр		120	16		14	90
7-й семестр						
Модуль 3 Планування та мови конструювання. Інтеграція						
Змістовий модуль 3. Кодування та Інтеграція						
12	Огляд об'єктно-орієнтованих алгоритмічних мов програмування	7	2			5
13	Інтегровані середовища підтримки управління розробкою програмного	14	2		2	10

	забезпечення розподіленими командами програмістів.				
14	Методи інтеграції ПЗ при проектуванні	18	4	4	10
15	Засоби інтеграції розподілених систем	9	2	2	5
Разом за змістовим модулем 3		48	10	8	30
Модуль 4 Якість конструювання					
Змістовий модуль 4. Тестування та забезпечення якості програмного забезпечення					
16	Основні поняття тестування ПЗ	14	2	2	10
17	Тестування ПЗ об'єктно-орієнтованих систем.	14	2	2	10
18	Особливості процесів програмування, налагодження і тестування програм для вбудованих систем і систем реального часу.	14	2	2	10
Разом за змістовим модулем 4		42	6	6	30
Разом за 7-й семестр		90	16	14	60
Усього годин за дисципліну		210	32	28	150

8 Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Вступне заняття. Ознайомлення з навчально-методичним програмним комплексом, робочими місцями та правилами проведення та здачі лабораторних робіт. Вступний інструктаж з техніки безпеки. Складання вимог до ПЗ	2
2	Аналіз предметної області і сценарії використання	4
3	Об'єктна модель ПЗ. Діаграма класів.	4
4	Розробка технічного завдання	4
5	Методологія управління проектами. Аналіз і управління ризиками.	4
6	Розробка технічного проекту. Пояснювальна записка.	4
7	Документація на основі програмного коду. DOxygen	2
8	Засоби автоматизованого тестування ПЗ. Пакет Rational purify plus. Створення дистрибутива ПЗ за допомогою InnoSetup.	4
Разом		28

9 Самостійна робота

№ з/п	Назва теми	Кількість Годин
1	Робота з конспектом лекцій, методичними вказівками, основною та додатковою літературою, підготовка до	20

	тестів	
2	Підготовка до лабораторних занять	20
3	Проробка окремих розділів програми, які не викладались на лекціях	90
4	Обробка теоретичних матеріалів по використанню технічних засобів, які використовуються при виконанні лабораторних робіт.	20
	Разом	150

10 Індивідуальні завдання

Робочим планом передбачено виконання розрахунково-графічної роботи та курсового проекту. Докладна інформація щодо змісту, варіантів завдань, порядку оформлення та захисту курсового проекту міститься в [14.2, 14.3].

Форми контролю виконання РГР

Вид роботи	Форма контролю	Кількість балів
Структура	Відповідність умовам завдання. Відповідність вимогам	0...5
Пояснювальна записка	Обґрунтованість рішень. Відповідність оформлення вимогам. Своєчасність здачі	0...5
Захист РГР	Самостійність виконання (відповіді на запитання)	0...10
	Разом	0...20

Оцінка за виконання курсового проекту

Вид роботи	Форма контролю	Кількість балів
Програмна частина	1. Відповідність умовам завдання	0... 10
	2. Демонстрація завершеної роботи	0... 10
	3. Своєчасність здачі етапів	0... 10
Пояснювальна записка	1. Обґрунтованість рішень	0... 10
	2. Відповідність оформлення вимогам	0... 10
	3. Своєчасність здачі	0... 10
Захист роботи	Самостійність виконання (відповіді на запитання)	0... 40
	Разом	0... 100

11 Методи контролю

Оцінювання знань студентів здійснюється відповідно до «Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол №6 від 31.08.2020 р.) та затвердженого наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26.

З дисципліни студент може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру і до 40% підсумкової оцінки – на екзамені.

Виконання та особистий захист усіх лабораторних робіт, зазначених у робочій навчальній програмі з дисципліни, є обов'язковим. Поточний контроль проводиться шляхом спілкування із студентами під час лекцій та консультацій та опитувань студентів під час захисту лабораторних робіт.

Результати поточного контролю за відповідний модуль оприлюднюються викладачем на наступному аудиторному занятті. Бали, які набрані студентом під час модульних контролів, складають оцінку поточного контролю.

Семестровий контроль у вигляді *іспиту (6-й семестр) та заліку (7-й семестр)*. *Іспит* проводиться під час сесії з трьома запитаннями: двома теоретичними (по 10 балів максимум за кожне) та одним практичним (20 балів максимум). *Залік* проводиться під час сесії з чотирьома запитаннями (по 10 балів максимум за кожне).

Оцінка за результатами вивчення дисципліни формується шляхом додавання підсумкових результатів поточного контролю до екзаменаційної оцінки. Ті студенти, які не виконали всі обов'язкові види робіт та за результатами роботи в семестрі набрали менше 20 балів, мають пройти повторний курс вивчення дисципліни. Взаємозв'язок між набраними балами і оцінкою наведений у розділі 12.

Якщо відповідь повна і зміст відповіді студента повністю відповідає сутності поставленого запитання, можна отримати від 33 до 40 балів. В тому випадку, коли студент виконує всі завдання без грубих помилок, можна отримати від 24 до 32 балів. Якщо при виконанні завдань студент допускає грубі помилки, і всі запитання вирішені менш, ніж на половину, можна отримати від 17 до 24 балів. За невиконання хоча б одного завдання, не можна отримати більше 16 балів.

Складання іспиту та заліку є обов'язковим елементом підсумкового контролю знань для студентів, які претендують на оцінку «добре» або «відмінно». Якщо студент виконав всі види робіт протягом семестру та набрав 60% підсумкової оцінки (тобто «задовільно»), то він, за бажанням, може залишити набрану кількість балів як підсумкову оцінку і не складати екзамен.

В випадку повторного складання екзамену або заліку всі набрані протягом семестру бали анулюються, а повторний екзамен складається з трьома питаннями: двома теоретичними (по 30 балів максимум за кожне) та одним практичним (40 балів максимум). Екзаменаційні білети знаходяться у пакеті документів на дисципліну.

У випадку, якщо студент протягом семестру не виконав в повному обсязі передбачених робочою програмою всіх видів навчальної роботи, має невідпрацьовані лабораторні роботи або не набрав мінімально необхідну кількість балів (20), він не допускається до складання екзамену під час семестрового контролю, але має право ліквідувати академічну заборгованість у порядку, передбаченому «Положенням про поточне та підсумкове оцінювання знань ЗВО НУ «Чернігівська політехніка».

Повторне складання іспиту або заліку з метою підвищення позитивної оцінки не дозволяється.

Політика дотримання академічної доброчесності ґрунтується на «Кодексі академічної доброчесності Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол №6 від 31.08.2020 р.) та введеного в дію наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26.

12 Розподіл балів, які отримують студенти

Поточний контроль за модулями

Модуль за тематичним планом дисципліни та форма контролю	Кількість балів
6-й семестр	
Модуль 1 Основи моделювання	
Змістовий модуль 1. Основи об'єктно-орієнтованої парадигми розробки ПЗ	0... 20
1 Рішення завдань до захисту лабораторних робіт.	0... 7
2 Самостійність виконання лабораторних робіт.	0... 7
3 Своєчасність виконання лабораторних робіт.	0... 6
Модуль 2 Планування конструювання	
Змістовий модуль 2. Планування конструювання програмного забезпечення	0... 20
1 Рішення завдань до захисту лабораторних робіт.	0... 7
2 Самостійність виконання лабораторної роботи.	0... 7
3 Своєчасність виконання лабораторних робіт.	0... 6
4 Захист розрахунково-графічної роботи	0... 20
Семестрова оцінка поточного контролю	0... 60
7-й семестр	
Модуль 3 Планування та мови конструювання. Інтеграція	
Змістовий модуль 3. Кодування та Інтеграція	0... 30
1 Рішення завдань до захисту лабораторної роботи.	0... 10
2 Самостійність виконання лабораторних робіт.	0... 10
3 Своєчасність виконання лабораторних робіт.	0... 10
Модуль 4 Якість конструювання	
Змістовий модуль 4. Тестування та забезпечення якості програмного забезпечення	0... 30
1 Рішення завдань до захисту лабораторної роботи.	0... 10

2	Самостійність виконання лабораторних робіт.	0... 10
3	Своєчасність виконання лабораторних робіт.	0... 10
Семестрова оцінка поточного контролю		0... 60

Для захисту лабораторної роботи студент повинен відповісти на всі контрольні запитання з методичних вказівок та на два запитання за вибором викладача з лекційного курсу за темою лабораторної роботи. Для *денної форми навчання* за кожну лабораторну роботу студент отримує певну кількість балів з урахуванням максимальної кількості балів згідно наведеної вище таблиці. При цьому враховується якість оформлення звіту та повнота відповідей на запитання при захисті лабораторної роботи.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту	для заліку
90 – 100	A	відмінно	Зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13 Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

Лекційний матеріал подається у вигляді презентацій за допомогою медіа-проектора. Під час лекцій аналізуються проблемні ситуації, організується зворотний зв'язок з аудиторією шляхом формулювання запитань і стислих відповідей з обох сторін.

Лабораторне заняття включає проведення поточного контролю підготовленості студентів до виконання конкретної лабораторної роботи, виконання завдань теми заняття, оформлення індивідуального звіту з виконаної роботи та його захисту перед викладачем.

Рекомендовані технології для реалізації лабораторних робіт, розрахунково-графічної роботи та курсової роботи – Umbrello, Java v.7 і старше, Eclipse, C++, Qt.

14 Методичне забезпечення

1. Методичні рекомендації до виконання лабораторних робіт з дисципліни „Проектування програмного забезпечення” для студентів спеціальності 121 – „Інженерія програмного забезпечення.
2. Методичні рекомендації до виконання розрахунково-графічної роботи з дисципліни „Проектування програмного забезпечення” для студентів спеціальності 121 – «Інженерія програмного забезпечення»
3. Методичні вказівки до курсового проектування з дисципліни „Проектування програмного забезпечення” для студентів спеціальності 121 – „Інженерія програмного забезпечення” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: ЧНТУ, 2016. — 19с.

15 Рекомендована література

Базова

1. Бородкіна І., Бородкин Г. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів / І. Бородкіна, Г. Бородкин – М: Центр навчальної літератури, 2018. – 204 с.
2. Постіл С. Д. UML. уніфікована мова моделювання інформаційних систем / С. Д. Постіл : Ун-т держ. фіск. служби України. - Ірпінь : Ун-т держ. фіск. служби України, 2019. - 321 с.
3. Мартін Р. Чистий код / Р. Мартін – М.: Фабула, 2019. – 416 с.
4. Роберт М. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.

Допоміжна

1. Табунщик Г. В., Каплієнко Т.І., Петрова О.А. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с
2. Леоненков О. Самовчитель UML. Ефективний інструмент моделювання інформаційних систем / О. Леоненков. – М.: БХВ-Петербург, 2001. – 304 с.
3. Лавріщева К.М. Програмна інженерія / К.М. Лавріщева –К.– 2008.–319 с.

16 Інформаційні ресурси

1. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/uml/index.htm>. – Назва з екрану.
2. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/java/index.htm>. – Назва з екрану.
3. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/cplusplus/index.htm>. – Назва з екрану.