

“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	1
	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:

к.т.н., доцент кафедри інформаційних технологій та програмної інженерії,

_____ (*I.V. Богдан*)

(підпис)

(прізвище та ініціали)

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

Протокол від “31” серпня 2021 року №1

Завідувач кафедри *інформаційних технологій та програмної інженерії*

_____ (*I.V. Білоус*)

(підпис)

(прізвище та ініціали)

Abstract

ESIEIT/SE OK20 Software Design 2021/2022 Sem. 2, 2022/2023 Sem. 1 Course Description

The **subject of studying** is the software design process, ie the phases and iterations of the software life cycle and the software development methodology.

The purpose of course is the formation of the scientific and professional outlook of the bachelor of specialty 121 – Software engineering in the field of software quality assurance.

The primary studying goals of the discipline:

- Introduction to the concept of software design procedures.
- Acquaintance with the existing software development methodologies.
- Introduction to analyze software requirements process.
- Practical acquaintance with features of construction of basic UML diagrams.

Contents: software development methodologies, analyze software requirements, SPS, technical task, UML, use case diagram, class diagram, activity diagram, sequence diagram, component diagram, deployment diagram.

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 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» першого (бакалаврського) 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 навчальної дисципліни “*Проектування програмного забезпечення*” є формування науково-професійного світогляду бакалавра спеціальності 121 – *Інженерія програмного забезпечення* в області проектування програмного забезпечення. Предмет вивчення – **процес проектування програмного забезпечення**, тобто фази та ітерації життєвого циклу програмного забезпечення та методології створення програмного забезпечення.

Одним з найважливіших етапів створення програмного забезпечення є етап проектування. На даний момент існує достатньо значна кількість різних підходів до аналізу вимог до створюваного програмного забезпечення, а також методологій його створення і від правильності вибору як першого, так і другого залежить успіх всього процесу розробки та його вартість. Успішне засвоєння дисципліни дозволяє бакалавру з програмної інженерії розширити коло застосування набутих раніше знань та практичних навичок для вирішення широкого кола зада, пов'язаних з проектуванням програмного забезпечення.

Під час вивчення дисципліни здобувач вищої освіти (ЗВО) має набути або розширити наступні загальні (ЗКх) та фахові (ФКх) компетентності, передбачені освітньою програмою:

- 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.
- ЗК13. 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.
- ФК23. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.
- 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 наступних програмних результатів 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.
- 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.
- ПР22. Знати та вміти застосовувати методи та засоби управління проектами.
- ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.
- ПР24. Вміти проводити розрахунок економічної ефективності програмних систем.
- ПР 25. Мати навички виконання певних ролей в ІТ-проектах будь-якої складності.

Після вивчення дисципліни студенти **повинні знати:**

- поняття життєвого циклу програмного забезпечення та основні його процеси, фази та ітерації;
- особливості найбільш часто використовуваних методологій створення програмного забезпечення, їх переваги та недоліки;
- класифікацію вимог до створюваного програмного забезпечення;
- методи та підходи до визначення та аналізу вимог до програмного забезпечення;
- основні ролі в процесі створення програмного забезпечення та

особливості кожної з них;

– особливості створення базових діаграм мови моделювання UML.

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

- обирати методологію створення програмного забезпечення в залежності від його особливостей та вимог до нього;
- розробляти робочу документацію проектування програмного забезпечення;
- розробляти вимоги користувача та системні програмні вимоги до програмних систем;
- будувати базові діаграми мови моделювання UML.

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:

- основні методології створення програмного забезпечення;
- основні підходи до визначення та аналізу вимог до програмного забезпечення;
- особливості створення базових діаграм мови моделювання UML.

2) вміє:

- проводити аналіз вимог до ПЗ, що розробляється;
- обирати методологію створення програмного забезпечення в залежності від його особливостей та вимог до нього;
- створювати базові діаграми мови моделювання UML.

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

Змістовий модуль 1. Базові поняття. Методології створення програмного забезпечення

Тема 1. Вступ. Поняття життєвого циклу програмного забезпечення та

проектування програмного забезпечення. Етапи життєвого циклу програмного забезпечення.

Предмет і мета вивчення дисципліни та основні вирішувані задачі. Структура навчального курсу. Навчально-методична література з дисципліни. Методичне забезпечення. Рекомендована література. Базова література. Допоміжна література. Інформаційні ресурси.

Поняття життєвого циклу програмного забезпечення. Етапи життєвого циклу програмного забезпечення. Поняття проектування програмного забезпечення, як одного з етапів його життєвого циклу.

Тема 2. Поняття методології створення програмного забезпечення. Класифікація методологій створення програмного забезпечення. Особливості, переваги та недоліки існуючих методологій створення програмного забезпечення.

Поняття методології створення програмного забезпечення. Класифікація методологій створення програмного забезпечення. Каскадна методологія створення програмного забезпечення, її особливості, переваги та недоліки. Спіральна методологія створення програмного забезпечення, її особливості, переваги та недоліки. Інкрементна методологія створення програмного забезпечення, її особливості, переваги та недоліки. Ітераційна методологія створення програмного забезпечення, її особливості, переваги та недоліки. V-образна методологія створення програмного забезпечення, її особливості, переваги та недоліки.

Agile. Різновидності Agile методологій. Методологія Scrum створення програмного забезпечення, її особливості, переваги та недоліки. Методологія Kanban створення програмного забезпечення, її особливості, переваги та недоліки. Методологія Extreme programming створення програмного забезпечення, її особливості, переваги та недоліки.

Тема 3. Управління проектами. Командна розробка програмного забезпечення

Поняття проекту. Класифікація проектів. Поняття управління проектами. Методи та засоби управління проектами. Ролі в процесі створення програмних проектів.

Змістовий модуль 2. Аналіз вимог до програмного забезпечення та їх документування

Тема 4. Інженерія вимог до програмного забезпечення

Поняття вимоги до програмного забезпечення. Класифікація вимог до програмного забезпечення. Підходи до визначення вимог до програмного забезпечення. Задачі управління вимогами. Трасування вимог. Стратегії реалізації вимог.

Тема 5. Технічне завдання до створення програмного забезпечення

Поняття технічного завдання. Фактори, що визначають характеристики створюваного програмного забезпечення. Дії, що виконуються перед створенням технічного завдання. Обов'язкові розділи технічного завдання: введення, підстави для розробки, призначення розробки, вимоги до програмного забезпечення, вимоги до програмної документації, розрахунок технічних та економічних показників, стадії та етапи розробки, порядок контролю та прийому. Додатки. Приклад створення технічного завдання.

Тема 6. Специфікація вимог до програмного забезпечення

Поняття специфікації вимог до програмного забезпечення. Основні питання, які повинні розглядатися перед створенням специфікації. Характеристики правильно створеної специфікації. Розділи специфікації вимог до програмного забезпечення. Приклад створення специфікації вимог до програмного забезпечення.

Змістовий модуль 3. Об'єктно-орієнтований аналіз та проектування

Тема 7. Об'єктно-орієнтований аналіз та проектування

Поняття об'єктно-орієнтованого аналізу. Поняття об'єктно-орієнтованого проектування. Поняття предметної області та словника предметної області.

Підходи до класифікації об'єктів: класична категоризація, концептуальна кластеризація та теорія прототипів.

Підходи до аналізу об'єктно-орієнтованих систем: аналіз поведінки, аналіз предметної області, аналіз варіантів, стс-картки, неформальний опис та структурний аналіз.

Тема 8. Структурний аналіз

Особливості такого підходу до аналізу об'єктно-орієнтованих систем, як структурний аналіз. Цілі виконання структурного аналізу. Підготовка до виконання структурного аналізу.

Діаграма Сутність-зв'язок. Діаграма функціонального моделювання. Діаграма потоків даних.

Тема 9. Об'єктно-орієнтовані графічні нотації

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

Змістовий модуль 4. Базові діаграми мови моделювання UML

Тема 10. Мова моделювання UML

Історія виникнення мови моделювання UML, її особливості та призначення. Основні діаграми UML та їх графічне представлення. Загальні рекомендації з побудови діаграм UML.

Тема 11. Діаграма варіантів використання та діаграма аналізу

Особливості діаграми варіантів використання та діаграми аналізу, ціль їх побудови. Елементи та відношення на даних діаграмах. Рекомендації з побудови діаграми варіантів використання та діаграми аналізу.

Тема 12. Діаграма класів

Особливості діаграми класів, ціль її побудови. Елементи та відношення на даній діаграмі. Рекомендації з побудови діаграми класів.

Тема 13. Діаграми взаємодії

Діаграма послідовності та діаграма кооперації. Особливості даних діаграм, ціль їх побудови. Елементи та відношення на даних діаграмах. Рекомендації з побудови діаграм послідовності та діаграм кооперації.

Тема 14. Діаграми поведінки

Діаграма станів та діаграма діяльності. Особливості даних діаграм, ціль їх побудови. Елементи та відношення на даних діаграмах. Рекомендації з побудови діаграм станів та діаграм діяльності.

Тема 15. Діаграма компонентів

Особливості діаграми компонентів, ціль її побудови. Елементи та відношення на

даній діаграмі. Рекомендації з побудови діаграми компонентів.

Тема 16. Діаграма розгортання

Особливості діаграми розгортання, ціль її побудови. Елементи та відношення на даній діаграмі. Рекомендації з побудови діаграми розгортання.

7 Структура навчальної дисципліни

Назви змістових модулів і тем		Кількість годин для денної форми навчання				
		Всього	У тому числі			
			Лек.	Пр.	Лаб.	С.р.
1	2	3	4	5	6	7
6-й семестр						
Змістовий модуль 1. Базові поняття. Методології створення програмного забезпечення						
1	Вступ. Поняття та етапи життєвого циклу	22	2			20
2	Методології створення програмного забезпечення	21	2		4	15
3	Управління проектами	12	2			10
Разом за змістовим модулем 1		55	6		4	45
Змістовий модуль 2. Аналіз вимог до програмного забезпечення та їх документування						
4	Вимоги до програмного забезпечення	12	2			10
5	Технічне завдання	30	4		6	20
6	Специфікація вимог до програмного забезпечення	23	4		4	15
Разом за змістовим модулем 2		65	10		10	45
Разом за 6-й семестр		120	16		14	90
7-й семестр						
Змістовий модуль 3. Об'єктно-орієнтований аналіз та проектування						
7	Об'єктно-орієнтований аналіз та проектування	7	2			5
8	Структурний аналіз	20	2		4	10
9	Об'єктно-орієнтовані графічні нотації	19	4			15
Разом за змістовим модулем 3		44	8		4	30
Змістовий модуль 4. Базові діаграми мови моделювання UML						
10	Мова моделювання UML	7	2			4
11	Діаграма варіантів використання та діаграма аналізу	7	1		2	5

12	Діаграма класів	7	1	2	4
13	Діаграми взаємодії	7	1	2	5
14	Діаграми поведінки	7	1	2	4
15	Діаграма компонентів	5	1	1	4
16	Діаграма розгортання	2	1	1	4
Разом за змістовим модулем 4		42	8	10	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	Методології створення програмного забезпечення	4
2	Технічне завдання до створення програмного забезпечення	6
3	Специфікація вимог до програмного забезпечення	4
4	Опис та аналіз предметної області	4
5	Діаграма варіантів використання та діаграма аналізу	2
6	Діаграма класів	2
7	Діаграми взаємодії	2
8	Діаграми поведінки	2
9	Діаграма компонентів	1
10	Діаграма розгортання	1
Разом		28

9 Самостійна робота

№ з/п	Назва теми	Кількість Годин
1	Робота з конспектом лекцій, методичними вказівками, основною та додатковою літературою, підготовка до тестів	20
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»](#).

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 невиконанні 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 здобувачів вищої освіти Національного університету «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»»](#).

12 Розподіл балів, які отримують студенти

Поточний контроль за модулями

Модуль за тематичним планом дисципліни та форма контролю	Кількість балів
6-й семестр	
Змістовий модуль 1. Базові поняття. Методології створення програмного забезпечення	0...20
1Рішення завдань до захисту лабораторних робіт.	0...7
2Самостійність виконання лабораторних робіт.	0...7
3Своєчасність виконання лабораторних робіт.	0...6
Змістовий модуль 2. Аналіз вимог до програмного забезпечення та їх документування	0...20
1Рішення завдань до захисту лабораторних робіт.	0...7
2Самостійність виконання лабораторної роботи.	0...7
3Своєчасність виконання лабораторних робіт.	0...6
4Захист розрахунково-графічної роботи	0...20
Семестрова оцінка поточного контролю	0...60
7-й семестр	
Змістовий модуль 3. Об'єктно-орієнтований аналіз та проектування	0...30
1Рішення завдань до захисту лабораторної роботи.	0...10
2Самостійність виконання лабораторних робіт.	0...10
3Своєчасність виконання лабораторних робіт.	0...10
Змістовий модуль 4. Базові діаграми мови моделювання UML	0...30
1Рішення завдань до захисту лабораторної роботи.	0...10
2Самостійність виконання лабораторних робіт.	0...10
3Своєчасність 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 кожен 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, курсової роботи, диференційованого заліку	для заліку
90 – 100	A	відмінно	зараховано
82-89	B	добре	
75-81	C		
66-74	D	задовільно	
60-65	E		
0-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання

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 інструментальні засоби для реалізації лабораторних робіт, розрахунково-графічної роботи та курсової роботи – Visual paradigm.

14 Методичне забезпечення

1. Проектування програмного забезпечення. Методичні вказівки до лабораторного практикуму з дисципліни “Проектування програмного забезпечення” для студентів денної форми навчання спеціальності 121 – „Інженерія програмного забезпечення” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: НУ “Чернігівська політехніка”, 2021. — 49 с. – [Електронний ресурс]. – Режим доступу : <https://eln.stu.cn.ua/course/view.php?id=2576>, обмежений. – Заголовок з екрану.

2. Методичні рекомендації до виконання розрахунково-графічної роботи з дисципліни „Проектування програмного забезпечення” для студентів спеціальності 121 – „Інженерія програмного забезпечення” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: НУ “Чернігівська політехніка”, 2021. — 16с. – [Електронний ресурс]. – Режим доступу : <https://eln.stu.cn.ua/course/view.php?id=2576>, обмежений. – Заголовок з екрану.

3. Проектування програмного забезпечення. Методичні вказівки до курсового проектування з дисципліни “Проектування програмного забезпечення” для студентів денної форми навчання спеціальності 121 – „Інженерія програмного

забезпечення" / Укл. А. М. Акименко, І. В. Богдан — Чернігів: НУ “Чернігівська політехніка”, 2021. — 19 с. — [Електронний ресурс]. — Режим доступу : <https://eln.stu.cn.ua/course/view.php?id=2576>, обмежений. — Заголовок з екрану.

15 Рекомендована література

Базова

1. Життєвий цикл програмного забезпечення : навчальний посібник / Є.В. Левус, Т.А. Марусенкова, О.О. Нитребич. - Львів. : Видавництво Львівської політехніки, 2017. - 208 с.
2. Кучеров, Д.П. Інженерія програмного забезпечення : навчальний посібник / Д.П. Кучеров, Є.Б. Артамонов. - Київ. : НАУ, 2017. - 386 С.
3. Об'єктно-орієнтоване моделювання при проектуванні вбудованих систем і систем часу : навчальний посібник : затверджено МОН України / В.В. Литвинов, С.В. Голуб, К.М. Григор'єв, В.Ю. Жигульська. - Київ-Черкаси. : ІнтролігаТОР, 2011. - 511 С.
4. Бородкіна, І.Л. Інженерія програмного забезпечення : навч. посіб. / І.Л. Бородкіна, Г.О. Бородкін. - Київ. : ЦУЛ, 2019. - 204 С.
5. Левус, Є.В. Вступ до інженерії програмного забезпечення : навчальний посібник / Є.В. Левус, Н.Б. Мельник. - Львів. : Видавництво Львівської політехніки, 2018. - 246 С.

Допоміжна

6. Бородкіна І., Бородкин Г. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів / І. Бородкіна, Г. Бородкин – М: Центр навчальної літератури, 2018. – 204 с.
7. 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.
8. Мартин Фаулер. UML. Основы. 3-е издание / Мартін Фаулер, Кендалл Скотт – М.: Символ-плюс, 2018. – 192 с.
9. Роберт М. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.
10. 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
11. Леоненков О. Самовчитель UML. Ефективний інструмент моделювання інформаційних систем / О. Леоненков. – М.: БХВ-Петербург, 2001. – 304 с.
12. Лавріщева К.М. Програмна інженерія / К.М. Лавріщева –К.– 2008.–319 с.

16 Інформаційні ресурси

1. Система дистанційного навчання НУ «Чернігівська політехніка». Курс: Проектування програмного забезпечення. – [Електронний ресурс]. – Режим доступу : <https://eln.stu.cn.ua/course/view.php?id=2576>. – Назва з екрану.
2. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/uml/index.htm>. – Назва з екрану.
3. Software Engineering | Introduction to Software Engineering [Електронний ресурс]. – Режим доступу: URL: <https://www.geeksforgeeks.org/software-engineering-introduction-to-software-engineering/>. – Назва з екрану.
4. Стандарт IEEE 830-1998 [Електронний ресурс]. – Режим доступу: <https://standards.ieee.org/standard/830-1998.html>. – Назва з екрану.
5. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/java/index.htm>. – Назва з екрану.
6. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/cplusplus/index.htm>. – Назва з екрану.