

Національний університет «Чернігівська політехніка»
Навчально-науковий інститут електронних та інформаційних технологій
Кафедра *інформаційних технологій та програмної інженерії*

“ЗАТВЕРДЖУЮ”

Завідувач кафедри

Білоус Ірина Володимирівна

“_____” _____ 20__ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ (ВБ15)

Освітня програма «Інженерія програмного забезпечення»

Рівень вищої освіти – *перший (бакалаврський)*

Спеціальність *121 – Інженерія програмного забезпечення*

Мова навчання: *українська*

Статус дисципліни: *вибіркова*

Форма навчан.	Рік навч.	Сем	Розподіл годин				Разом	За тиждень		ІНДЗ	Контр.
			Всього ауд.	Лек	Лаб.	СРС		Ауд.	СРС		
Денна	3	6	30	16	14	90	120	1,85	5,6	РГР	I

Чернігів – 2021 рік

Робоча програма Архітектура програмного забезпечення
(назва навчальної дисципліни)

для студентів галузі знань 12 – «Інформаційні технології»
спеціальності 121 – «Інженерія програмного забезпечення»

Розробник робочої навчальної програми:

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 ББ15 Software architecture

2021/2022 Sem. 2

Course Description

The **subject of studying** is the software architecture design process, ie principles, frameworks, styles and tools for creating a software architecture.

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 architecture design.

The primary studying goals of the discipline:

- Introduction to the concept of software architecture design.
- Introduction to basic concept of software architecture design.
- Practical acquaintance with features of the implementation of design patterns.

Contents: software architecture design, architectural styles, architectural frames, client-server architecture, design patterns.

1 Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 4	Галузь знань 12 Інформаційні технології	Обов'язкова
Модулів – 1	Спеціальність: <i>121 – Інженерія програмного забезпечення</i> Освітньо-професійна програма: <i>Інженерія програмного забезпечення</i>	Рік підготовки:
Змістових модулів – 3		3-й
Індивідуальне науково-дослідне завдання – розрахунково-графічна робота		Семестр
Загальна кількість годин – 120		6-й
Тижневих годин: аудиторних – 1,8; самостійної роботи і індивідуальної студента – 5,6;	Освітньо-кваліфікаційний рівень: бакалавр	Лекції
		16
		Лабораторні
		14
		Самостійна робота
		90
Вид контролю:		
		Іспит

Примітка.

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

у 6-му семестрі – $1,875: 5,625=1:3$.

Навички, що необхідні для виконання лабораторних та домашніх завдань в межах дисципліни, студенти отримують з дисциплін «Java та C# технології прикладного програмування», «Бази даних», «Об'єктно-орієнтоване програмування», що викладалися раніше.

2 Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни “*Архітектура програмного забезпечення*” є формування науково-професійного *світогляду* бакалавра спеціальності 121 – *Інженерія програмного забезпечення* в області проектування програмного забезпечення. Предмет вивчення – **процес проектування архітектури програмного забезпечення**, тобто принципи, каркаси, стилі та інструменти створення архітектури програмного забезпечення.

Одним з найважливіших етапів створення програмного забезпечення є етап створення архітектури програмного забезпечення. Саме від вибору структурних елементів, їх інтерфейсів, а також особливостей їх взаємодії в першу чергу і залежить надійність, якість, а також безпека створюваного програмного забезпечення. 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 архітектури програмного забезпечення.

Під час вивчення дисципліни здобувач вищої освіти (ЗВО) має набутися 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.

ЗК7. Здатність працювати в команді.

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.

ФК19. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК21. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК24. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК25. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК26. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

ФК27. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК28. Здатність до алгоритмічного та логічного мислення.

3 Очікувані результати навчання з дисципліни

Навчальна дисципліна «Архітектура програмного забезпечення» має допомогти сформуванню наступних програмних результатів навчання.

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

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.

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.

ПР25. Мати навички виконання певних ролей в ІТ-проектах будь-якої складності.

Після вивчення дисципліни студенти **повинні знати:**

- принципи проектування архітектури програмного забезпечення;
- архітектурні шаблони, методи проектування і оцінювання архітектури, основні концепції архітектури програмного забезпечення;
- патерни та антипатерни проектування.

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

- орієнтуватися в типах сучасного програмного забезпечення;
- використовувати на практиці основні патерни проектування;
- створювати проектну документацію для опису, планування та підтримки програмного забезпечення.

4 Критерії оцінювання результатів навчання

До іспиту допускаються здобувачі вищої освіти, що виконали усі заплановані на семестр завдання з підсумковою оцінкою не менше 20 балів.

З тими здобувачами вищої освіти, 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:

- принципи проектування архітектури програмного забезпечення;
- основні патерни проектування;
- архітектурні шаблони, методи проектування і оцінювання архітектури, основні концепції архітектури програмного забезпечення.

2) вміє:

- орієнтуватися в типах сучасного програмного забезпечення;
- створювати проектну документацію для опису, планування та підтримки програмного забезпечення.

5 Засоби діагностики результатів навчання

Засобами оцінювання та методами демонстрування результатів навчання з дисципліни є поточний та семестровий контроль. Поточний контроль складається з опитувань, які проводяться під час лекцій, а також – захисту лабораторних та розрахунково-графічної роботи. Запитання для поточного контролю знаходяться у відповідних методичних рекомендаціях. Семестровий контроль проводиться у вигляді іспиту, запитання до якого на початку семестру розміщується у системі дистанційного навчання. Екзаменаційні білети знаходяться в пакеті документації на дисципліну.

6 Програма навчальної дисципліни

Змістовий модуль 1. Введення в архітектуру програмного забезпечення

Тема 1. Введення в архітектуру програмного забезпечення

Історія розвитку. Базові терміни. Елементи архітектури програмного забезпечення. Ключові принципи архітектури. Цілі вибору архітектури. Основні архітектурні каркаси.

Тема 2. Архітектурні шаблони та стилі

Поняття архітектурного стилю. Основні архітектурні стилі. Поєднання декількох архітектурних стилів. Архітектура клієнт/сервер. Компонентна архітектура.

Проектування на основі предметної області. Багатошарова архітектура. Архітектура, заснована на шині повідомлень. Сервісно-орієнтована архітектура. Об'єктно-орієнтована архітектура.

Змістовий модуль 2. Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення

Тема 3. Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення

Високорівневі та низькорівневі інструментальні засоби. Стандарт IEEE/AM81 для опису проекту. Контроль якості при виборі архітектури. Метрики для вибору архітектури.

Змістовий модуль 3. Патерни проектування програмного забезпечення

Тема 4. Структурні патерни проектування

Патерн Adapter, його основні задачі, особливості та приклад реалізації. Патерн Proxy, його основні задачі, особливості та приклад реалізації. Патерн Singleton, його основні задачі, особливості та приклад реалізації.

Тема 5. Породжуючі патерни проектування

Патерн Singleton, його основні задачі, особливості та приклад реалізації. Патерн Abstract factory, його основні задачі, особливості та приклад реалізації. Патерн Builder, його основні задачі, особливості та приклад реалізації. Патерн Prototype, його основні задачі, особливості та приклад реалізації. Патерн Factory method, його основні задачі, особливості та приклад реалізації.

Тема 6. Патерни поведінки

Патерн Memento, його основні задачі, особливості та приклад реалізації. Патерн Strategy, його основні задачі, особливості та приклад реалізації. Патерн Observer design pattern, його основні задачі, особливості та приклад реалізації.

Тема 7. Антипатерни проектування

Антипатерн Копі-паст, його суть та наслідки використання. Антипатерн Spaghetti, його суть та наслідки використання. Антипатерн Golden hummer, його суть та наслідки використання. Антипатерн Hard code, його суть та наслідки використання. Антипатерн Magic numbers, його суть та наслідки використання. Антипатерн Reinventing the wheel, його суть та наслідки використання. Антипатерн Blob, його суть та наслідки використання.

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
Модуль 1						
Змістовий модуль 1 Введення в архітектуру програмного забезпечення						
1	Введення в архітектуру програмного забезпечення	14	2		0	12
2	Архітектурні шаблони та стилі	16	2		2	12
Разом за змістовим модулем 1		30	4		2	24
Змістовий модуль 2 Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення						
3	Стандарти та інструментальні засоби при виборі архітектури програмного забезпечення	20	4		4	12
Разом за змістовим модулем 2		20	4		4	12
Змістовий модуль 3. Патерни проектування програмного забезпечення						
4	Структурні патерни проектування	15	2		2	11
5	Породжуючі патерни проектування	17	2		4	11
6	Патерни поведінки	14	2		2	10
7	Антипатерни проектування	12	2			10
8	РГР	-	-		-	12
Разом за змістовим модулем 3		70	8		8	54
Усього годин за дисципліну		120	16		14	90

8 Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Архітектурні стилі	2
2	Вибір інструментальних засобів в процесі проектування архітектури програмного забезпечення	4
3	Структурні патерни проектування	2
4	Породжуючі патерни проектування	4
5	Патерни поведінки	2
Разом		14

9 Самостійна робота

№ з/п	Назва теми	Кількість Годин
1	Вивчення мови UML як засобу моделювання програмного забезпечення	20
2	Вивчення шаблонів проектування як засобів об'єктно-орієнтованої розробки програмного забезпечення	20
3	Розглянути 4+1 View Model of Software Architecture	20
4	Розглянути стандарти мережевого протоколу HTTP стеку TCP/IP	10
5	Вивчення мови моделювання даних XML	8
6	РГР	12
Разом		90

10 Індивідуальні завдання

Метою розрахунково-графічних робіт є перевірка засвоєння студентами отриманого на лекціях та лабораторних роботах матеріалу та вміння самостійно вирішувати типові задачі за прослуханими темами. Варіанти завдань до даних робіт містяться в відповідних методичних вказівках.

Форми контролю виконання РГР

Вид роботи	Форма контролю	Кількість балів
Структура системи, схема та текст програми	1. Відповідність умовам завдання	0... 4
	2. Відповідність вимогам стандартів	0... 2
Пояснювальна записка	1. Обґрунтованість програмних рішень	0... 4
	2. Посилання на першоджерела	0... 1
	3. Відповідність оформлення вимогам	0... 2

	4. Своєчасність здачі	0... 2
Захист РГР	Самостійність виконання (відповіді на запитання або презентація)	0... 5
Разом		0... 20

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 здобувач може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру і до 40% підсумкової оцінки – на іспиті.

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*. *Іспит* проводиться під час сесії з трьома запитаннями: двома теоретичними (по 10 балів максимум за кожне) та одним практичним (20 балів максимум).

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 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 іспит.

В випадку повторного складання іспиту 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 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

	Змістовний модуль 1	M1=0...100
1	Робота на лекціях з вивчення лекційного матеріалу	0...10
2	Виконання та захист лабораторних робіт	0...40
3	Тестування з вивчення теоретичного матеріалу	0...50
	Змістовний модуль 2	M2=0...100
1	Робота на лекціях з вивчення лекційного матеріалу	0...10
2	Виконання та захист лабораторних робіт	0...40
3	Тестування з вивчення теоретичного матеріалу	0...50
	Змістовний модуль 3	M3=0...100
1	Робота на лекціях з вивчення лекційного матеріалу	0...10
2	Виконання та захист лабораторних робіт	0...40
3	Тестування з вивчення теоретичного матеріалу	0...50
	РГР	M4=0...100

Ітогова оцінка Оцінка1 поточного контролю обчислюється як

$$\text{Оцінка1} = 60\% * (M1 + M2 + M3 + M4) / 4.$$

Для захисту лабораторної роботи здобувач 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.

13 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

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

Лекційний матеріал подається у вигляді презентацій за допомогою медіапроектора. 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.

Особливістю виконання лабораторних робіт є застосування спеціального обладнання та прикладного програмного забезпечення навчальної лабораторії.

15 Методичне забезпечення

1. Архітектура програмного забезпечення. Методичні вказівки до лабораторного практикуму з дисципліни “Архітектура програмного забезпечення” для студентів денної форми навчання спеціальності 121 – „Інженерія програмного забезпечення” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: НУ “Чернігівська політехніка”, 2021. — 56 с. – Електронні дані – Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=4319>, обмежений. – Заголовок з екрану.

2. Методичні рекомендації до виконання розрахунково-графічної роботи з дисципліни „Архітектура програмного забезпечення” для студентів спеціальності 121 – „Інженерія програмного забезпечення” / Укл. А. М. Акименко, І. В. Богдан — Чернігів: НУ “Чернігівська політехніка”, 2021. — 27 с. – Електронні дані – Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=4319>, обмежений. – Заголовок з екрану.

16 Рекомендована література

Базова

1. Левус, Є.В. Життєвий цикл програмного забезпечення : навчальний посібник / Є.В. Левус, Т.А. Марусенкова, О.О. Нитребич. – Львів : Видавництво Львівської політехніки, 2017. - 207 С.
2. Кузьміна, О.О. Архітектура і програмне забезпечення ЕОМ : навч. посібник / О.О. Кузьміна. - Дніпропетровськ. : РВВ ДНУ, 2002. - 64 С.
3. Кучеров, Д. П. Інженерія програмного забезпечення : навч. посіб. / Д. П. Кучеров, Є. Б. Артамонов. – Київ : НАУ, 2017. – 386 с. : іл.

Допоміжна

1. Мартин Фаулер. Шаблоны корпоративных приложений. - К. “Диалектика”, 2019. - 544 с.
2. Гамма, Э. Паттерны объектно-ориентированного проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. - СПб. : Питер, 2020. - 448 С.
3. Брауде Э. Технология разработки программного обеспечения. - СПб. i Питер, 2004. - 655 с.
4. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.

17 Інформаційні ресурси

1. Система дистанційного навчання НУ «Чернігівська політехніка». Курс Архітектура ПЗ – [Електронний ресурс]. – Режим доступу : <https://eln.stu.cn.ua/course/view.php?id=4319>. – Назва з екрану.
2. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/uml/index.htm>. – Назва з екрану.
3. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/java/index.htm>. – Назва з екрану.
4. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/cplusplus/index.htm>. – Назва з екрану.