

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 (ВБ8.1)

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 – 2020 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:

старший викладач кафедри інформаційних технологій та програмної інженерії

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

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

Протокол від “__” _____ 20__ року № __

Завідувач кафедри *інформаційних технологій та програмної інженерії*

_____ (*I.V. Білоус*)
(підпис) (прізвище та ініціали)

Abstract

ESIEIT/PI BБ8.1 Software architecture

2020/2021 Sem. 6 2020/2021 Sem. 6

Course Description

The purpose of the course is to provide future professionals with knowledge of information models and systems for software development, relational and distributed databases, programming languages and software design.

The main tasks of studying the discipline "Software Architecture " is to obtain competencies for the student to participate in software design, implementation of databases for DBMS of various types, creation of complex processing procedures, user interfaces, management of software structure.

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» «Об'єктно-орієнтований аналіз та проектування», що викладалися раніше.

2 Мета та завдання навчальної дисципліни

Метою викладання дисципліни «Архітектура програмного забезпечення» є надання майбутнім фахівцям знань про інформаційні моделі та системи побудови програмного забезпечення, реляційні та розподілені бази даних, мови програмування та проектування програмного забезпечення.

Основними завданнями вивчення дисципліни «Архітектура та проектування програмного забезпечення» є отримання здобувачем вищої освіти компетенцій для того, щоб приймати участь у проектуванні програмного забезпечення, реалізації баз даних для СУБД різних типів, створення складних процедур обробки, інтерфейсів користувача, управління структурою 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.

ЗК7. Здатність працювати в команді.

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.

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.

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 до задачі методологію створення 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.

Після вивчення дисципліни студенти **повинні знати:**

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

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

- орієнтуватися в типах сучасного програмного забезпечення;
- 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;
- 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:

- іспит;
- розрахунково-графічна робота;
- презентації результатів виконаних завдань;
- завдання, які виконуються в навчальній лабораторії.

6 Програма навчальної дисципліни

Змістовий модуль 1. Введення в архітектуру та проектування програмного забезпечення

Тема 1. Введення в архітектуру та проектування програмного забезпечення

Історія розвитку. Базові терміни. Елементи Архітектури ПЗ. Ключові принципи архітектури. Інструменти моделювання архітектури. Мови опису, Представлення, Архітектурні каркаси. Приклади архітектурних стилів.

Тема 2. Техніка моделювання архітектури

Ідентифікація цілей та ключових сценаріїв. Огляд Програми. Ідентифікація ключових проблем. Вибір рішення – кандидату.

Змістовий модуль 2. Рівнева організація прикладу

Тема 3. Рівнева організація прикладу

Пошарова організація прикладу та виділення рівнів Представлення, Бізнес-логіки та Даних. Дизайн рівневої структури. Визначення необхідних рівнів, прийняття рішення про розподіл компонентів, визначення правил взаємодії між рівнями, визначення інтерфейсу між рівнями, вибір стратегій реалізації, Вибір протоколів взаємодії.

Змістовий модуль 3. Дизайн рівнів ПЗ

Тема 4. Дизайн рівню представлення.

Компоненти рівня представлення. Кешування, Комунікації, Композиція, Керування виключеннями, Специфічні проблеми рівню представлення.

Тема 5. Дизайн рівню бізнес-логіки.

Компоненти рівня бізнес-логіки. Специфічні проблеми рівню бізнес-логіки. Шаблони проектування.

Тема 6. Дизайн рівню даних.

Рекомендації щодо проектування рівню даних. Компоненти рівня даних. Специфічні проблеми рівню даних. Шаблони проектування.

Тема 7. Дизайн рівню сервісів.

Компоненти рівню сервісів. Специфічні проблеми рівню даних. Шаблони проектування. Робота з картами Google Map як приклад користування веб-сервісом. Огляд та здобуття навичок побудування API RESTful Web-сервісу.

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 Інформаційні моделі та системи						
1	Введення в архітектуру та проектування програмного забезпечення	14	2		0	12
2	Техніка моделювання архітектури	16	2		2	12
Разом за змістовим модулем 1		30	4		2	24
Змістовий модуль 2 Рівнева організація прикладу						
3	Рівнева організація прикладу	20	4		4	12
Разом за змістовим модулем 2		20	4		4	12
Змістовий модуль 3. Дизайн рівнів ПЗ						
4	Дизайн рівню представлення	15	2		2	11
5	Дизайн рівню бізнес-логіки	15	2		2	11
6	Дизайн рівню даних	14	2		2	10
7	Дизайн рівню сервісів	14	2		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
Модуль 1		
1	Побудування діаграми стану UML для гри RED DOG	2
2	Побудування діаграми дії UML для гри RED DOG	2
3	Побудування діаграми послідовностей UML для гри “Дуель” та опис варіантів використання	2
4	Робота з картами Google Map як приклад користування веб-сервісом. Огляд API RESTful Web-сервісу	2
5	Побудування колажу веб-сервісів Google Map та Twitter	2
6	Документування архітектури попередньо побудованої програми	4
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 студентів Чернігівського національного технологічного університету», погодженого вченою радою ЧНТУ (протокол № 9 від 26.10.2015 р.) та затвердженого наказом ректора ЧНТУ від 29.10.2015 р. №181.

Лекційний матеріал подається у вигляді презентацій за допомогою мультимедіа-проектора. Під час лекцій аналізуються проблемні ситуації, організується зворотний зв'язок з аудиторією шляхом формулювання запитань у режимі діалогу.

Під час лабораторних занять коротко розглядаються теоретичні положення відповідно до тематичного плану занять, докладно розбираються приклади, а надалі студентами самостійно вирішуються практичні задачі. Особливістю виконання лабораторних робіт є застосування спеціального обладнання та системного програмного забезпечення навчальних лабораторій.

Переліки екзаменаційних питань знаходяться в 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 студентів ЧНТУ».

Повторне складання екзамену з метою підвищення позитивної оцінки не дозволяється.

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 вищої освіти 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.

Для захисту розрахунково-графічної роботи здобувач вищої освіти повинен відповісти на всі контрольні запитання з методичних вказівок та на чотири за вибором викладача які стосуються безпосередньо варіанту завдання РГР.

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.

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

14 Методичне забезпечення

1. Методичні вказівки до розрахунково-графічної роботи з дисципліни «Архітектура програмного забезпечення» для студентів спеціальності 121 - "інженерія програмного забезпечення".

2. Методичні вказівки до лабораторних робіт з дисципліни "Архітектура програмного забезпечення " для студентів спеціальності 121 - "інженерія програмного забезпечення".

15 Рекомендована література

Базова

1. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення / Р. Мартін. – Фабула, 2019. – 416 с.

2. Фрімен Е., Робсон Е. Патерни проектування / Е. Фрімен, Е. Робсон. – :Фабула, 2020. – 672 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

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.