

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук



декан факультету інформаційних
технологій

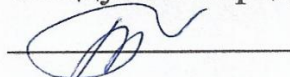
Глазунова О.Г.
2022 р.

«СХВАЛЕНО»

на засіданні кафедри комп'ютерних наук

Протокол № _____ від «__» _____ 20__ р.

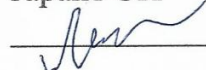
Завідувач кафедри

 Б. Л. Голуб

«РОЗГЛЯНУТО»

Гарант ОП 121 «Інженерія програмного
забезпечення»

гарант ОП

 Лялецький О.В.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА

«Інженерія програмного забезпечення»

за спеціальністю 121 «Інженерія програмного забезпечення»
галузі знань 12 «Інформаційні технології»

Факультет інформаційних технологій

Розробник: старший викладач кафедри комп'ютерних наук Бородкін Г.О.

Київ 2022

1 ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Галузь знань, спеціальність, освітній ступінь	
Галузь знань	12 «Інформаційні технології»
Спеціальність	121 «Інженерія програмного забезпечення»
Освітній ступінь	Бакалавр
Характеристика навчальної дисципліни	
Вид	Обов'язкова
Загальна кількість годин	150
Кількість кредитів ECTS	5
Кількість змістових модулів	4
Курсовий проект (робота) (за наявності)	194
Форма контролю	екзамен
Показники навчальної дисципліни для денної та заочної форм навчання	
	денна форма навчання
Рік підготовки (курс)	3
Семестр	6
Лекційні заняття	30 год.
Практичні, семінарські заняття	-
Лабораторні заняття	60 год.
Самостійна робота	60 год.
Індивідуальні завдання	-
Кількість тижневих аудиторних годин для денної форми навчання	6 год.

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою дисципліни «Архітектура та проектування програмного забезпечення» є вивчення сучасних технологій, методів та засобів проектування складних програмних систем.

Завдання дисципліни «Архітектура та проектування програмного забезпечення»:

- Ознайомлення з основними принципами побудови складних програмних систем;

- Ознайомлення та вивчення різних типів архітектур програмного забезпечення;
- Постановка та аналіз вимог до програмного продукту;
- Оволодіння навичками програмування на мові Python при роботі зі складними програмними системами;
- Розробка та застосування проектних рішень у власних програмних проектах.

У результаті вивчення навчальної дисципліни у студентів формуються наступні компетентності:

Загальні компетентності:

ЗК05. Здатність вчитися і оволодівати сучасними знаннями.

Спеціальні (фахові, предметні) компетентності

ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Формування цих компетентностей забезпечує досягнення наступних **програмних результатів:**

ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення

ПР08. Вміти розробляти людино-машинний інтерфейс.

ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

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.

ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

3. ПРОГРАМА ТА СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1.

Тема 1. Вступ до дисципліни. Створення складних програмних систем. Термінологія. Проблематика.

Причина виникнення науки та дисципліни «архітектура програмного забезпечення». Основна термінологія. Принципологія та концепції створення складних програмних систем.

Тема 2. Архітектура програмних систем. Типи архітектур.

Поняття архітектури програмних систем. Основні типи архітектур програмного забезпечення. Архітектурні стилі проектування програмного забезпечення.

Тема 3. Модульність та декомпозиція в проектуванні програмних систем.

Поняття модуля програмної системи. Модульність в проектуванні програмного забезпечення. Декомпозиція, правила декомпозиції при побудові складних програмних систем.

Тема 4. Життєвий цикл програмного продукту. Моделі, стадії, поняття.

Поняття та вимоги до життєвого циклу програмного забезпечення. Моделі життєвого циклу. Основні стадії та особливості життєвого циклу програмного забезпечення для різних моделей.

Змістовий модуль 2.

Тема 5. Постановка вимог та цілей до архітектури програмних систем. Аналіз вимог та розробка зовнішнього представлення проекту.

Підходи програмного інженера на етапі постановки вимог до розробки програмних систем. Комунікативна складова програмного інженера на етапі постановки вимог. Зовнішнє представлення та аналіз макету проекту.

Тема 6. Проектування архітектури програмних систем. Методи та методологія.

Основні підходи до проектування складних програмних систем. Стилi та підходи до проектування програмного забезпечення, розробка та прийняття проектних рішень програмного інженера. Реалізація програмної системи.

Тема 7. Аналіз архітектури програмного проекту.

Огляд та застосування методу аналізу компромісних архітектурних рішень АТАМ для високонавантажених програмних систем.

Змістовий модуль 3.

Тема 8. Атрибути якості програмного продукту.

Визначення атрибутів якості програмних систем на основі аналізу нефункціональних вимог до розробки програмного забезпечення.

Тема 9. Архітектурні шаблони. Вибір архітектурних шаблонів при проектуванні складних програмних систем.

Огляд основних патернів архітектури. Методологія вибору архітектурних патернів та подальше їх застосування в складних програмних системах. Патерни: клієнт-сервер, майстер-робітники, сервіс-орієнтована архітектура, гнучка архітектура, канали та фільтри і т.д.

Тема 10. Багат шаровий принцип побудови архітектури програмного забезпечення.

Представлення архітектури програмної системи у вигляді сукупності архітектурних шарів. Розглядається класична багат шарова модель, та модель open-layer.

Тема 11. Патерни проектування нижнього рівня. Класифікація. Принципи застосування.

Огляд основних низькорівневих патернів проектування (design patterns). Наведення їх класифікації за трьома типами. Прийоми та правила застосування патернів низькорівневого проектування кожного з типів.

Змістовий модуль 4.

Тема 12. Патерни, що породжують.

Огляд та правила застосування патернів, що призначені для створення нових об'єктів в програмній системі. Розглядаються наступні патерни: Builder, Abstract factory, Singleton.

Тема 13. Структурні патерни.

Огляд та правила застосування патернів, призначених для компонування програмної системи на основі об'єктів та класів. Розглядаються наступні структурні патерни: Adapter, Facade, Bridge.

Тема 14. Патерни поведінки.

Огляд та правила застосування патернів, призначених для регламентації поведінки об'єктів в програмній системі. Розглядаються наступні патерни поведінки: Iterator, Template method, Strategy.

Тема 15. Оцінка якості архітектури програмного продукту.

Оцінка відповідності архітектури програмної системи до нефункціональних вимог та визначених атрибутів якості.

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	тижні	усього	у тому числі					усього	у тому числі					
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Змістовий модуль 1.														
Тема 1. Вступ до дисципліни. Створення складних програмних систем. Термінологія. Проблематика.		10	2		4		4							
Тема 2. Архітектура програмних систем. Типи архітектур.		10	2		4		4							
Тема 3. Модульність та декомпозиція в проектуванні програмних систем.		10	2		4		4							
Тема 4. Життєвий цикл програмного продукту. Моделі, стадії, поняття.		10	2		4		4							
Разом за змістовим модулем 1		40	8		16		16							
Змістовий модуль 2.														
Тема 5. Постановка вимог та цілей до архітектури програмних систем. Аналіз вимог та розробка зовнішнього представлення проекту.		10	2		4		4							
Тема 6. Проектування архітектури програмних систем. Методи та методологія.		10	2		4		4							
Тема 7. Аналіз архітектури програмного проекту.		10	2		4		4							
Разом за змістовим		30	6		12		12							

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	тижні	усього	у тому числі					усього	у тому числі					
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
модулем 2														
Змістовий модуль 3.														
Тема 8. Атрибути якості програмного продукту.		10	2		4		4							
Тема 9. Архітектурні шаблони. Вибір архітектурних шаблонів при проектуванні складних програмних систем.		10	2		4		4							
Тема 10. Багатошаровий принцип побудови архітектури програмного забезпечення.		10	2		4		4							
Тема 11. Патерни проектування нижнього рівня. Класифікація. Принципи застосування.		10	2		4		4							
Разом за змістовим модулем 3		40	8		16		16							
Змістовий модуль 4.														
Тема 12. Патерни, що породжують.		10	2		4		4							
Тема 13. Структурні патерни.		10	2		4		4							
Тема 14. Патерни поведінки.		10	2		4		4							
Тема 15. Оцінка якості архітектури програмного продукту.		10	2		4		4							
Разом за змістовим модулем 4		40	8		16		16							

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	тижні	усього	у тому числі					усього	у тому числі					
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	
Усього годин	150		30		60		60							
Курсовий проект (робота) з _____ _____ (якщо є в робочому навчальному плані)	194		-	-	-		-		-	-	-		-	

4. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин
1	Підготовчий етап, вивчення мови Python на прикладі з використанням бібліотек NumPy та matplotlib.	4
2	Вибір та розробка стилю архітектури проекту. Розробка програми мовою Python, використовуючи обраний архітектурний вид.	4
3	Проведення декомпозиції програмних модулів. Реалізація мовою Python.	4
4	Розробка основних етапів проектування та етапів реалізації програмної системи.	4
5	Аналіз та розробка зовнішнього інтерфейсу проекту. Постановка цілей проекту.	6
6	Проектування програмної системи обраної архітектури.	6
7	Проаналізувати архітектуру програмного проекту по методу АТАМ.	4
8	Аналіз та прийняття рішень щодо атрибутів якості програмної системи.	6
9	Проектування архітектурних шарів програмної системи.	6
10	Вибір архітектурних шаблонів при проектуванні програмної системи.	6
11	Проектування нижнього рівня. Застосування низькорівневих шаблонів проектування.	4
12	Оцінка програмної архітектури на відповідність встановленим атрибутам якості.	6

5 КОНТРОЛЬНІ ПИТАННЯ, КОМПЛЕКТИ ТЕСТІВ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ЗАСВОЄННЯ ЗНАТЬ СТУДЕНТАМИ

Контрольні питання для визначення рівня засвоєння знань студентами.

1. Дайте визначення поняттям «архітектура програмного забезпечення», «програмна система», «проектування програмної системи», «програмна інженерія». Проаналізуйте значення архітектури програмного забезпечення при проектуванні складних програмних систем.
2. Проаналізуйте та перелічте роль архітектора програмної архітектури. Комуникативні зв'язки архітектора програмного забезпечення.
3. Які фактори впливають на вибір архітектури програмного забезпечення?
4. Охарактеризуйте основні типи програмних архітектур. Дайте класифікацію їх за типами.
5. Розкрийте поняття «Модульні види архітектури програмного забезпечення». Декомпозиція та правила декомпозиції програмних модулів.
6. Модульна розробка програмних систем мовою Python. Створення програмних модулів мовою Python.
7. Охарактеризуйте поняття життєвого циклу програмного забезпечення. Опишіть основні етапи життєвого циклу.
8. Що таке проектування архітектури програмних систем? Дати визначення архітектурного стилю. Навести приклади.
9. Назвати та описати методи оцінки складності ієрархічних структур.
10. Які ви знаєте методи аналізу програмної архітектури? Коротко описати їх.
11. Атрибути якості архітектури складних програмних систем.
12. Дайте визначення архітектурних рівнів програмної системи. Відкриті та закриті багаторівневі архітектури.
13. Які ви знаєте архітектурні патерни програмного забезпечення? Охарактеризувати їх.
14. Надати класифікацію патернам проектування нижнього рівня. Охарактеризувати кілька патернів кожної групи.

Комплект тестів для визначення рівня засвоєння знань студентами

знаходиться за посиланням:

<https://elearn.nubip.edu.ua/course/view.php?id=2524>

6 МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни використовуються такі методи навчання:

М1. Лекція (проблемна, інтерактивна).

М2. Лабораторна робота – для використання набутих знань до розв'язування практичних завдань.

М3. Проблемне навчання – створення проблемної ситуації для зацікавленого і активного сприйняття матеріалу.

М4. Проектне навчання (індивідуальне).

М8. Дослідницький метод.

7 ФОРМИ КОНТРОЛЮ

МК1. Тестування.

МК4. Методи усного контролю.

МК5. Екзамен.

МК7. Звіт

8. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамен та заліки у НУБіП України» (наказ про уведення в дію від 27.12.2019 р. No 1371).

Поточний контроль		Рейтинг з навчальної роботи $R_{НР}$	Рейтинг з додаткової роботи $R_{ДР}$	Рейтинг штрафний $R_{ШТР}$	Підсумкова атестація (екзамен чи залік)	Загальна кількість балів
Змістовий модуль 1	Змістовий модуль 2					
0-100	0-100	0-70	0-20	0-5	0-30	0-100

Шкала оцінювання

Рейтинг здобувача вищої освіти, бали	Оцінка національна за результати складання	
	екзаменів	заліків
90-100	відмінно	зараховано
74-89	добре	
60-73	задовільно	
0-59	незадовільно	не зараховано

Для визначення рейтингу студента із засвоєння дисципліни $R_{ДИС}$ (до 100 балів) одержаний рейтинг з атестації $R_{АТ}$ (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{НР}$ (до 70 балів): $R_{ДИС} = R_{НР} + R_{АТ}$.

Рейтинг з додаткової роботи $R_{ДР}$ додається до $R_{НР}$ і не може перевищувати 20 балів. Він визначається лектором і надається студентам рішенням кафедри за виконання робіт, які не передбачені навчальним планом, але сприяють підвищенню рівня знань студентів з дисципліни.

Рейтинг штрафний R_{ШТР} не перевищує 5 балів і віднімається від **R_{НР}**. Він визначається лектором і вводитья рішенням кафедри для студентів, які матеріал змістового модуля засвоїли невчасно, не дотримувалися графіка роботи, пропускали заняття тощо.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

ЕНК по дисципліні знаходиться за електронною адресою:
<https://elearn.nubip.edu.ua/course/view.php?id=2524>

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

– основна:

1. Л. Баас, П. Клементс, Р. Кацман. Архитектура программного обеспечения на практике. 2006. 576 с.
2. Л Гагарина, П Федоров, А Федоров. Введение в архитектуру программного обеспечения. Учебное пособие. 2016. 320 с.
3. Станислав Назаров. Архитектура и проектирование программных систем. 2016. 376 с.
4. Ф. Брукс. Проектирование процесса проектирования. Записки компьютерного эксперта. 2017. 464с.
5. Sandro Tosi. Matplotlib for Python Developers. 2009. 307р.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования
7. Д.Марка, К.МакГоуэн. SADT. Методология структурного анализа и проектирования.

– допоміжна:

1. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
2. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. – М.: Альт Линукс, 2010. — 126 с. (Библиотека ALT Linux).
3. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения. – СПб.: Питер, 2002. – 496 с.: ил.
4. Коберн А. Современные методы описания функциональных требований к системам. М: Издательство «Лори», 2002. 263 с.: ил.
5. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. СПб: Символ-Плюс, 2006. – 304 с., ил.

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. ЕНК по дисципліні знаходиться за електронною адресою:
<https://elearn.nubip.edu.ua/course/view.php?id=2524>
2. Tproger. Інтернет-ресурс для вивчення мови python, режим доступу:
<https://tproger.ru/digest/10-useful-python-resources/>
3. Matplotlib. Офіційна сторінка бібліотеки, режим доступу:
<https://matplotlib.org>
4. Визначення архітектури додатків за допомогою Rational Software Architect режим доступу: <https://www.ibm.com/developerworks/ru/library/r-define-application-architecture-rational-software-architect-2/index.html>