

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»

Декан факультету інформаційних технологій

_____ О. Г. Глазунова

«_____» _____ 20 ____ р.

РОЗГЛЯНУТО І СХВАЛЕНО

на засіданні кафедри комп'ютерних наук

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

Завідувач кафедри

_____ Б. Л. Голуб

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Спеціальність – 121 «Інженерія програмного забезпечення»

Факультет інформаційних технологій

Розробники: доц., к.т.н. кафедри комп'ютерних наук Голуб Б. Л.,
асист. кафедри комп'ютерних наук Пономаренко Р. М.

Київ – 2019 р.

1. Опис навчальної дисципліни

АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

(назва)

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	90
Кількість кредитів ECTS	3
Кількість змістових модулів	2
Курсовий проект (робота) (якщо є в робочому навчальному плані)	так
Форма контролю	іспит
Показники навчальної дисципліни для денної та заочної форм навчання	
	денна форма навчання
Рік підготовки	4
Семестр	8
Лекційні заняття	12 год
Практичні, семінарські заняття	
Лабораторні заняття	24год.
Самостійна робота	54год.
Індивідуальні завдання	
Кількість тижневих годин для денної форми навчання: - аудиторних - самостійної роботи студента	3 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 «Архітектура та проектування програмного забезпечення» студент повинен

знати:

- Основні етапи, принципи та стилі побудови складних програмних систем;
- Життєвий цикл програмного забезпечення, стадії у різних моделях життєвого циклу;
- Принципи та методи проектування різних архітектур у складних програмних системах;
- Архітектурні рівні (layers) програмних систем;
- Основні архітектурні патерни та патерни проектування нижнього рівня;

вміти:

- Проводити постановку вимог до програмного забезпечення;
- Приймати рішення щодо вибору архітектури програмної системи;
- Розробляти інтерфейс програмних модулів, проводити декомпозицію;
- Приймати рішення щодо застосування та впровадження патернів різних рівнів абстракції;
- Реалізувати проектні рішення за допомогою мови програмування Python;

- *Проводити аналіз архітектури програмного забезпечення на предмет відповідності атрибутам якості.*

3. Програма та структура навчальної дисципліни для:

- повного терміну денної (заочної) форми навчання;
- скороченого терміну денної (заочної) форми навчання.

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.

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.

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. Багатошаровий принцип побудови архітектури програмного забезпечення.

Представлення архітектури програмної системи у вигляді сукупності архітектурних шарів. Розглядається класична багатошарова 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.

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.

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.	2		2		4								
Тема 2. Архітектура програмних систем. Типи архітектур.	2		2		4								
Тема 3. Модульність та декомпозиція в проектуванні програмних систем.	2		2		4								
Тема 4. Життєвий цикл програмного продукту. Моделі, стадії, поняття.	2		2		4								
Разом за змістовим модулем 1	8		8		16								
Змістовий модуль 2.													
Тема 5. Постановка вимог та цілей до архітектури програмних систем. Аналіз вимог та розробка зовнішнього представлення проекту.	3		3		6								
Тема 6. Проектування архітектури програмних систем. Методи та методологія.	2		2		4								
Тема 7. Аналіз архітектури програмного проекту.	2		2		4								
Разом за змістовим модулем 2	7		7		14								
Змістовий модуль 3.													
Тема 8. Атрибути якості програмного продукту.	1		3		3								
Тема 9. Архітектурні шаблони. Вибір архітектурних шаблонів при проектуванні складних програмних систем.	2		3		3								

Тема 10. Багатошаровий принцип побудови архітектури програмного забезпечення.	1		3		3								
Тема 11. Патерни проектування нижнього рівня. Класифікація. Принципи застосування.	2		3		3								
Разом за змістовим модулем 3	6		12		12								
Змістовий модуль 4.													
Тема 12. Патерни, що породжують.	1		3		3								
Тема 13. Структурні патерни.	2		3		3								
Тема 14. Патерни поведінки.	1		3		3								
Тема 15. Оцінка якості архітектури програмного продукту.	2		3		3								
Разом за змістовим модулем 4	6		12		12								
Усього годин	93		39		54								
Курсовий проект (робота) з _____ (якщо є в робочому навчальному плані)	34		-	-	-		-		-	-	-		-
Усього годин	127		39		54								

4. Теми семінарських занять

Не передбачені програмою

5. Теми практичних занять

Не передбачені програмою

6. 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.	4

	Реалізація мовою Python.	
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.	4
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.	
12	Оцінка програмної архітектури на відповідність встановленим атрибутам якості.	6

7. Контрольні питання, комплекти тестів 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. Комунікативні 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. Модулярна 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.

8. Методи навчання.

Вербальні методи навчання: лекції, дискусії зі студентами, аудиторний захист доповідей для самостійного опрацювання.

Наочні методи навчання: представлення ілюстративних матеріалів для, проектування програмного продукту та матеріалів для роботи з середовищем розробки.

Практичні методи навчання: лабораторні заняття (робота з програмним забезпеченням, аналіз архітектурних та проектних рішень програмних систем, робота з програмними системами).

9. Форми контролю.

- Захист лабораторних робіт;
- Дві письмові модульні контрольні (за матеріалами змістовних модулів);
- Самостійна робота з вивчення окремих тематик (поза межами основних тем курсу), доповіді (представлення аудиторії вивчених матеріалів);
- Залік;
- Екзамен;
- Захист курсового проекту.

10. Розподіл балів, які отримують студенти. Оцінювання студента відбувається згідно положенням «Про екзамени та заліки у НУБіП України» від 27.02.2019р. протокол №7

Оцінка національна	Рейтинг здобувача вищої освіти, бали
Відмінно	90-100
Добре	74-89
Задовільно	60-73
Незадовільно	0-59

Для визначення рейтингу студента (слухача) із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{\text{нр}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{нр}} + R_{\text{ат}}$.

11. Методичне забезпечення

Матеріали лекційних (у вигляді презентацій) та лабораторних (у вигляді теоретичних відомостей та методичних рекомендацій щодо виконання лабораторних робіт) занять.

12. Рекомендована література

– основна:

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

– допоміжна:

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

13. Інформаційні ресурси

1. Tproger. Інтернет-ресурс для вивчення мови python, режим доступу: <https://tproger.ru/digest/10-useful-python-resources/>
2. Matplotlib. Офіційна сторінка бібліотеки, режим доступу: <https://matplotlib.org>
3. 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>