



## СИЛАБУС ДИСЦИПЛІНИ «АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Ступінь вищої освіти - Бакалавр  
Спеціальність 121 «Інженерія програмного забезпечення»  
Освітня програма «Інженерія програмного забезпечення»  
Рік навчання 4, семестр 7,8  
Форма навчання денна  
Кількість кредитів ЄКТС 5  
Мова викладання українська

Лектор курсу

Голуб Белла Львівна, к.т.н., доцент ([портфоліо](#))

Контактна інформація  
лектора (e-mail)

Кафедра комп'ютерних наук,  
навчальний корпус 15, к. 236, 237,  
тел.: (044) 527-87-23  
E-mail: [bellalg@it.nubip.edu.ua](mailto:bellalg@it.nubip.edu.ua)

Сторінка курсу в eLearn

<https://elearn.nubip.edu.ua/enrol/index.php?id=2524>

### ОПИС 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 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.

## СТРУКТУРА КУРСУ

Тема	Години (лекції/лабораторні)	Результати навчання	Завдання	Оцінювання
<b>1-й семестр</b>				
<b>Модуль 1</b>				
<b>Тема 1.</b> Вступ до дисципліни. Створення складних програмних систем. Термінологія. Проблематика.	2/4	Розуміти призначення науки та дисципліни. Знати основну термінологію дисципліни. Розуміти концепції створення складних програмних систем	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 2.</b> Архітектура програмних систем. Типи архітектур	2/4	Знати основні типи архітектур програмного забезпечення. Відрізнити основні стилі проектування програмного забезпечення	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 3.</b> Модульність та декомпозиція в проектуванні програмних систем	2/4	Розуміти поняття модуля програмної системи. Уміти проводити декомпозицію при побудові складних програмних систем	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 4.</b> Життєвий цикл програмного продукту. Моделі, стадії, поняття	2/4	Знати вимоги до життєвого циклу програмного забезпечення. Розрізнити моделі життєвого циклу та розуміти особливості стадій у кожному з них.	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
Модульний контроль №1			Тестування, виконання завдань	
<b>Модуль 2</b>				
<b>Тема 5.</b> Постановка вимог та цілей до архітектури програмних систем. Аналіз вимог та розробка зовнішнього представлення проекту.	4/6	Застосовувати підходи програмного інженера на етапі постановки вимог до розробки програмних систем. Проводити аналіз макету проекту	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 6.</b> Проектування	2/4	Знати основні підходи до проектування	Здача лабораторної	

Тема	Години (лекції/лабораторні)	Результати навчання	Завдання	Оцінювання
архітектури програмних систем. Методи та методологія		складних програмних систем. Уміти розробляти та приймати проектні рішення програмного інженера	роботи. Опитування теоретичного матеріалу.	
<b>Тема 7.</b> Аналіз архітектури програмного проекту	1/4	Використовувати метод аналізу компромісних архітектурних рішень АТАМ для високонавантажених програмних систем	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
Модульний контроль №2			Тестування, виконання завдань	
<b>Всього</b>				<b>70</b>
<b>Залік</b>				<b>30</b>
<b>Всього за 1-й семестр</b>				<b>100</b>
<b>2-й семестр</b>				
<b>Модуль 3</b>				
<b>Тема 8.</b> Атрибути якості програмного продукту	3/3	Уміти визначати атрибути якості програмних систем на основі аналізу нефункціональних вимог до розробки програмного забезпечення	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 9.</b> Архітектурні шаблони. Вибір архітектурних шаблонів при проектуванні складних програмних систем	3/3	Знати основні патерни архітектури. Уміти обирати та застосовувати патерни для складних програмних систем	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 10.</b> Багатошаровий принцип побудови архітектури програмного забезпечення	3/3	Знати архітектурні шари програмної системи. Відрізнити класичну багатошарову модель, та модель open-layer	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 11.</b> Патерни проектування нижнього рівня. Класифікація. Принципи	3/3	Розуміти різницю між основними низькорівневими патернами проектування. Уміти обирати патерн відповідно до потреб	Здача лабораторної роботи. Опитування теоретичного матеріалу.	

Тема	Години (лекції/лабораторні)	Результати навчання	Завдання	Оцінювання
застосування		та використовувати за правилами застосування		
Модульний контроль №3			Тестування, виконання завдань	
<b>Модуль 4</b>				
<b>Тема 12.</b> Патерни, що породжують	3/3	Розуміти різницю між породжувальними патернами та призначення кожного (Builder, Abstract factory, Singleton). Уміти обирати патерн залежно від потреб	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 13.</b> Структурні патерни	3/3	Розуміти різницю між структурними патернами та призначення кожного (Adapter, Facade, Bridge). Уміти обирати патерн залежно від потреб	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 14.</b> Патерни поведінки	3/3	Розуміти різницю між патернами поведінки та призначення кожного (Iterator, Template method, Strategy). Уміти обирати патерн залежно від потреб	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
<b>Тема 15.</b> Оцінка якості архітектури програмного продукту	3/3	Уміти проводити оцінку відповідності архітектури програмної системи до нефункціональних вимог та визначених атрибутів якості	Здача лабораторної роботи. Опитування теоретичного матеріалу.	
Модульний контроль №4			Тестування, виконання завдань	
<b>Всього</b>				<b>70</b>
<b>Іспит</b>				<b>30</b>
<b>Всього за 2-й семестр</b>				<b>100</b>

#### ПОЛІТИКА ОЦІНЮВАННЯ

<b>Політика щодо дедлайнів та перескладання:</b>	Терміни виконання робіт визначені в ЕНК. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу лектора за наявності поважних причин (наприклад, лікарняний).
--	---

<b><i>Політика щодо академічної доброчесності:</i></b>	Списування під час контрольних робіт та екзаменів заборонені (в т.ч. із використанням мобільних девайсів). Курсові роботи, реферати повинні мати коректні текстові посилання на використану літературу
<b><i>Політика щодо відвідування:</i></b>	Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в он-лайн формі за погодженням із деканом факультету)

### **ШКАЛА ОЦІНЮВАННЯ СТУДЕНТІВ**

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