



Компоненти програмної інженерії. Частина 3 .

Архітектура програмного забезпечення.

Програма навчальної дисципліни (Syllabus)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший рівень (бакалавр)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення комп'ютерних систем
Статус дисципліни	Нормативна
Форма навчання	Очна (денна)
Рік підготовки, семестр	2 рік (4 семестр)
Обсяг дисципліни	5 кредити (150 год.), Лекції 18 (36 год.), Лабораторні 18 (36 год.)
Семестровий контроль/ контрольні заходи	Екзамен
Розклад занять	https://roz.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Керівник курсу: завідувач каф. ОТ, д. т. н., проф. Стіренко С. Г., sergii.stirenko@gmail.com Лекції: проф., д.т.н. Богдан Корнієнко , b.korniyenko@kpi.ua Ст. викл. Тимур Шемсєдинов, timur.shemsedinov@gmail.com Лабораторні : проф. , д.т.н. Богдан Корнієнко , b.korniyenko@kpi.ua Ст. викл. Тимур Шемсєдинов, timur.shemsedinov@gmail.com Асистент Мазур Роман Федорович, mazur.roman@gmail.com
Розміщення курсу	https://classroom.google.com

Програма навчальної дисципліни

1 Опис курсу, цілі і завдання, результати навчання

Вивчення дисципліни «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення» займає важливе місце в структурі засвоєння знань освітньої програми «Інженерія програмного забезпечення комп'ютерних систем». Процес проектування програмного забезпечення передбачає розробку архітектури для зменшення складності системи шляхом абстракції та розподілу повноважень. Архітектура програмного забезпечення поєднує різні точки зору на систему, що є вагомим аргументом на користь необхідності та доцільності розробки архітектури програмного забезпечення. Ця дисципліна формує системний підхід до проектування програмного забезпечення та інженерний світогляд IT-фахівця.

Метою вивчення дисципліни «Складові програмної інженерії. Частина 3. Моделювання програмного забезпечення. Аналіз вимог до програмного забезпечення» є теоретична та практична підготовка студентів, яка має забезпечити набуття ними базових знань у галузі сучасних технологій проектування та вивчення архітектури програмного забезпечення, набуття практичних навичок реалізації програмних систем, основ моделювання та аналізу програмних систем, аналізу розробки, специфікації та управління вимогами.

Предметом вивчення дисципліни є сучасні методи, засоби та технології розробки програмного забезпечення.

Згідно з вимогами ОПП дисципліна “Компоненти програмної інженерії” має забезпечувати набуття здобувачами компетентностей та програмних результатів навчання: ФК01-05, ФК07, ФК08, ФК11-13, ПРН01-04, ПРН06-11, ПРН13-20. Так, зокрема, в результаті вивчення модуля “Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення” здобувачі набувають компетентності:

- здатність формувати та забезпечувати вимоги до якості програмного забезпечення відповідно до вимог замовника, специфікацій та стандартів;
- здатність дотримуватись специфікацій, стандартів, правил та рекомендацій у професійній сфері при реалізації процесів життєвого циклу;
- здатність застосовувати фундаментальні та міждисциплінарні знання для успішного вирішення завдань програмної інженерії;
- здатність оцінювати та враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності;
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення та супроводу програмного забезпечення та визнання важливості навчання впродовж життя;
- здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей та підходів до розробки програмного забезпечення.

За результатами вивчення навчальної дисципліни «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення» необхідно отримати такі **знання** :

- здатність брати участь у проектуванні програмного забезпечення, включаючи моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- здатність дотримуватись специфікацій, стандартів, правил та рекомендацій у професійній сфері при реалізації процесів життєвого циклу;
- здатність оцінювати та враховувати економічні, соціальні, технологічні та екологічні фактори, що впливають на сферу професійної діяльності.

Навички, які необхідно набути в рамках вивчення навчальної дисципліни «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення»:

- проектування, розробка та тестування програмного забезпечення різних типів;
- розробляти технічну документацію до програмного забезпечення;
- знати та застосовувати професійні стандарти та інші нормативно-правові документи у сфері програмної інженерії;
- вміти розробляти людино-машинний інтерфейс;
- знати, аналізувати, обирати, грамотно застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) та цілісності даних відповідно до прикладних завдань, що вирішуються, та створюваних програмних систем.

Таке поєднання загальних і спеціальних компетентностей, теоретичних і практичних знань, умінь і навичок сприяє підвищенню професійного рівня бакалаврів для здійснення ефективної діяльності у сфері розв'язування завдань програмного забезпечення.

2 Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Необхідні дисципліни: «Алгоритми та структури даних», «Основи програмування», «Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії», «Компоненти програмної інженерії. Частина 2. Моделювання програмного забезпечення. Аналіз вимог до програмного забезпечення», «Об'єктно-орієнтоване програмування».

Модуль «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення» необхіден для вивчення наступних частин дисципліни «Компоненти програмної інженерії», а також може бути корисним при вивченні дисциплін «Методології і технології розроблення програмного забезпечення», «Управління ризиками та якістю проектів».

3 Структура кредитного модуля

Перелік основних тем, що входять до навчальної програми дисципліни «Компоненти програмної інженерії. Частина 3. Архітектура програмного забезпечення»:

Розділ1. Варіанти архітектур програмної системи.

Тема 1.1. Поняття про архітектуру програмного забезпечення.

Тема 1.2. Архітектурні зразки, еталонні моделі та варіанти архітектур.

Тема 1.3. Архітектурні споруди та уявлення.

Тема 1.4. Зв'язки між структурами.

Тема 1.5. Архітектура на основі портів.

Тема 1.6. Архітектура незалежних компонентів.

Тема 1.7. Сервісно-орієнтовані архітектури.

Розділ2. Проектування архітектури програмного забезпечення. Методика проектування. Модульність.

Тема 2.1. Модулі, модульний підхід до інтерфейсу.

Тема 2.2. Обґрунтування модульності.

Тема 2.3. Внутрішня характеристика модуля - зв'язність.

Тема 2.4. З'єднання модулів є зовнішньою характеристикою модуля.

Тема 2.5. Багаторівнева архітектура

Тема 2.6. Логічна структура багаторівневої архітектури.

Тема 2.7. Шаблони проектування багатошарової архітектури.

Розділ3. Архітектура веб-додатків.

Тема3.1. Компоненти веб-додатку.

Тема 3.2. Архітектура багаторівневого веб-додатку.

Тема 3.3. Архітектура змішаного веб-додатку.

Тема 3.4. Архітектура багаторівневого веб-додатку на основі технології CORBA.

Тема 3.5. Архітектура веб-додатків з інтерфейсами OLE DB, ADO та ODBC.

Розділ4. Патерни проектування.

Тема 4.1. Поняття візерунка.

Тема 4.2. Види візерунків.

Тема 4.3. Генерування шаблонів.

Тема 4.4. Структурні закономірності.

Тема 4.5. Патерни поведінки.

Тема 4.6. Поняття про розетку.

Тема 4.7. Функції розеток.

Розділ 5. Прийоми об'єктно-орієнтованого проектування .

Тема 5.1. Основні принципи об'єктно-орієнтованої архітектури (ООА).

Тема 5.2. Переваги ООА.

Тема 5.3. Поняття та елементи ООА.

Тема 5.4. Поняття технології COM.

Тема 5.5. Проблеми, які вирішуються за допомогою технології COM.

4 Навчальні ресурси та матеріали

Основна література :

1. Прессман, Роджер (2010) *Розробка програмного забезпечення: підхід практиків*, McGraw Hill, Нью-Йорк, Нью-Йорк.
2. Соммервіль, Ян (2011) *Розробка програмного забезпечення* , Addison-Wesley, Бостон, Массачусетс.
3. Stephens, Rod (2015) *Beginning Software Engineering* , Wrox .
4. Цуй , Френк, Орландо Карам і Барбара Бернал (2013) *Основи програмної інженерії* , Jones & Bartlett Learning, Садбері, Массачусетс.
5. Pfleeger, Shari (2001) *Розробка програмного забезпечення : теорія та практика* , Prentice Hall, Upper Saddle River, NJ.

Додаткова :

- 1 Larman, C. (2005) *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and iterative Development*, Pearson
- 2 Ambler, S. (2002) *AgileModeling: Effective Practices for Extreme Programming and the Unified Process*, NewYork, John Wiley&Sons.
- 3 Bass, D.L., Clements, D.P. and Kazman, D.R. (2012) *Software Architecture in Practice*, 3rd edn, Upper Saddle River, NJ, Addison Wesley
- 4 Beck, K. (2004) *Extreme Programming Explained: Embrace Change*, Upper Saddle River, NJ, Addison Wesley
- 5 Clemens Szyperski (2002) *Component Software: Beyond object-oriented programming*, Addison-Wesley
- 6 John Cheesman & John Daniels (2000) *UML Components: A simple process for specifying component-based software (The component software series)* Addison-Wesley
- 7 Rob Pooley, Perdita Stevens (2006) *Using UML Software Engineering with Objects and Components*, second edition. Addison-Wesley
- 8 Christopher Fox (2006) *Introduction to Software Engineering Design*. Addison Wesley

5 Методика опанування навчальної дисципліни (освітнього компонента)

Розділи та теми	ГОДИНИ			
	Всього	в тому числі		
		Лекції	Лабораторні	Самостійна робота
Розділ1. Варіанти архітектур програмної системи. Тема 1.1. Поняття про архітектуру програмного забезпечення. Тема 1.2. Архітектурні зразки, еталонні моделі та варіанти архітектур. Тема 1.3. Архітектурні споруди та уявлення. Тема 1.4. Зв'язки між структурами. Тема 1.5. Архітектура на основі портів. Тема 1.6. Архітектура незалежних компонентів. Тема 1.7. Сервісно-орієнтовані архітектури.	28	4	8	16
Розділ2. Проектування архітектури програмного забезпечення. Методика проектування. Модульність. Тема 2.1. Модулі, модульний підхід до інтерфейсу. Тема 2.2. Обґрунтування модульності. Тема 2.3. Внутрішня характеристика модуля - зв'язність. Тема 2.4. З'єднання модулів є зовнішньою характеристикою модуля. Тема 2.5. Багаторівнева архітектура Тема 2.6. Логічна структура багаторівневої архітектури. Тема 2.7. Шаблони проектування багатошарової архітектури.	32	6	10	16
Розділ3. Архітектура веб-додатків. Тема3.1. Компоненти веб-додатку. Тема 3.2. Архітектура багаторівневого веб-додатку. Тема 3.3. Архітектура змішаного веб-додатку. Тема 3.4. Архітектура багаторівневого веб-додатку на основі технології CORBA. Тема 3.5. Архітектура веб-додатків з інтерфейсами OLE DB, ADO та ODBC.	30	8	6	16
Розділ4. Патерни проектування. Тема 4.1. Поняття візерунка. Тема 4.2. Види візерунків. Тема 4.3. Генерування шаблонів. Тема 4.4. Структурні закономірності. Тема 4.5. Патерни поведінки. Тема 4.6. Поняття про розетку. Тема 4.7. Функції розеток .	26	6	6	14
Розділ5. Прийоми об'єктно-орієнтованого проектування Тема 5.1. Основні принципи об'єктно-орієнтованої архітектури (ООА). Тема 5.2. Переваги ООА.	34	12	6	16

Тема 5.3. Поняття та елементи ООА.				
Тема 5.4. Поняття технології СОМ.				
Тема 5.5. Проблеми, які вирішуються за допомогою технології СОМ.				
Усього годин у семестрі	150	36	36	78

Лабораторні роботи :

Метою проведення лабораторних занять є закріплення студентами теоретичних знань та набуття необхідних практичних навичок роботи з сучасними технологіями розробки програмного забезпечення .

- Лабораторна робота №1: Розробка . веб-сервіс asmx . Тестування послуг;
- Лабораторна робота №2: Розробка клієнта (веб-серверної програми) для . веб-сервіс asmx . тестування програми;
- Лабораторна робота №3: Реалізація сервісу WCF за допомогою Visual Studio;
- Лабораторна робота №4: Реалізація сервісу WCF як сервісу REST з використанням стандартних прив'язок. Розробка веб- та мобільних сервісів клієнтів ;
- Лабораторна робота № 5: Розробка веб-сервісу Web API;
- Лабораторна робота №6 : Розробка та використання власного сервісу для реалізації залежностей у додатку ASP.NETCore .

6 Самоостійна робота студентів

- підготовка до лекцій шляхом вивчення попереднього лекційного матеріалу;
- підготовка до лабораторної роботи з вивченням теорії лабораторної роботи з усною відповіддю на подані питання розділу;
- оформлення результатів лабораторної роботи у вигляді протоколу.

Політика та контроль

7 Політика навчальної дисципліни (освітнього компонента)

Під час занять з навчальної дисципліни студенти повинні дотримуватись певних дисциплінарних правил:

- не допускаються сторонні розмови та інший шум, що заважає заняттям;
- без дозволу вчителя не можна користуватися мобільними телефонами та іншими технічними засобами.

Лабораторні роботи здаються персонально з попередньою перевіркою теоретичних знань, необхідних для виконання лабораторних робіт. Перевірка практичних результатів включає перевірку коду та виконання тестових завдань.

8 Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: [опитування за темою уроку](#)

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог навчального плану.

Семестровий контроль: [екзамен](#)

Умови допуску до семестрового контролю: [зарахування всіх лабораторних робіт](#)

Система рейтингових балів і критерії оцінювання

Рейтинг студента з дисципліни складається з балів, які він отримує за:

1. виконання та захист 6 лабораторних робіт;
2. виконання 2 модульних контрольних робіт (МКР).

Лабораторні роботи :

«відмінно», повна відповідь на питання під час захисту (не менше 90% необхідної інформації) та правильно оформлений протокол лабораторної роботи – 5 балів;

«добре», достатньо повна відповідь на питання під час захисту (не менше 75% необхідної інформації) та правильно оформлений протокол лабораторної роботи – 4 бали;

«задовільно», неповна відповідь на питання під час захисту (не менше 60% необхідної інформації), незначні помилки та правильно складений протокол лабораторної роботи – 3 бали;

«незадовільно», незадовільна відповідь та/або неналежним чином оформлений протокол лабораторної роботи – 0 балів.

Модульні контрольні роботи :

«Відмінно», повна відповідь (не менше 90% необхідної інформації) – 10 балів;

«Добре», повна відповідь (не менше 75% необхідної інформації), або повна відповідь з незначними помилками – 8 балів;

«Задовільно», неповна відповідь (але не менше 60% необхідної інформації) та незначні помилки – 6 балів;

«Незадовільно», незадовільна відповідь (неправильне розв'язання задачі), потребує обов'язкового переписування в кінці семестру – 0 балів.

Максимальна сума балів контрольних заходів протягом семестру становить:

$$R=6 \cdot R_{\text{lab}} + 2 \cdot R_{\text{mcw}} = 6 \cdot 5 + 2 \cdot 10 = 50.$$

іспит:

Допуск до іспиту здійснюється за умови здачі всіх лабораторних робіт, написання обох модульних контрольних робіт та стартового рейтингу не менше 17 балів.

На іспиті студенти виконують письмову контрольну роботу. Кожен білет містить два теоретичних запитання (завдання). Кожне питання (завдання) оцінюється в 25 балів.

Таблиця 1 — Відповідність рейтингових балів оцінкам за університетською шкалою

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
нижче 60	Незадовільно
Не виконані умови допуску	Не допущено

Робочу програму навчальної дисципліни (силабус):

Розробив професор кафедри Інформаційних систем і технологій Богдан Корнієнко

Ухвалено: кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено: методичною комісією факультету (протокол № 10 від 09.06.2022)