

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ  
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

**«ЗАТВЕРДЖУЮ»**

Заступник декана  
з навчальної роботи

\_\_\_\_\_ Кашпур О.Ф.

«\_\_» \_\_\_\_\_ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
АКТУАЛЬНІ ПРОБЛЕМИ ОБ'ЄКТНО-  
ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ  
для студентів**

галузь знань	<b>12 Інформаційні технології</b>
спеціальність	<b>121 Інженерія програмного забезпечення</b>
освітній рівень	<b>магістр</b>
освітня програма	<b>Програмне забезпечення систем</b>
вид дисципліни	<b>обов'язкова</b>

Форма навчання	<b>денна</b>
Навчальний рік	<b>2021/2022</b>
Семестр	<b>3</b>
Кількість кредитів ECTS	<b>3</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>іспит</b>

Викладачі: **к. ф.-м. н., асистент Жереб К.А.** (лекції, лабораторні заняття).

Пролонговано: на 20\_\_/20\_\_ н. р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.

на 20\_\_/20\_\_ н. р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.

Розробник: Жереб Костянтин Анатолійович, к. ф.-м. н., асистент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

\_\_\_\_\_ О.І. Провотар

Протокол № \_\_ від «\_\_» \_\_\_\_\_ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «\_\_» \_\_\_\_\_ 2019 року №\_\_

Голова науково-методичної комісії \_\_\_\_\_ Л.Л. Омельчук

«\_\_» \_\_\_\_\_ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «\_\_» \_\_\_\_\_ 2019 року №\_\_

Голова вченої ради факультету \_\_\_\_\_ А.В. Анісімов

**1. Мета дисципліни** – оволодіння сучасними методами проектування та розробки об'єктно-орієнтованих застосунків, підготовка до практичного використання технологій та платформ об'єктно-орієнтованого програмування.

**2. Попередні вимоги до опанування або вибору навчальної дисципліни:**

1. **Знати:** базові відомості про розробку програмного забезпечення з використанням сучасних мов програмування, принципи проектування об'єктно-орієнтованих застосунків.
2. **Вміти:** використовувати інструментальні засоби побудови, запуску та відлагодження програм, порівнювати технології реалізації та вибирати оптимальну технологію для вирішення конкретної задачі, шукати додаткову інформацію про об'єктно-орієнтовані технології.
3. **Володіти елементарними навичками:** програмування мовами Python, Java, C#, JavaScript.

**3. Анотація навчальної дисципліни.** Навчальна дисципліна «Актуальні проблеми об'єктно-орієнтованого програмування» є складовою освітньо-наукової програми підготовки фахівців за другим (магістерським) рівнем вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення в рамках освітньо-наукової програми «Програмне забезпечення систем».

Дана дисципліна є нормативною навчальною дисципліною в рамках освітньої програми «Програмне забезпечення систем». Викладається у **3 семестрі в обсязі – 90 год. (3 кредити ECTS)** зокрема: лекції – 14 год., лабораторні заняття – 14 год., консультації – 2 год., самостійна робота – 60 год. У курсі передбачено 2 змістовні частини та 2 контрольні роботи. Завершується дисципліна – **іспитом**.

В результаті вивчення навчальної дисципліни студент повинен:

**знати** методи та підходи до розробки програмного забезпечення з використанням сучасних мов програмування; можливості сучасних об'єктно-орієнтованих технологій, бібліотек та сервісів; принципи проектування об'єктно-орієнтованих застосунків.

**вміти** проектувати та розробляти об'єктно-орієнтовані застосунки з використанням сучасних технологій; вибирати найбільш вдалі технології для реалізації проекту; забезпечувати продуктивність застосунків; використовувати інструментальні засоби розробки.

Для допуску до дисципліни «Актуальні проблеми об'єктно-орієнтованого програмування» освітньо-професійної програми «Програмне забезпечення систем» студент повинен опанувати компетентності та результати навчання, які надають дисципліни «Кластерні розрахунки» та «Розробка та використання інформаційних мереж» програми «Програмне забезпечення систем».

Дисципліна «Актуальні проблеми об'єктно-орієнтованого програмування» є базовою для засвоєння дисциплін вибіркової компоненти програми «Програмне забезпечення систем» спеціалізації «Інтелектуальні програмні системи», зокрема «Системи на основі знань», «Мультиагентні системи» та «Пошуково-аналітичні системи», а також спеціалізації «Програмне забезпечення систем», зокрема «Проектування мультиагентних систем».

**4. Завдання (навчальні цілі).** Основними завданнями дисципліни «Актуальні проблеми об'єктно-орієнтованого програмування» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області об'єктно-орієнтованого програмування відповідно до освітньої кваліфікації магістр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність мотивувати людей та рухатися до спільної мети, працювати в команді співробітників (ЗК04).
- Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності) (ЗК05).
- Здатність аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення (СК01).
- Здатність ідентифікувати, класифікувати та описувати проектні завдання, знаходити раціональні методи й підходи до їх розв'язання (СК02).
- Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК03).
- Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (СК04).
- Здатність оцінювати ступінь обґрунтованості застосування специфікацій, стандартів, правил і рекомендацій в професійній галузі та дотримуватися їх при реалізації процесів життєвого циклу програмного забезпечення (СК05).
- Здатність систематизувати професійні знання щодо створення і супроводження програмного забезпечення (СК07).
- Здатність розробляти і координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення (СК08).

## 5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати методи та підходи до розробки програмного забезпечення з використанням сучасних мов програмування	Лекції, лабораторні, заняття.	Тест, 60% правильних відповідей, іспит.	10%
РН1.2	Знати можливості сучасних об'єктно-орієнтованих технологій, бібліотек та сервісів	Лекції, лабораторні, заняття.	Тест, 60% правильних відповідей, іспит.	15%
РН1.3	Знати принципи проектування об'єктно-орієнтованих застосунків	Лекції, лабораторні, заняття.	Тест, 60% правильних відповідей, іспит.	10%
РН2.1	Вміти проектувати та розробляти об'єктно-орієнтовані застосунки з	Лабораторні заняття, самостійна	Захист лабораторної роботи, іспит.	15%

	використанням сучасних технологій	робота.		
РН2.2	Вміти вибирати найбільш вдалі технології для реалізації проекту	Лабораторні заняття, самостійна робота.	Захист лабораторної роботи.	10%
РН2.3	Вміти забезпечувати продуктивність застосунків	Лабораторні заняття, самостійна робота.	Захист лабораторної роботи, іспит	10%
РН2.4	Вміти використовувати інструментальні засоби розробки	Лабораторні заняття, самостійна робота.	Захист лабораторної роботи.	10%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, писати технічну документацію програм	Лабораторні заняття.	Захист лабораторної роботи, іспит.	10%
РН4.1	Організувати самостійну роботу для вивчення нових технологій	Самостійна робота.	Захист лабораторної роботи.	5%
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Лабораторна робота.	Захист лабораторної роботи.	5%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН1.1	РН1.2	РН1.3	РН2.1	РН2.2	РН2.3	РН2.4	РН3.1	РН4.1	РН4.2
	Програмні результати навчання									
<b>ПРН01.</b> Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення.		+				+			+	+
<b>ПРН02.</b> Обґрунтовувати вибір методів формування вимог до програмної системи, розробляти, аналізувати та систематизувати вимоги.		+				+	+			
<b>ПРН03.</b> Знати і застосовувати базові концепції і методології моделювання інформаційних процесів.	+			+			+			+
<b>ПРН06.</b> Аналізувати, оцінювати і обирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії.	+	+		+	+	+			+	+
<b>ПРН07.</b> Обґрунтовано обирати парадигми і мови			+					+	+	

програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення.										
<b>ПРН08.</b> Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення.							+			
<b>ПРН09.</b> Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення.					+				+	+
<b>ПРН12.</b> Формулювати, експериментально підтверджувати, обґрунтовувати і застосовувати на практиці в процесі розробки програмного забезпечення конкурентоспроможні ідеї, методи, технології вирішення професійних, науково-технічних завдань в умовах невизначеності.			+					+		+
<b>ПРН14.</b> Пояснити, аналізувати, цілеспрямовано шукати і обирати необхідні для вирішення фахових наукових і прикладних задач інформаційно-довідкові та науково-технічні ресурси і джерела знань з урахуванням сучасних досягнень науки і техніки.			+		+		+			+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### Семестрове оцінювання:

1. Контрольна робота (тест) 1: РН 1.1, РН 1.2 – **10 балів/6 балів.**
2. Контрольна робота (тест) 2: РН1.2, РН 1.3 – **10 балів/6 балів.**
3. Лабораторна робота 1 (проект): РН2.1, РН2.2, РН3.1, РН4.1, РН4.2 – **10 балів/6 балів.**
4. Лабораторна робота 2 (проект): РН2.1, РН2.2, РН3.1, РН4.1, РН4.2 – **10 балів/6 балів.**
5. Лабораторна робота 3 (проект): РН2.2, РН2.3, РН3.1, РН4.1, РН4.2 – **10 балів/6 балів.**
6. Лабораторна робота 4 (проект): РН2.2, РН2.3, РН2.4, РН3.1, РН4.1, РН4.2 – **10 балів/6 балів.**

#### Підсумкове оцінювання (у формі іспиту):

- Максимальна кількість балів які можуть бути отримані студентом: 40 балів.
- Результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН2.3, РН3.1.
- Форма проведення і види завдань: письмова робота.
- Види завдань: 4 письмових завдання.

#### Критерії оцінювання на іспиті.

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1	Письмове запитання з використання нереляційних та постреляційних сховищ даних	25%	25%

Завдання 2	Письмове запитання з технологій роботи з великими обсягами даних (Big Data)	25%	25%
Завдання 3	Письмове запитання з використання технологій роботи з текстом	25%	25%
Завдання 4	Письмове запитання з методів забезпечення продуктивності застосунків	25%	25%
			<b>100%</b>

### Запитання для підготовки до іспиту.

1. Сучасні об'єктно-орієнтовані технології.
2. Модульне тестування.
3. Підтримка модульного тестування в засобах розробки.
4. Засоби підрахунку покриття коду.
5. Mock object frameworks.
6. Принципи створення ефективних модульних тестів.
7. Принципи проектування об'єктно-орієнтованих систем.
8. Мова UML
9. Патерни проектування.
10. Інтеграція засобів UML в процес розробки.
11. Порівняння імперативних та декларативних засобів опису графічного інтерфейсу.
12. Основні особливості мови XAML.
13. Елементи керування. Події.
14. Розташування елементів (layout).
15. Зв'язування даних. Залежні властивості.
16. Ефекти і перетворення.
17. Анімація.
18. Робота з аудіо і відео.
19. Відкриті стандарти векторної графіки. Технологія SVG.
20. Технології SMIL, XUL, XForms.
21. Порівняння можливостей HTML5 та SVG.
22. Невідповідність об'єктно-орієнтованої і реляційної моделі («impedance mismatch»).
23. Засоби доступу до даних. Послідовний та довільний доступ. Доступ до реляційних даних та XML.
24. Системи Object Relational Mapping.
25. Microsoft Entity Framework.
26. NoSql бази даних. Порівняння з реляційними системами.
27. Технологія LINQ. Доступ до даних різних типів (реляційні, XML, об'єкти)
28. Технології XPath, XQuery, XSLT.
29. Функції обробки текстових даних.
30. Регулярні вирази.
31. Синтаксичні аналізатори. Система ANTLR.
32. Системи повнотекстового пошуку.
33. Системи Natural Language Processing (NLP).
34. Програмування графічних прискорювачів. Платформа CUDA.
35. Програмування обчислювальних ядер (kernels).
36. Виклик обчислювальних ядер. C for CUDA, driver API. Виклик з високорівневих мов (C#, Java).
37. Використання розділюваної пам'яті.
38. Системи та бібліотеки для програмування графічних прискорювачів.

Студенти не допускаються до екзамену, якщо вони під час семестру набрали менше ніж 36 балів та/або не виконали хоча б 70% передбачених планом лабораторних робіт.

## 7.2 Організація оцінювання.

### Терміни проведення форм оцінювання:

1. Контрольна робота (тест): до 7 тижня семестру.
2. Контрольна робота (тест): до 13 тижня семестру.
3. Лабораторна робота 1 (проект): до 4 тижня семестру.
4. Лабораторна робота 2 (проект): до 8 тижня семестру.
5. Лабораторна робота 3 (проект): до 12 тижня семестру.
6. Лабораторна робота 4 (проект): до 13 тижня семестру.

Студенти мають право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студентів з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до «Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу» від 1 жовтня 2010 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студенти мають право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

## 7.3 Шкала відповідності оцінок.

<b>Відмінно / Excellent</b>	90-100
<b>Добре / Good</b>	75-89
<b>Задовільно / Satisfactory</b>	60-74
<b>Незадовільно / Fail</b>	0-59

## 8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять.

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабораторні заняття	Самостійна робота
<b>Частина 1. Процеси розробки об'єктно-орієнтованих застосунків.</b>				
1	<b>Тема 1</b> Сучасний стан об'єктно-орієнтованого програмування.	2	2	6
2	<b>Тема 2.</b> Модульне тестування (Unit testing).	2	2	8
3	<b>Тема 3.</b> Проектування та архітектура об'єктно-орієнтованих програм. Архітектурні патерни.	2	2	8
4	<b>Тема 4.</b> Технології Big Data.	2	2	8
Контрольна робота 1				2



Всього по частині 1		8	8	32
<b>Частина 2. Об'єктно-орієнтовані технології.</b>				
5	<b>Тема 5.</b> Робота з реляційними та нереляційними даними.	2	2	9
6	<b>Тема 6.</b> Робота з текстовими даними.	2	2	8
7	<b>Тема 7.</b> Розробка паралельних програм для графічних прискорювачів.	2	2	9
Підсумкова контрольна робота				2
Всього по частині 2		6	6	28
Консультації		2		
<b>ВСЬОГО</b>		<b>14</b>	<b>14</b>	<b>60</b>

Загальний обсяг – **90** год., в тому числі:

Лекції – **14** год.

Лабораторні заняття – **14** год.

Самостійна робота – **60** год.

Консультації – **2** год.

#### **Теми, винесені на самостійне вивчення.**

Технології та бібліотеки машинного навчання.

#### **Умови лабораторних робіт:**

- **Лабораторна робота 1:** Розробка застосунку з використанням NoSql баз даних.
- **Лабораторна робота 2:** Розробка застосунку з використанням технологій Big Data.
- **Лабораторна робота 3:** Розробка застосунку для роботи з великими обсягами текстових даних.
- **Лабораторна робота 4:** Програмування відеографічних прискорювачів.
- Деталізовані умови лабораторних робіт розміщено за посиланням:  
<https://drive.google.com/drive/folders/18DF7GsGFdbqk8vmtFZHsgVBiRfODmGhU>

#### **9. Рекомендовані джерела.**

##### **Основні:**

1. Буч Г. Объектно-ориентированное проектирование с примерами применения. – М.: Конкорд, 1992.
2. К. Нейгел, Б. Иввен, Дж. Глинн, К. Уотсон, М. Скиннер. С# 4.0 и платформа .NET 4 для профессионалов. – М.: Диалектика-Вильямс, 2010. 1440 стр.
3. Дж. Месарош. Шаблоны тестирования xUnit: рефакторинг кода тестов. – М.: Диалектика-Вильямс, 2008. 832 стр.
4. М. Кон. Scrum: гибкая разработка ПО. – М.: Диалектика-Вильямс, 2011. 576 стр.
5. М. Фаулер. UML. Основы. 3-е издание. – М.: Символ, 2005. 192 стр.

6. Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влссидес Приемы объектно-ориентированного проектирования. Паттерны проектирования – СПб.: «Питер», 2007. – 366 стр.
7. М. Мак-Дональд. Silverlight 3 с примерами на С# для профессионалов. – М.: Диалектика-Вильямс, 2010. 656 стр.
8. Бенкен Е.С. PHP, MySQL, XML. Программирование для Интернета. – СПб: БХВ-Петербург, 2007 г. 336 стр.
9. П. Лабберс, Б. Олберс, Ф. Салим. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений. – М.: Диалектика-Вильямс, 2011. 304 стр.
10. Дж. С. Ратц-мл. LINQ: язык интегрированных запросов в С# 2010 для профессионалов. – М.: Диалектика-Вильямс, 2011. 656 стр.

#### **Додаткові:**

1. К. Смит. Программирование на F#. – М.: Символ, 2011. 448 стр.
2. Open Source Development with CVS <http://cvsbook.red-bean.com/>
3. Version Control with Subversion <http://svnbook.red-bean.com/>
4. Pro Git <http://progit.org/book/>
5. Mercurial: The Definitive Guide <http://hgbook.red-bean.com/>
6. ANTLR Parser Generator <http://www.antlr.org/>
7. Intel Parallel Studio <http://software.intel.com/en-us/articles/intel-parallel-studio-home/>
8. NVidia CUDA [http://www.nvidia.com/object/cuda\\_home\\_new.html](http://www.nvidia.com/object/cuda_home_new.html)
9. IronPython <http://ironpython.net/>
10. Wikipedia, the free encyclopedia <http://en.wikipedia.org>, <http://uk.wikipedia.org>, <http://ru.wikipedia.org>
11. ASP .NET <http://msdn.microsoft.com/asp.net/>
12. Best Practices for Speeding Up Your Web Site <http://developer.yahoo.com/performance/rules.html>

#### **10. Додаткові ресурси.**

- <https://drive.google.com/drive/folders/18DF7GsGFdbqk8vmtFZHsgVBiRfODmGhU>