

# КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики  
Кафедра математичної інформатики

«ЗАТВЕРДЖУЮ»  
Заступник декана  
з навчальної роботи  
  
Олена КАШПУР  
« 12 » \_\_\_\_\_ 2021 року

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ Сучасні технології розробки програм

для студентів

галузь знань **12 «Інформаційні технології»**  
спеціальність **122 «Комп'ютерні науки»**  
освітній рівень **магістр**  
освітня програма **«Інформатика»**  
вид дисципліни **вибіркова**

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	3
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: асистент **Богдан БОБИЛЬ**

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

КИЇВ – 2021

Розробник: **Богдан БОБИЛЬ**, асистент кафедри математичної інформатики

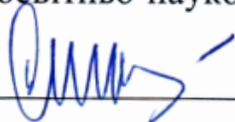
ЗАТВЕРДЖЕНО

Завідувач кафедри  
математичної інформатики

  
\_\_\_\_\_ Василь ТЕРЕЩЕНКО

Протокол № 6 від «06» лютого 2021 року

Схвалено гарантом освітньо-наукової програми «Інформатика»

  
\_\_\_\_\_ Степан ШКІЛЬНЯК

«11» лютого 2021 року

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

Протокол від «11» лютого 2021 року № 7

Голова науково-методичної комісії \_\_\_\_\_  Людмила ОМЕЛЬЧУК

«11» лютого 2021 року

**1. Мета дисципліни** «Сучасні технології розробки програм» – ознайомлення студентів із сучасними технологіями розробки програм, ознайомлення із життєвим циклом розробки, тестування та підтримки програмних систем; підготовка студентів до ефективного використання сучасних методів розробки програм: інструменти для безперервної інтеграції та впровадження, контейнеризації, хмарні обчислення.

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

1. *Знати:* базові дисципліни «Програмування», «Інформаційні технології», «Розподілене та паралельне програмування».

2. *Вміти:* розробляти, аналізувати та застосовувати алгоритми та програмне забезпечення для розв'язання завдань та прикладних задач, використовуючи сучасні методи розробки програм.

## **3. Анотація навчальної дисципліни:**

Навчальна дисципліна «Сучасні технології розробки програм» є складовою освітньо-наукової програми підготовки фахівців за освітньо-кваліфікаційним рівнем «магістр» галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки», освітньої програми «Інформатика».

Дана дисципліна викладається у 3 семестрі (2 курс) в обсязі 120 год. (4 кредити ECTS); зокрема: *лекції – 34 год., консультації – 2 год., самостійна робота – 84 год.*

**Місце дисципліни.** Навчальна дисципліна «Сучасні технології розробки програм» є складовою циклу професійної підготовки фахівців освітнього рівня "магістр" за вибіркоким блоком «Інтелектуальні інформаційні технології» освітньої програми «Інформатика».

**4. Завдання (навчальні цілі):** *знати* інструменти організації розробки, тестування та впровадження програмних систем; *вміти* застосовувати сучасні технології розробки програм для розв'язання широкого класу задач науки та техніки. Зокрема, розвивати:

**СК19.1.** Здатність обґрунтовано обирати та застосовувати інструментарій з розробки та супроводження програмного забезпечення.

## 5. Результати навчання за дисципліною:

Результат навчання (РН) (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			
РН 1.1	Знати постановку основних задач, розробки, тестування та впровадження	Лекція	Контрольна робота, поточне оцінювання іспит	35%
РН 1.2	Знати основні інструменти розробки, тестування та впровадження	Лекція		
РН 1.3	Знати основні поняття та засоби хмарних обчислень	Лекція	Контрольна робота, поточне оцінювання, іспит	35%
РН 2.1	Вміти проводити аналіз задачі організації розробки та обирати найкращі методи для її розв'язання	Лекція, самостійна робота	Захист завдань самостійної роботи, іспит	25%
РН 2.2	Вміти застосовувати методи для розв'язання прикладних задач	Лекція, самостійна робота	Захист завдань самостійної роботи роботи	5%

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

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 2.2
<b>Програмні результати навчання</b>					
<i>(з опису освітньої програми)</i>					
<b>ПРН 17.1.</b> Мотивовано обирати технології програмування для розв'язання завдань створення і супроводження програмного забезпечення.	+			+	
<b>ПРН 18.1.</b> Знати та вміти застосовувати сучасні програмно-апаратні та обчислювальні засоби, технології та програмні рішення для ефективного аналізу конкретних виробничих задач для розробки програмного забезпечення.		+	+	+	+

## 7. Схема формування оцінки.

### 7.1. Форми оцінювання студентів:

– семестрове оцінювання (максимальна кількість балів):

1. Контрольна робота 1: РН1.1, РН1.2 – 16 балів;
2. Контрольна робота 2: РН1.3 – 16 балів;
3. Виконання завдань, винесених на самостійну роботу: РН2.1, РН2.2 – 20 балів;
4. Активна робота на лекції, усні відповіді: РН1.1, РН1.2, РН1.3 – 8 балів.

- підсумкове оцінювання: іспит.

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН 2.1;
- форма проведення: письмова.
- види завдань: тестове завдання, 3 письмові завдання.

## 7.2. Організація оцінювання:

Обов'язковим є виконання завдань, винесених на самостійну роботу, та контрольних робіт.

Переписування чи перескладання контрольних робіт не практикується.

Дозволяється здача окремих завдань тем у проміжках між написанням контрольних робіт (наприклад, перша тема здається до написання контрольної роботи у будь-який зручний для викладача та студента час).

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 6 тижня семестру.
2. Контрольна робота 2: до 13 тижня семестру.

У випадку відсутності студента з поважних причин відпрацювання та перездачі МКР здійснюються у відповідності до «Положення про організацію освітнього процесу» від 07.05.2018 року.

Студент допускається до іспиту, якщо він під час семестру набрав не менше 30 балів, у тому числі набрав не менше 18 балів за контрольні роботи

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів

## 8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ЛАБОРАТОРНИХ ЗАНЯТЬ

№ лекції	Назва лекції	Кількість годин		
		Лекції	Консультації	Самостійна робота
<b>Частина 1. Життєвий цикл розробки програмних систем</b>				
1	<b>Тема 1.</b> Вступ до розробки програмних систем <i>Самостійна робота:</i> Провести аналіз програми і змісту курсу та рекомендованої літератури.	1		6
2	<b>Тема 2.</b> Методології організації розробки програмних систем <i>Самостійна робота:</i> Розуміти та обирати ефективну методології розробки.	3		8
3	<b>Тема 3.</b> Методи контейнеризації програмних систем. <i>Самостійна робота:</i> Розуміти та вміти користуватися інструментами контейнеризації, таким як Docker.	4		8

4	<b>Тема 4.</b> Інструменти для безперервної інтеграції та впровадження. <i>Самостійна робота:</i> Розуміти та вміти користуватися інструментами для інтеграції та впровадження, такими як Jenkins, CircleCI.	3		8
<i>Контрольна робота 1</i>		1		
Всього за частиною 1		12		30
<b>Частина 2. Хмарні обчислення</b>				
5	<b>Тема 5.</b> Платформи для хмарних обчислень. <i>Самостійна робота:</i> Обирати ефективну платформу для хмарних обчислень.	2		6
6	<b>Тема 6.</b> Організація обчислень в хмарі. <i>Самостійна робота:</i> Опрацювати методи організації обчислень в хмарі.	4		8
7	<b>Тема 7.</b> Інструменти для організації даних в хмарі. <i>Самостійна робота:</i> Опрацювати методи збереження даних в хмарі.	4		10
8	<b>Тема 8.</b> Побудова програмних систем в хмарі. <i>Самостійна робота:</i> Опрацювати методи побудови.	4		10
Всього за частиною 2		14		34
<b>Частина 3. Огляд сучасних архітектур нейронних мереж</b>				
9	<b>Тема 9.</b> Тестування програмного забезпечення. <i>Самостійна робота:</i> Опрацювати основні методи тестування.	4		10
10	<b>Тема 10.</b> Організація автоматичного тестування. <i>Самостійна робота:</i> Опрацювати основні методи тестування	3		10
<i>Модульна контрольна робота 2</i>		1		
Всього за частиною 3		8		20
Консультації			2	
ВСЬОГО		34	2	84

Загальний обсяг 120 годин, в тому числі:

Лекцій – 34 годин

Консультацій – 2 години

Самостійна робота – 84 годин.

## 9. РЕКОМЕНДОВАНІ ДЖЕРЕЛА

### *Основні:*

1. Thomas Erl, Ricardo Puttini, Zaigham Mahmood. *Cloud Computing: Concepts, Technology & Architecture (The Pearson Service Technology Series from Thomas Erl) 1st Edition*, Pearson, 2013
2. Martin Kleppmann. *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems 1st Edition*, O'Reilly Media, 2017
3. Robert Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition*, Pearson, 2017
4. Mark Richards, Neal Ford. *Fundamentals of Software Architecture: An Engineering Approach 1st Edition*, O'Reilly Media, 2020
5. Sam Newman. *Building Microservices: Designing Fine-Grained Systems 2nd Edition*, O'Reilly Media, 2021
6. Sam Newman. *Monolith to Microservices: Evolutionary Patterns to Transform Your Monolith 1st Edition*, O'Reilly Media, 2019
7. Paul M. Duvall. *Enterprise DevOps on Amazon Web Services. Releasing Software to Production at Any Time with AWS*, Addison-Wesley Professional, 2021
8. Dave Farley. *Continuous Delivery Pipelines: How To Build Better Software Faster*, Independently published, 2021

### *Додаткові:*

1. Stephen Fleming. *Continuous Delivery Handbook: Non Programmer's Guide to DevOps, Microservices and Kubernetes*, CreateSpace Independent Publishing Platform, 2018
2. Brendan Burns. *Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services 1st Edition*, O'Reilly Media, 2018
3. Maarten van Steen. *Distributed Systems 3rd Edition*, CreateSpace Independent Publishing Platform, 2017