

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

 Олена КАШПУР

« 7 » _____ 20 21 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
"ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ"

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)
спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)
освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)
освітня програма **«Інформатика»**
(назва освітньої програми)

вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2023/2024
Семестр	7
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: к.ф.-м.н., доц. Кузенко В.Ф.

Пролонговано: на 20 __/20 __ н.р. _____ (_____) «__» 20 __ р.
(підпис, ПІБ, дата)

на 20 __/20 __ н.р. _____ (_____) «__» 20 __ р.
(підпис, ПІБ, дата)

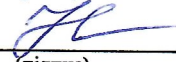
КИЇВ – 2021

Розробник: Кузенко Володимир Федорович, к.ф.-м.н., доцент кафедри «Теорії та технології програмування»



ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»



Микола НІКІТЧЕНКО
(прізвище та ініціали)

Протокол № 10 від «27» квітня 2021 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»



(підпис)

Людмила ОМЕЛЬЧУК «6» травня 2021 рік
(прізвище та ініціали)

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

Протокол від «6» травня 2021 року № 10

Голова науково-методичної комісії _____



Людмила ОМЕЛЬЧУК

ВСТУП

1. Мета дисципліни – засвоєння базових знань та оволодіння навичками щодо використання інформаційних технологій при розробці програмних систем (ПС), зокрема, оволодіння навичками моделювання і проектування клієнт-серверних програмних систем, використання сервісно-орієнтованої архітектури (COA), *REST*-архітектури та деяких альтернативних технологічних рішень (*GraphQL*, *gRPC*), оволодіння навичками *Web*-програмування, програмування під мобільні платформи, використання хмарних обчислень.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:*

- основні поняття та шаблони об'єктно-орієнтованого програмування;
- основні етапи життєвого циклу ПС;

2. *Вміти:* застосовувати на практиці інструментальні засоби моделювання та розробки ПС.

3. *Володіти навичками:* моделювання та проектування ПС, модульного та інтегрованого тестування.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Інформаційні технології” є складовою програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти *галузі знань 12* „Інформаційні технології” зі спеціальності 122 „Комп'ютерні науки”, *освітньо-професійної програми* – „Інформатика”.

Дана дисципліна є обов'язковою навчальною дисципліною за *програмою “Інформатика”*.

Викладається у 7 семестрі 4 курсу в **обсязі – 120 год.**

(**4 кредити ECTS**) зокрема: *лекції – 42 год., лабораторні – 14 год., консультації – 2 год., самостійна робота – 62 год.* Завершується дисципліна – **іспитом у 7 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

знати основні поняття стосовно моделювання, проектування та розробки ПС клієнт-серверної архітектури, **знати** особливості сервісно-орієнтованої архітектури, *REST*-архітектурного стилю, **знати** основні поняття та принципи особливості *Web*-програмування, програмування під мобільні платформи, програмування із використанням хмарних обчислень;

вміти застосовувати на практиці інструментальні середовища, спеціальні бібліотеки та фреймворки при розробки ПС різноманітної архітектури (клієнт-сервер, COA, *REST* тощо) та призначення (*Web*-проекти, мобільні проекти тощо).

Для допуску до дисципліни „Інформаційні технології” освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надають дисципліни “Об'єктно-орієнтоване програмування”, “Інструментальні середовища та технології програмування”, “Системне програмування”. Дисципліна „Інформаційні технології” є базовою для виробничої практики „Інформаційні системи та технології” .

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно до кваліфікації фахівців з інформаційних технологій. Зокрема, розвивати:

- здатність застосовувати знання у практичних ситуаціях;
- здатність працювати в команді;
- здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктноорієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління;
- здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах;
- здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- здатність до аналізу та функціонального моделювання бізнеспроцесів, побудови та практичного застосування функціональних моделей організаційно-економічних і виробничо-технічних систем, методів оцінювання ризиків їх проектування;
- здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації

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 основні поняття, патерни, технології розробки програмних систем із використанням клієнт-серверної архітектури	Лекції, лабораторні роботи	Тест(3%), іспит(10%)	13%
РН 1.2	Знати суть і підгрунтя сервісно-орієнтованої архітектури, технології веб-сервісів <i>SOAP</i> , використання архітектурного стилю <i>REST</i> у програмних системах.	Лекції, лабораторні роботи	Тест(3%), іспит(15%)	18%
РН 1.3	Знати суть і підгрунтя мовних засобів, фреймворків, орієнтованих на використанням веб-технологій, мобільних платформ, хмарних обчислень.	Лекції, лабораторні роботи	Тест(4%), іспит(15%)	19%
РН 2.1	Вміти застосовувати на практиці технології розробки програмних систем, що орієнтовані на клієнт-серверну взаємодію.	Лабораторні роботи, самостійна робота	Захист лабораторних робіт	10%
РН 2.2	Вміти застосовувати у програмних системах <i>SOAP</i> веб-сервіси, <i>REST</i> веб-сервіси.	Лабораторні роботи, самостійна робота	Захист лабораторних робіт	20%
РН 2.3	Отримання навичок використання мовних засобів, фреймворків та інструментальних середовищ підтримки віддаленої взаємодії із залученням веб-технологій, мобільних платформ, хмарних обчислень.	Лабораторні роботи, самостійна робота	Захист лабораторних робіт	20%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	1.1	1.2	1.3	2.1	2.2	2.3
Програмні результати навчання						
<i>(з опису освітньої програми)</i>						
ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.	+	+	+	+	+	+
ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).	+	+	+	+	+	+
ПРН15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.	+			+		

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Тести: РН 1.1., РН 1.2, РН 1.3 — $10(3+3+4)/6$ балів.
2. Лабораторні роботи : РН 2.1 — $10(5+5)/6$ балів.
2. Лабораторні роботи : РН 2.2 — $20(5+5+5+5)/12$ балів.
4. Лабораторні роботи : РН 2.3 — $20(5+5+5+5)/12$ 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;
- форма проведення і види завдань: письмова робота.

Структура екзаменаційної роботи та критерії оцінювання:

2 теоретичних питання (по 12 балів), 1 задача (16 балів)

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

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Тест РН 1.1. : до 5 тижня семестру.
 2. Тест РН 1.2. : до 11 тижня семестру.
 3. Тест РН 1.3. : до 14 тижня семестру.
- Лабораторні роботи — щотижня, починаючи з другого навчального тижня

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

Студент має 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

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

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб. занят.	Самост. робота
1	Тема 1. Уніфікована мова моделювання <i>UML</i> . Спрощена стратегія використання <i>UML</i> -діаграм при моделюванні програмних систем (ПС). Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	4
2	Тема 2. Огляд технологій <i>J2EE</i> . Технологія віддаленої взаємодії <i>Java RMI</i> . Версії <i>JRMP</i> та <i>IIOP</i> . (<i>JRMP</i> – <i>IIOP</i>) портабельність <i>Java RMI</i> -проектів. Сумісність <i>RMI/IIOP</i> - та <i>CORBA</i> -проектів. Технологія <i>Java IDL</i> . Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	4
3	Тема 3. Міжмовна інтеграція у <i>.NET</i> . Віддалена взаємодія об'єктів <i>.NET</i> (<i>.NET-Remoting</i>). Поняття активізації об'єктів, режими активізації. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	4
4	Тема 4. Сервісно-орієнтована архітектура. Універсальність технології веб-сервісів та її підтримка на платформі <i>.NET</i> . Поняття <i>API</i> . Веб-сервіси, <i>WSDL</i> та <i>SOAP API</i> . Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	4
5	Тема 5. Веб-сервіси на платформі <i>Java</i> . Міжплатформна та міжмовна інтеграція на основі веб-сервісів. Приклади інтеграції <i>Java</i> та <i>.NET</i> . Мікросервіси та їх використання. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	4
6	Тема 6. Технологія <i>Microsoft Windows Communication Foundation (WCF)</i> як розвиток технологій <i>.NET-Remoting</i> та веб-сервісами. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	4
7	Тема 7. Архітектурний стиль <i>REST</i> , його особливості та використання. <i>HATEOAS</i> . <i>REST API</i> , документація <i>REST API</i> та проект <i>SWAGGER</i> . Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	5
8	Тема 8. Технологія <i>Google gRPC</i> , її особливості порівняно з <i>REST API</i> . Приклади використання. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	5
9	Тема 9. Технологія <i>Facebook GraphQL</i> , її гасло та особливості. Приклади використання. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	4

10	Тема 10. Веб-проектування. Технології <i>Java Servlet</i> та <i>JSP</i> . Технології <i>ASP Net</i> та їх використання. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	4
11	Тема 11. Архітектура <i>MVC</i> у веб-проектуванні. <i>Action servlet</i> , поняття <i>Action-oriented Frameworks</i> . Огляд <i>MVC</i> веб- фреймворків на платформі <i>Java</i> . Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	4
12	Тема 12. Фреймворк <i>Spring</i> , патерн (принцип) <i>IOC&DI</i> . Проблема валідації даних у веб-проектах. Використання <i>AJAX</i> . Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	6
13	Тема 13. Програмування під мобільні платформи. Основи розробки проектів під <i>OS Android</i> . Приклади. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	2	1	6
14	Тема 14. Хмарні обчислення та технології. Платформа <i>MS Azure</i> та її використання на прикладі. Виконання лабораторної роботи. Самостійна робота: опрацювання лекційного матеріалу.	4	1	4
	ВСЬОГО	42	14	62

Загальний обсяг 120 год., в тому числі:

Лекцій – **42 год.**

Лабораторні заняття - **14 год.**

Консультації – **2 год.**

Самостійна робота - **62 год.**

9. Рекомендовані джерела:

Основні:

1. <https://magistrs2016.github.io/> - презентації та умови завдань лабораторного практикуму.
2. Лаврішчева К. Програмна інженерія, Київ., 2008, 319 с.
3. Unhelkar B., Software Engineering with UML, 2018, 426 p.
4. Немраджани А., Agile Java Development with Spring, Hibernate and Eclipse, 2006, 362 p.
(Хемрадджани А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse. 2008, 352 с.).
5. Naylor L., ASP.NET MVC with Entity Framework and CSS, 2016, 602p.
6. Savas Parastatidis S., Robinson I., REST in Practice: Hypermedia and Systems Architecture, 2010, 418p.
7. Franceschi H. Android App Development, 2017, 673p.
8. Porcello E., Banks A. Learning GraphQL: Declarative Data Fetching for Modern Web Apps, 2018, 181p.
9. Humphries J., Konsumer D., Muto D., Ross R., Sistare C. Practical gRPC, 2018, 234p.

Додаткові:

1. Phillips B., Stewart C., Marsicano K., Android Programming: The Big Nerd Ranch Guide, 2017, 278p.
2. Grosso W., Java RMI: Designing & Building Distributed Applications, 2002, 521p.
3. Sahay R., Pragmatic WCF Paperback, 2015, 329p.
4. Snell J., Tidwell D., Kulchenko P., Programming Web Services With SOAP, 2002, 237p.
5. Kurtz J., Wortman B. ASP.NET Web API 2: Building a REST Service from Start to Finish, 2014, 280p.
6. <http://msdn.microsoft.com>
7. <https://developers.google.com/appengine/>
8. <http://www.hibernate.org/>
9. <http://www.springframework.org/>
10. <http://developer.android.com/index.html>
11. <https://graphql.org/>
12. <https://grpc.io/>