

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ**

**Кафедра теорії та технології програмування**

**«ЗАТВЕРДЖУЮ»**

Заступник декана  
з навчальної роботи

\_\_\_\_\_ Кашпур О.Ф.

« \_\_\_\_ » \_\_\_\_\_ 2018 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
Теорія обчислень та комп'ютерні технології**

для студентів

галузь знань      **12 «Інформаційні технології»**  
*(шифр і назва)*

спеціальність    **122 «Комп'ютерні науки»**  
*(шифр і назва спеціальності)*

освітній рівень    **магістр**  
*(молодший бакалавр, бакалавр, магістр)*

освітня програма **«Інформатика»**  
*(назва освітньої програми)*

вид дисципліни    **обов'язкова**

Форма навчання	<b>денна</b>
Навчальний рік	<b>2018/2019</b>
Семестр	<b>1</b>
Кількість кредитів ECTS	<b>5</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>екзамен</b>

Викладачі: **д.ф.-м.н., проф. Буй Д.Б.** (лекції, лабораторні заняття)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» \_\_ 20\_\_ р.  
*(підпис, ПІБ, дата)*

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» \_\_ 20\_\_ р.  
*(підпис, ПІБ, дата)*

**КИЇВ – 2018**

Розробник: Буй Дмитро Борисович, д.ф.-м.н., професор кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

В.о. Зав. кафедри «Теорії та технології програмування»

\_\_\_\_\_ Панченко Т.В.  
(підпис) (прізвище та ініціали)

Протокол № \_\_\_\_ від «\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

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

Протокол від «\_\_\_\_» \_\_\_\_\_ 20\_\_ року № \_\_\_\_

Голова науково-методичної комісії \_\_\_\_\_ Хусаїнов Д.Я.  
(підпис) (прізвище та ініціали)

Затверджено Вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «\_\_\_\_» \_\_\_\_\_ 20\_\_ року № \_\_\_\_

Голова Вченої ради \_\_\_\_\_ Анісімов А.В.  
(підпис) (прізвище та ініціали)

## ВСТУП

**1. Мета дисципліни** – отримання знань та оволодіння навичками використання інформаційних технологій розробки програмних систем, зокрема, навичками програмної інженерії, застосування моделювання, CASE-технологій, у тому числі при розробці розподілених програмних систем, Web-систем, а також систем із сервісно-орієнтованою архітектурою.

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

1. *Успішне опанування курсів:* "Дискретна математика", "Програмування", "Бази даних", "Операційні системи", "Системне програмування", "Інтернет-технології".

2. *Знання теоретичних основ:* формальних методів розробки програмних систем.

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

Навчальна дисципліна "Теорія обчислень та комп'ютерні технології" є складовою освітньо-наукової програми підготовки фахівців за *освітнім рівнем* «магістр» галузі знань 12 "Інформаційні технології" за *спеціальністю* 122 „Комп'ютерні науки”, програми «Інформатика».

Викладається у 1 семестрі в **обсязі – 150 годин.**

**(5 кредитів ECTS)** зокрема: *лекції – 26 год., лабораторні – 66 год., консультації – 2 год., самостійна робота – 56 год.* У курсі передбачено 3 частини та 2 контрольні роботи. Завершується дисципліна **екзаменом в 1 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

**знати:** основні поняття об'єктно-орієнтованої (ОО) парадигми, принципи ОО проектування та програмної інженерії, основні поняття мови UML, стратегію використання UML-діаграм при моделюванні програмних систем, основні архітектурні патерни, основи .NET- та Java-технологій, поняття сервісно-орієнтованої архітектури, парадигму декларативного програмування та її застосування у технологіях .NET та Java , поняття сервісно-орієнтованої архітектури, поняття фреймворку.

**вміти:** використовувати об'єктно-орієнтований стиль програмування при проектуванні програмних систем, застосовувати моделювання програмних систем на основі UML-діаграм, розробляти розподілені системи з використанням різноманітних технологій (.NET-Remoting, Java-RMI), застосовувати сервісно-орієнтовану архітектуру, проектувати і розробляти Web-системи з використанням сучасних фреймворків, проектувати і розробляти мобільні проекти, застосовувати хмарні обчислення та технології.

**Місце дисципліни.** Навчальна дисципліна "Програмна інженерія" є складовою циклу професійної підготовки фахівців освітнього рівня "магістр" освітньої програми «Інформатика».

### **4. Завдання (навчальні цілі):**

набуття знань, умінь та навичок (компетенцій) на рівні новітніх досягнень у програмуванні, відповідно до освітньої кваліфікації «Магістр з комп'ютерних наук за спеціалізацією теорія та технологія програмування». Зокрема, розвивати:

- здатність розробляти й управляти проектами;
- здатність застосовувати квантові операції, виміри, алгоритми для розв'язання задач, пов'язаних з особливостями реалізації квантових обчислень в різних фізичних системах, зокрема в квантовій криптографії;

- здатність аналізувати сучасні світові тенденції розвитку комп'ютерних наук та уявляти перспективи розвитку інформаційних технологій, моделювати процеси розвитку і трансформації інформаційно- комунікаційних технологій в практичній професійній роботі;
- здатність проводити дослідження функціональної та економічної ефективності та надійності інформаційних систем;
- здатність проектування динамічних веб-додатків як інформаційної системи із застосуванням об'єктно-орієнтованих технологій програмування, зокрема сучасних програмних засобів підтримки взаємодії клієнта та сервера із застосуванням розподілених систем керування базами даних, супроводження та оптимізація веб-сторінок.

### 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			
PH1.1	<i>Знати основи інженерії програмних систем. Хмарні обчислення та технології. Програмування під мобільні платформи</i>	<i>Лекція, лабораторне заняття</i>	<i>Тест, 60% правильних відповідей, Екзамен</i>	25%
PH1.2	<i>Знати Web-проекткування на платформах Java та .NET. .NET та Java-платформи.</i>	<i>Лекція, лабораторне заняття, доповідь</i>	<i>Тест, 60% правильних відповідей, Екзамен</i>	25%
PH1.3	<i>Знати сервісно-орієнтована архітектуру та веб-служби.</i>	<i>Лекція, лабораторне заняття, доповідь</i>	<i>Тест, 60% правильних відповідей, Екзамен</i>	20%
PH2.1	<i>Вміти створювати UML діаграми при проектуванні програмних систем. Патерни проектування.</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Доповідь</i>	15%
PH3.1	<i>Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти</i>	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи</i>	10%
PH4.1	<i>Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість</i>	<i>Лабораторна робота, самостійна робота</i>	<i>Захист лабораторної роботи</i>	5%

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

Результати навчання дисципліни	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 3.1	PH 4.1
<b>Програмні результати навчання</b>						
<i>(з опису освітньої програми)</i>						
<b>ПРН4.</b> Аналізувати великі дані та моделювати високорівневі абстракції у великих наборах даних різної природи, проектувати сховища великих даних, для видобутку даних і знань, візуалізувати великі дані, будувати і оцінювати регресивні моделі, що генеруються на основі великих даних	+	+				
<b>ПРН5.</b> Вирішувати складні проблеми, що вимагають	+		+			

систем з великою обчислювальною потужністю для забезпечення масштабованості паралельних алгоритмів і програм.					
<b>ПРН6.</b> Використовувати розподілені високопродуктивні обчислювальні технології для забезпечення ефективного вибору та використання консолідованих ресурсів і послуг			+		
<b>ПРН8.</b> Аналізувати особливості використання сучасних квантових технологій для забезпечення вирішення проблем, зокрема конфіденційного зв'язку, квантової криптографії, здійснювати дослідження теоретичних та експериментальних аспектів квантової інформатики		+		+	+

## 7. Схема формування оцінки.

### 7.1 Форми оцінювання студентів:

#### - семестрове оцінювання:

1. Контрольна робота (тест) 1: РН 1.1., РН 1.2 — 15 балів/9 балів.
2. Контрольна робота (тест) 2: РН1.2, РН1.3 - 15 балів/9 балів.
3. Підготовка доповіді (за темами лекційних занять та самостійної роботи): РН 2.1 – 15 балів/9 балів.
4. Лабораторна робота (проект): РН 3.1, РН 4.1 – 15 балів/9 балів.

#### - підсумкове оцінювання (у формі екзамену):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН 1.1 – РН 1.3, РН 2.1
- форма проведення і види завдань: письмова форма

Види завдань:

#### Структура екзаменаційної роботи та критерії оцінювання:

1. Теоретичне запитання (РН 1.1 – РН 1.2).
2. Теоретичне запитання (РН 1.2 – РН 1.3).
3. Письмове практичне завдання (РН 2.1).
4. Письмове практичне завдання (РН 2.1).

### Критерії оцінювання екзаменаційної роботи

Завдання	Вид завдання	Максимальний бал (відсоток)	Всього балів (відсотків)
Завдання 1, 2	Теоретичне запитання	по 10 балів (25 %)	20 балів (25 %)
Завдання 3	Письмове практичне завдання	9 балів (22.5 %)	9 балів (22.5 %)
Завдання 4	Письмове практичне завдання	11 балів (27.5 %)	11 балів (27.5 %)
<b>Всього</b>			<b>40 балів (100%)</b>

### Запитання для підготовки до екзамену

1. Поняття програмної інженерії. Моделювання у програмній інженерії.
2. Життєвий цикл програмних систем (ПС). Моделі життєвого циклу ПС. Ітеративно-інкрементні моделі життєвого циклу. Керування ризиками.
3. Візуальне моделювання. Моделювання та CASE-технології. Уніфікована мова моделювання UML. Призначення UML у розрізі проектування ПС. Види діаграм UML. Спрощена стратегія використання UML-діаграм при моделюванні ПС.
4. Засоби розширення UML: стереотипи (stereotype), помічені значення (tagged value), обмеження (constraint). Профілі предметних областей.
5. Діаграми прецедентів. Моделювання контексту та вимог до ПС. Прецеденти. Специфікація прецедентів у Rational Rose. Потоків подій та сценарії.

6. Актори, основні актори. Відношення між акторами та прецедентами. Відношення узагальнення для прецедентів та акторів.
  7. Організація прецедентів. Відношення залежності між прецедентами. Відношення включення (include) та розширення (extend). Варіанти діаграм прецедентів.
  8. Реалізація прецедентів. Використання діаграм послідовностей. Анатомія діаграм послідовності. Двохетапне розроблення діаграм послідовностей. Узгодженість (цілісність) моделей. Діаграми класів-учасників VOPC (View of Participating Classes) прецедентів.
  9. Використання класів при проектуванні ПС. Класи етапу аналізу: прикордонні (boundary) або інтерфейсні класи, класи-сутності (entity), управляючі (control) класи (класи-менеджери). Класи етапу проектування. Діаграми співробітництва (collaboration) та їх використання.
  10. Відношення між класами та їх виявлення (узагальнення, залежність, асоціація, агрегація, композиція). Проектування класів, відношень між класами. Проектування атрибутів та операцій. Пакетування класів.
  11. Діаграми класів та патерни проектування. Структура патернів. Класифікація патернів: породжуючі, структурні, поведінкові.
  12. Приклади патернів: «Singleton», «Adapter», «Proxy», «Decorator», «Composite», «Bridge», «Observer».
  13. Патерн IOC&DI. Ілюстративний приклад.
  14. Патерн IOC&DI. Spring: IoC + декларативний стиль.
  15. Використання діаграм класів для кодогенерації. Кодогенерація та реінженіринг.
  16. Платформа .NET. Основні складові частини CLI. Середовище виконання .NET.
- Загальна система типів CTS.
17. Поняття керованого коду. .NET-компіляція. Міжмовна інтеграція у .NET.
  18. Метадані. Самоопис керованого коду .NET. Механізм рефлексії.
  19. Мова CIL. Just In Time (JIT) компіляція.
  20. Платформа .NET. Збірки .NET.
  21. Платформа .NET. Управління пам'яттю. Збирання сміття.
  22. Типи об'єктів для віддаленої взаємодії. Домени. Типи marshal-by-value. Сerialізація. Типи marshal-by-reference.
  23. Серверна активізація. Режими Singleton та SingleCall. Клієнтська активізація.
  24. Канали. Стандартні типи каналів: TcpChannel та HttpChannel. Конфігурування інфраструктури .NET Remoting: програмне; з використанням конфігураційних файлів.
  25. Основи управління часом життя об'єктів .NET.
  26. Веб-служби (Web Services) та сервісно-орієнтована архітектура (COA). Стандарти веб-служб.
  27. Документування веб-служб: генерація документації для сприйняття людиною (з використанням веб-браузерів), генерація документації, орієнтованої на використання програмами – wsdl-файли.
  28. Розробка веб-служб на платформі .NET. Директива @ WebService. Тест-форми веб-служб. Утиліта .NET WebService Studio як універсальний клієнт.
  29. Розробка веб-служб на платформі Java.
  30. Розробка клієнтських програм для веб-служб на платформі .NET. Утиліта Wsd.exe.
  31. Розробка клієнтських програм для веб-служб на платформі Java.
  32. Протокол SOAP. Конверт, заголовок SOAP-повідомлення.
  33. Стандарти XML, XML-Schema. Простори імен, монікери XML.
  34. Структура wsdl-файлів. Оркестровка Web-сервісів. BPEL (BPEL4WS – Business Process Execution Language for Web Services). Візуальне проектування BPEL-програм.
  35. Основи архітектури WCF. Кінцеві точки.
  36. Прив'язки WCF. Стандартні прив'язки.
  37. Метадані WCF-служб.
  38. WCF. Behavior. Режими інстанціації (instance).

39. WCF. Підтримка асинхронних викликів. Використання транзакцій.
  40. WCF. DataContract.
  41. Сумісність служб ASMX та WCF.
  42. Спрощена архітектура RMI. Особливості програмування RMI/JRMP-проектів.
  43. “Віддалені” інтерфейси та класи реалізації “віддалених” інтерфейсів. Порівняння RMI/JRMP та RMI/ПОР. «Експортування» об'єктів. Статичний метод ExportObject.
  44. RMI/ПОР-проекти. Використання rmic та orbd.
  45. JNDI та конкретні служби іменування. Особливості налаштування JNDI.
  46. (JRMP – ПОР) портабельність Java RMI-проектів.
  47. Сервлети. Сценарій використання сервлетів.
  48. Web-технологія JSP. Основні підходи до реалізації Web-проектів на основі JSP.
- Патерн Model-View-Controller у web-проекуванні.
49. Основи Action-oriented Frameworks: диспетчерський сервлет, mapping; action-класи.
  50. Використання технології action-класів у web-проекуванні. Приклад.
  51. Struts, WebWork, Spring як приклади Action-орієнтованих фреймворків.
  52. ASP.NET-проекти. Життєвий цикл сторінок.
  53. Структура ASP.NET-проектів та їх розробка. Використання технології відокремлення коду. Засоби візуального проектування, управляючі елементи (контроли). Валідація даних.
  54. Проблеми postback-запитів та можливості збереження стану сторінок. Сховані поля ASP.NET-проектів. Використання ViewState та ControlState. Можливості використання Cookie, Session- та Application-контейнерів.
  55. Знайомство з AJAX на модельному проекті електронного магазину.
  56. Засоби ASP.NET AJAX Extensions.
  57. Приклад Java-проекту з AJAX на основі JSP та сервлетів.
  58. Фреймворк Google Web Toolkit (GWT).
  59. AJAX-проекти із використанням GWT.
  60. Розробка Android-програм. Android Developer Tools, Android SDK Manager, Virtual Device Manager. Архітектура Android OS. Розробка клієнтів веб-служб на платформі Android.
  61. Можливості візуального проектування програм під Android OS. Активності (activities) та розмітки (layouts). Віджети. Програмне оперування віджетами. Приклад.
  62. Android OS. Управління активностями. Передача даних в іншу активність. Динамічне наповнення активності віджетами. Приклад.

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

*Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів.*

## **7.2 Організація оцінювання:**

### **Терміни проведення форм оцінювання:**

1. Контрольна робота (тест): до 7 тижня семестру.
2. Контрольна робота (тест): до 14 тижня семестру.
3. Підготовка доповіді (за темами лекційних занять та самостійної роботи): до 10 тижня семестру.
4. Лабораторна робота (проект): до 14 тижня семестру.

Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 10 балів за кожну. Термін перескладання визначається викладачем.



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

Студент має право здавати лабораторну роботу та доповідь 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
<b>Зараховано / Passed</b>	60-100
<b>Не зараховано / Fail</b>	0-59

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лабор.	Сам. р-та
	<b>Частина 1. Основи інженерії програмних систем. Використання UML при проектуванні програмних систем. Патерни проектування</b>			
1	Основи інженерії програмних систем. Уніфікована мова моделювання UML (огляд). Діаграми прецедентів	2	4	4
2	Використання діаграм послідовностей та діаграм класів при проектуванні програмних систем	2	6	6
3	Патерни проектування. Патерн (принцип) IOC&DI та його підтримка у фреймворку Spring	2	6	6
	Всього за частиною 1	6	16	16
	<b>Частина 2. .NET та Java-платформи. Розподілені програмні системи та віддалена взаємодія. Сервісно-орієнтована архітектура та веб-служби</b>			
4	Основи платформи .NET Framework. Міжмовна інтеграція у .NET. Віддалена взаємодія об'єктів .NET (.NET-Remoting)	2	6	4
5	Огляд технологій J2EE. Технологія Java RMI.	2	6	4
6	Сервісно-орієнтована архітектура. Веб-служби.	2	6	4
7	Технологія Windows Communication Foundation	2	6	4
	Контрольна робота 1		2	
	Всього за частиною 2	8	26	16
	<b>Частина 3. Хмарні обчислення та технології. Web-проектування на платформах Java та .NET. Програмування під мобільні платформи</b>			
8	Web-технології Java Servlet та JSP	2	2	4
9	Web-розробка із використанням ASP Net	2	4	4
10	Патерн Model-View-Controller у Web-проектах. Web MVC-проектування. Технологія action servlet. Поняття фреймворку. Action-oriented Frameworks. Огляд Web MVC фреймворків.	2	4	4
11	Проблема валідації даних у Web-проектах. Підтримка AJAX.	2	4	4
12	Хмарні обчислення та технології. Платформа MS Azure.	2	4	4
13	Програмування під мобільні платформи. Приклад розробки під OS Android	2	4	4
	Контрольна робота 2		2	
	Всього за частиною 3	12	24	24
	<b>ВСЬОГО</b>	<b>26</b>	<b>66</b>	<b>56</b>

**Загальний обсяг 180 год., в тому числі:**

Лекцій – 26 год.

Лабораторні заняття - 66 год.

Консультації – 2 год.

Самостійна робота - 56 год.

## 9. Рекомендовані джерела:

### Основна

1. Соммервилл И. Инженерия программного обеспечения – М.: Вильямс, 2002.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. Второе издание М.: Бинум, СПб.: Невский диалект, 2000.
3. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения – СПб.: Питер, 2002.
4. Коберн А. Современные методы описания требований к системам.– М.: Лори, 2002.
5. Буч. Г. , Рамбо Дж. , Джекобсон А. Язык *UML*. Руководство пользователя –М.: ДМК, 2000.
6. Рамбо Дж., Якобсон А., Буч Г. *UML*: Специальный справочник – СПб.: Питер, 2002.
7. Фаулер М., Скотт К. *UML* в кратком изложении – М.: Мир, 1999.
8. Кватрани Т. *Rational Rose 2000* и *UML*. Визуальное моделирование. М.: ДМК, 2001.
9. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования – СПб.: Питер-ДМК, 2001.
10. Эммерих В. Конструирование распределенных объектов – М.: Мир, 2002.
11. Троелсен Э. С# и платформа *.NET*.— СПб.: Питер,2004.—796 с.
12. Платт Д.С. Знакомство с *Microsoft .NET*— М.: Издательско-торговый дом «Русская Редакция »,2001.— 240с.
13. Просиз Дж. Программирование для *Microsoft .NET* — М.: Издательско-торговый дом "Русская Редакция",2003.— 704 стр.
14. Маклин С.,Нафтел Дж.,Уильямс К. *Microsoft .NET Remoting* — М.: Издательско-торговый дом «Русская Редакция »,2003.— 384 с.
15. Рихтер Дж. Программирование на платформе *Microsoft .NET Framework* — М.: Издательско-торговый дом «Русская Редакция »,2003 — 512 стр.
16. *Microsoft Corporation* Основы *Microsoft Visual Studio .NET 2003* М.: Издательско-торговый дом «Русская Редакция », 2003.— 464 стр.
17. Шилдт Г. , Холмс Дж. Искусство программирования на *Java*.
18. Хорстманн К., Корнелл Г. *Java 2*. Библиотека профессионала. Том 2.Тонкости программирования. — М.:Издательский дом "Вильямс", 2002.— 1120 с.
19. Ахмед Х.З., Амриш К.Е. Разработка корпоративных *Java*-приложений с помощью *J2EE* и *UML* . К.: Вильямс, 2002.

## 10. Додаткові ресурси:

20. <http://java.sun.com/javase/5/javatech.html>
21. <http://msdn.microsoft.com>
22. <http://java.sun.com/j2ee/index.jsp>
23. <http://java.sun.com/docs/index.html>
24. <https://developers.google.com/appengine/>
25. <http://www.hibernate.org/>
26. <http://www.springframework.org/>
27. <http://developer.android.com/index.html>