

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**  
**"ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ "**

для студентів

галузь знань	<b>11 «Математика та статистика»</b> <i>(шифр і назва)</i>
спеціальність	<b>113 «Прикладна математика »</b> <i>(шифр і назва спеціальності)</i>
освітній рівень	<b>магістр</b> <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	<b>«ПРИКЛАДНА МАТЕМАТИКА »</b> <i>(назва освітньої програми)</i>
вид дисципліни	<b>вибіркова</b>

Форма навчання	<b>денна</b>
Навчальний рік	<b>2018/2019</b>
Семестр	<b>4</b>
Кількість кредитів ECTS	<b>3</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>екзамен</b>

Викладачі: **к.ф.-м.н., доцент 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 – 2018**

Розробник: Кузенко Володимир Федорович, к.ф.-м.н., доцент 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)

«\_\_\_\_» \_\_\_\_\_ 20\_\_ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «\_\_\_\_» \_\_\_\_\_ 2018 року № \_\_\_\_

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

## 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 розподілених програмних систем, використання сервісно-орієнтованої архітектури, *REST*-архітектури, оволодіння окремими навичками *Web*-програмування та 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 проектування програмного забезпечення;
- основні етапи життєвого циклу ПС;

2. *Вміти:* застосовувати на практиці інструментальні засоби проектування та розробки програмного забезпечення.

3. *Володіти навичками:* програмування із використанням об'єктно-орієнтованих мов.

**3. Анотація навчальної дисципліни (до 700 символів):**

Навчальна дисципліна “Інформаційні технології” є складовою програми підготовки фахівців за другим (магістерським) рівнем вищої освіти галузі знань 11 „Математика та статистика” зі спеціальності 113 „Прикладна математика”, освітньо-наукової програми – „Прикладна математика”.

Дана дисципліна є **дисципліною вільного вибору студента**.

Викладається у 2 семестрі 2 курсу в **обсязі – 90 год.**

(**3 кредити ECTS**) зокрема: *лекції – 10 год., лабораторні – 10 год., консультації – 2 год., самостійна робота – 68 год.* У курсі передбачено **2 частини** та **3 тестові контрольні роботи**. Завершується дисципліна – **екзаменом у 4 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

**знати** основні поняття стосовно моделювання, проектування та розробки програмних систем із використанням віддаленої взаємодії на основі клієнт-серверної архітектури, зокрема, поняття сервісно-орієнтованої архітектури, *REST*-архітектури, *Web*-програмування, хмарних обчислень;

**вміти** застосовувати на практиці програмні засоби, фреймворки та інструментальні середовища при розробці програмних систем, орієнтованих на віддалену взаємодію, зокрема, на використання технологій, спряжених із *Web*, мобільними платформами, хмарними обчисленнями.

Для допуску до дисципліни студент повинен опанувати компетентності та результати навчання, які надають дисципліні “Об'єктно-орієнтоване програмування”, “Інструментальні середовища та технології програмування”.

**4. Завдання (навчальні цілі):**

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні. Зокрема, розвивати:

- здатність використовувати інтелектуальні інформаційні технології;
- здатність використовувати інструментальні засоби розробки клієнт-серверних застосувань;
- здатність забезпечити виконання розподілених обчислень, використання мов програмування при розробці та експлуатації розподіленого програмного забезпечення.
- здатність оцінювати та забезпечувати якість виконуваних робіт;

## 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 віддаленої взаємодії на основі клієнт-серверної архітектури	Лекції, лабораторні роботи	Тест(3%), екзамен(10%)	13%
РН 1.2	Знати суть і підгрунтя сервісно-орієнтованої архітектури, технології <i>Web</i> -сервісів, використання <i>REST</i> -обмежень у розподілених програмних системах.	Лекції, лабораторні роботи	Тест(3%), екзамен(15%)	18%
РН 1.3	Знати суть і підгрунтя мовних засобів, фреймворків та інструментальних середовищ розробки розподілених програмних систем, орієнтованих на віддалену взаємодію із використанням <i>Web</i> -технологій, мобільних платформ, хмарних обчислень.	Лекції, лабораторні роботи	Тест(4%), екзамен(15%)	19%
РН 2.1	Вміти застосовувати <i>на практиці</i> технології розробки програмних систем, що орієнтовані на клієнт-серверну взаємодію	Лабораторні роботи, самостійна робота	Захист лабораторних робіт	10%
РН 2.2	Вміти застосовувати технологію <i>Web</i> -сервісів і, зокрема, <i>REST Web</i> -сервісів у програмних системах.	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 <i>Web</i> -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>						
ПРН3. Отримання знань для здатності проводити оцінку наявних технологій та на основі аналізу формувати вимоги до розроблення перспективних інформаційних технологій.	+	+	+			
ПРН6. Вміння проектувати та використовувати наявні засоби інтеграції даних, опрацьовувати дані, що зберігаються у різних системах.	+	+	+	+	+	+
ПРН7. Вміння організовувати, конфігурувати та розробляти Web-системи, використовуючи принципи розподілених систем, гіпертекстових систем, відповідні технічні та програмні засоби.				+	+	+
ПРН10. Вміння побудови моделей фізичних та виробничих процесів, проектування сховища і простору даних, бази знань, використовуючи діаграмну техніку і стандарти розроблення інформаційних 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 екзамену):

- 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

#### Структура екзаменаційної роботи та критерії оцінювання:

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

### 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. Потoki 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 послідовностей. 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. Пакування класів.
11. Використання діаграм класів для кодогенерації. Кодогенерація та реінженіринг.
12. Веб-служби (Web Services) та сервісно-орієнтована архітектура (COA). Стандарти веб-служб.
13. Розробка веб-служб та клієнтських програм на платформах .NET та Java.
14. Спрощена архітектура RMI. Особливості програмування RMI/JRMP-проектів.
15. Поняття web-проектів. Сервлети. Життєвий цикл сервлетів. GenericServlet. HttpServlet.
16. Контейнерне управління сервлетами (web-сервери; типи HttpServletRequest, HttpServletResponse; використання декларативних засобів у web-проектах.
17. Сервлетні об'єкти їх особливості та використання (отримання HTTP-сторінок; використання HTTP-get та HTTP-post команд у web-проектах; singleton-властивість сервлетних об'єктів; особливості підтримки клієнтських сесій; класи HttpSession, Cookie; JSESSIONID-cookie).
18. Web-проект для спрощеної моделі електронного магазину.
19. Java Server Pages (JSP), їх особливості та використання (трансляція у сервлети; виклики JSP із браузера та сервлетів, "перенаправлення" запитів JSP-сторінці; використання MVC-архітектури при web-проектуванні та роль JSP у ній).
20. Розробка Android-програм. Android Developer Tools, Android SDK Manager, Virtual Device Manager.
21. Можливості візуального проектування програм під Android OS. Активності (activities) та розмітки (layouts). Віджети. Програмне оперування віджетами. Приклад.
22. Android OS. Управління активностями. Передача даних в іншу активність. Динамічне наповнення активності віджетами. Приклад.
23. Крос-платформена розробка мобільних проектів із Xamarin.
24. Хмарні обчислення. Класифікація хмарних сервісів (Platform as a Service, Infrastructure as a Service, Software as a Service). Платформа Microsoft Azure. Розгортання веб-проектів у хмарі (на прикладі).

*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. Тест РН 1.1. : до 5 тижня семестру.
2. Тест РН 1.2. : до 10 тижня семестру.
3. Тест РН 1.3. : до 14 тижня семестру.

У випадку відсутності студента з поважних причин відпрацювання та прездачі МКР 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.

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

<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	Лаб. занят.	Самост. робота
	<b>Частина 1. Об'єктна парадигма та технології віддаленої взаємодії</b>			
1	<b>Тема 1.</b> Уніфікована мова моделювання <i>UML</i> . Спрощена стратегія використання <i>UML</i> -діаграм при моделюванні та проектуванні інформаційних систем .	2	2	12
2	<b>Тема 2.</b> Основні патерни, технології та фреймворки спряжені із віддаленою програмною взаємодією.	2	2	14
	<b>Частина 2. Розробка програмних систем із використанням віддаленої взаємодії</b>			
3	<b>Тема 3.</b> <i>Web-проекти. Проблема валідації даних у Web-проектах. Підтримка AJAX.</i>	2	2	14
4	<b>Тема 4.</b> Програмування під мобільні платформи. Крос-платформена розробка мобільних проектів.	2	2	14
5	<b>Тема 5.</b> Хмарні обчислення та технології. Платформа <i>MS Azure</i> .	2	2	14
	<b>ВСЬОГО</b>	10	10	68

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

Лекцій – **10 год.**

Лабораторні заняття - **10 год.**

Консультації – **2 год.**

Самостійна робота - **68 год.**

## 9. Рекомендовані джерела:

### *Основні:*

1. Боггс У., Боггс М., Дранишников И., UML и Rational Rose, «Лори», М., 2008, 600с.
2. Соммервилл И., Инженерия программного обеспечения. М, «Вильямс», 2002.- 624 с.
3. Лавріщева К. Програмна інженерія, К., 2008, 319 с.
4. Хохгуртль Б., C# и Java: межплатформенные Web-сервисы, М., Кудиц-образ, 2004, 410с.
5. Машнин Т.С. Web-сервисы Java, БХВ-Петербург, 2012, 560 с.
6. Erik Wilde, Cesare Pautasso. REST: From Research to Practice, 2011, 528 p.
7. Джон Фландерс. Введение в службы RESTful с использованием WCF. MSDN, 2009. ([msdn.microsoft.com/ru-ru/magazine/dd315413.aspx](http://msdn.microsoft.com/ru-ru/magazine/dd315413.aspx))
8. Перри Б., Java сервлеты и JSP: сборник рецептов, М., КУДИЦ-Пресс, 2009, 768с.
9. Хемраджани А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse. — М.:Издательский дом "Вильямс", 2008, 352 с.



### Додаткові:

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