

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
"Інструментарій розробки інформаційних систем"

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 **«Бізнес інформатика»**
(назва освітньої програми)

вид дисципліни **дисципліна вільного вибору студента**

Форма навчання

денна

Навчальний рік

2018/2019

Семестр

3

Кількість кредитів ECTS

4

Мова викладання, навчання

та оцінювання

українська

Форма заключного контролю

залік

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 – 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 (ІТ) та інструментальних середовищ розробки інформаційних систем (ІС), зокрема, оволодіння навичками проектування мобільних та *Web-проектів*, використання сервісно-орієнтованої архітектури, мікросервісів, *REST-* та *GraphQL-API*.

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 та, зокрема, ІС;
- основні поняття та фреймворки для віддаленої взаємодії;

2. *Вміти*: застосовувати на практиці інструментальні середовища програмування.

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

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Інструментарій розробки інформаційних систем” є складовою програми підготовки фахівців за другим (*магістерським*) рівнем вищої освіти *галузі знань* 12 „Інформаційні технології” зі *спеціальності* 122 „Комп'ютерні науки” *освітньо-наукової програми* „Бізнес інформатика”.

Викладається у 1 семестрі 2 курсу в **обсязі – 120 год.**

(4 кредити ECTS) зокрема: *лекції – 34 год., консультації – 6 год., самостійна робота – 80 год.* У курсі передбачено **2 частини** та **3 тестові контрольні роботи**. Завершується дисципліна – заліком.

В результаті вивчення навчальної дисципліни студент повинен:

знати основні поняття стосовно моделювання, проектування та розробки інформаційних 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 інформаційних систем, орієнтованих на віддалену взаємодію, зокрема, на використання інформаційних *Web-*технологій та мобільних технологій.

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

4. Завдання (навчальні цілі):

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

- здатність використовувати інтелектуальні інформаційні технології;

- здатність використовувати інструментальні засоби розробки клієнт-серверних, веб- та мобільних проектів;
- здатність забезпечити використання фреймворків при розробці інформаційних систем.
- здатність використовувати хмарні технології.

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 патерни, технології, фреймворки та інструментальні середовища розробки інформаційних систем.	Лекції, лабораторні роботи	Тест(3%), залік(10%)	13%
РН 1.2	Знати суть і підгрунтя сервісо-орієнтованої архітектури (COA), технології <i>Web</i> -сервісів, використання <i>REST</i> -обмежень, <i>GraphQL API</i> у програмних системах.	Лекції, лабораторні роботи	Тест(3%), залік(15%)	18%
РН 1.3	Знати суть і підгрунтя мовних засобів, фреймворків та інструментальних середовищ розробки систем, орієнтованих на використання <i>Web</i> -технологій, мобільних платформ, хмарних обчислень.	Лекції, лабораторні роботи	Тест(4%), залік(15%)	19%
РН 2.1	Вміти застосовувати на практиці технології розробки інформаційних систем, що орієнтовані на клієнт-серверну взаємодію	Лабораторні роботи, самостійна робота	Захист лабораторних робіт	10%
РН 2.2	Вміти застосовувати <i>Web</i> -сервіси і, зокрема, <i>REST Web</i> -сервіси та <i>GraphQL API</i> в інформаційних 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 (з опису освітньої програми)						
ПРН3. Опанувати нові інструменти роботи з даними, здійснюючи пошук та обробку інформації в мережах для прогнозування бізнес-процесів та ситуаційного управління, SWOP-аналізу, відгуків, розробки інформаційно-аналітичних	+	+				

систем для реалізації бізнес процесів в техніці, економічних та соціальних системах, сфері електронної комерції, медіа, соціальних мережах, банкінгу, рекламній діяльності, охороні здоров'я, тощо.						
ПРН6. Розробляти концепції бізнес стратегії при дослідженні задач в організаціях, тенденції використання інформаційних технологій в організаціях, визначати потенціал та можливості відповідних бізнес моделей при їх впровадженні.	+	+	+			
ПРН7. Визначати методологічні принципи та методи наукового дослідження галузі інформаційних технологій в залежності від об'єкту і предмету, використовуючи міждисциплінарний підхід.	+	+	+			
ПРН-11. Демонструвати результати виконаної роботи, створювати презентації, писати звіти та публікації за результатами виконаної роботи.				+	+	+
ПРН-13. Працювати зі студентською аудиторією в галузі інженерії математичного та програмного забезпечення, вміти організувати навчальний процес, запроваджувати інноваційні підходи в освіті.	+	+	+			
ПРН-14.1. Систематичне розуміння відповідних знань про організації, їх зовнішнього контексту та способів їх управління та автоматизації бізнес процесів в організаціях.	+					
ПРН-18.2 Здатним самостійно діяти при плануванні та реалізації проектів на професійному рівні на основі моделей та методів оптимізації та штучного інтелекту в інформаційному середовищі.	+		+			

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Тести: РН 1.1., РН 1.2, РН 1.3 — 20/12 балів.
2. Лабораторні роботи : РН 2.1 — 20/12 балів.
2. Лабораторні роботи : РН 2.2 — 30/18 балів.
4. Лабораторні роботи : РН 2.3 — 30/18 балів.

- підсумкове оцінювання у формі заліку. Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.

Запитання для підготовки до заліку

1. Інструментарій управління проектами (Maven, Gradle).
2. Стратегії розробки програмних систем. Test-Driven Development. Проект xUnit.
3. Автономне тестування проектів. Середовища JUnit, TestNG.
4. Інтеграційне тестування, поняття зазорів (англ. seam) та імітаторів. Використання імітаторів та патерн IOC&DI.
5. Патерн IOC&DI та його підтримка фреймворком Spring.
6. Фреймворк Mockito Інтеграційне тестування. Mock- та Spy- об'єкти.
7. Веб-служби (Web Services) та сервісно-орієнтована архітектура (COA). Стандарти веб-служб.

8. Розробка веб-служб та клієнтських програм на платформах .NET та Java.
9. Розробка для веб-служб на платформі .NET. Утиліта Wsd.exe.
10. Основи архітектури WCF. Кінцеві точки. Прив'язки WCF. Стандартні прив'язки.
11. Метадані WCF-служб. Behavior. Режими інстанціації (instance).
12. WCF. Підтримка асинхронних викликів. Використання транзакцій.
13. REST веб-служби. Ідеологія відповідного архітектурного стилю. JAX-RS (Java API for RESTful Web Services).
14. Поняття web-проектів. Сервлети. Життєвий цикл сервлетів. GenericServlet. HttpServlet.
15. Контейнерне управління сервлетами (web-сервери; типи HttpServletRequest, HttpServletResponse; використання декларативних засобів у web-проектах).
16. 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).
17. Web-проект для спрощеної моделі електронного магазину.
18. Java Server Pages (JSP), їх особливості та використання (трансляція у сервлети; виклики JSP із браузера та сервлетів, "перенаправлення" запитів JSP-сторінки; використання MVC-архітектури при web-проекуванні та роль JSP у ній).
19. Основи Action-oriented Frameworks: диспетчерський сервлет, mapping, action-класи.
20. Знайомство з AJAX на модельному проекті електронного магазину.
21. Засоби ASP.NET AJAX Extensions.
22. Розробка Android-програм. Android Developer Tools, Android SDK Manager, Virtual Device Manager.
23. Можливості візуального проектування програм під Android OS. Активності (activities) та розмітки (layouts). Віджети. Програмне оперування віджетами. Приклад.
24. Android OS. Управління активностями. Передача даних в іншу активність. Динамічне наповнення активності віджетами. Приклад.
25. Крос-платформена розробка мобільних проектів із Xamarin.
26. Хмарні обчислення. Класифікація хмарних сервісів (Platform as a Service, Infrastructure as a Service, Software as a Service). Платформа Microsoft Azure.
27. Розгортання веб-проектів у хмарі (на прикладі).

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

Студент не допускається до заліку, 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

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

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб. занят.	Самост. робота
	Частина 1. Інструментальна підтримка загальних стратегій розробки та управління програмними проектами.			
1	Тема 1. Інструментарії управління проектами (<i>Maven, Gradle</i>).	2		5
2	Тема 2. Стратегії розробки програмних систем. <i>Test-Driven Development</i> . Проект <i>xUnit</i> .	2		5
3	Тема 3. Автономне тестування проєктів. Середовища <i>jUnit, TestNG</i> .	2		5
4	Тема 4. Інтеграційне тестування, поняття зазорів (англ. <i>seam</i>) та імітаторів. Використання імітаторів та патерн <i>IOC&DI</i> .	2		5
5	Тема 5. Патерн <i>IOC&DI</i> та його підтримка фреймворком <i>Spring</i> .	2		5
6	Тема 6. Фреймворк <i>Mockito</i> . Інтеграційне тестування. <i>Mock-</i> та <i>Spy-</i> об'єкти.	3		6
	Частина 2. Використання та інструментальна підтримка спеціальних архітектурно-технологічних принципів розробки інформаційних систем			
7	Тема 7. Сервісно-орієнтована архітектура (COA). Технологія веб-служб.	3		6
8	Тема 8. Поняття про мікросервісну архітектуру та її використання.	2		5
9	Тема 9. Основи архітектури <i>WCF</i> . Кінцеві точки. Режими інстанціації (<i>instance</i>). Підтримка асинхронних викликів. Використання транзакцій.	4		9
10	Тема 10. <i>REST</i> архітектура та <i>REST API</i> у програмних системах.	2		5
11	Тема 11. Проблеми із використанням <i>REST API</i> та перехід до <i>GraphQL-API</i> .	2		5
12	Тема 12. Програмування під мобільні платформи. Приклад розробки під <i>OS Android</i>	4		9
13	Тема 13. Крос-платформена розробка мобільних проєктів (<i>Xamarin</i>).	2		5
14	Тема 14. Хмарні обчислення та технології. Платформа <i>MS Azure</i> .	2		5
	ВСЬОГО	34		80

Загальний обсяг 120 год., в тому числі:

Лекцій – **34 год.**

Консультації – **6 год.**

Самостійна робота - **80 год.**

9. Рекомендовані джерела:

Основні:

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

Додаткові:

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования – СПб.: Питер-ДМК, 2001.
2. Эммерих В. Конструирование распределенных объектов – М.: Мир, 2002.
3. Троелсен Э. C# и платформа .NET.— СПб.: Питер, 2004.—796 с.
4. Платт Д.С. Знакомство с *Microsoft .NET*— М.: Издательско-торговый дом «Русская Редакция », 2001.— 240с.
5. Просиз Дж. Программирование для *Microsoft .NET* — М.: Издательско-торговый дом "Русская Редакция", 2003.— 704 стр.
6. Рихтер Дж. Программирование на платформе *Microsoft .NET Framework* — М.: Издательско-торговый дом «Русская Редакция », 2003 — 512 стр.
7. Шилдт Г. , Холмс Дж. Искусство программирования на *Java*.
8. Хорстманн К., Корнелл Г. *Java 2*. Библиотека профессионала. Том 2. Тонкости программирования. — М.:Издательский дом "Вильямс", 2002.— 1120 с.