

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

МАГІСТР

Кафедра інформаційних технологій

**«ЗАТВЕРДЖУЮ»**

Декан факультету автоматизації і  
інформаційних технологій

  
\_\_\_\_\_ / I.V. Русан /  
« 30 » \_\_\_\_\_ 01 \_\_\_\_\_ 2019 року

**НАВЧАЛЬНА РОБОЧА ПРОГРАМА ДИСЦИПЛІНИ**

"Додаткові обов'язкові компоненти ОПП"

**«СИСТЕМНА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**

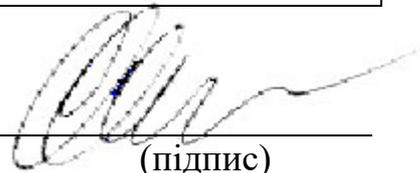
(назва навчальної дисципліни)

шифр	назва спеціальності
121	Інженерія програмного забезпечення
	назва освітньо-наукової програми
	Розподілені програмні системи і технології

Розробник:

Гончаренко Т.А., кандидат технічних наук, доцент

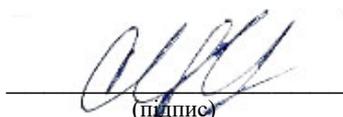
(прізвище та ініціали, науковий ступінь, звання)

  
\_\_\_\_\_ (підпис)

Робоча програма затверджена на засіданні кафедри інформаційних технологій

протокол № 10 від " 11 " \_\_\_\_\_ 02 \_\_\_\_\_ 2019 року

Завідувача кафедри

  
\_\_\_\_\_ (підпис)

(Цюцюра С.В.)  
(прізвище та ініціали)

Схвалено навчально-методичною комісією спеціальності (НМКС):

"Інженерія програмного забезпечення"

протокол № 4 від " 23 " \_\_\_\_\_ 01 \_\_\_\_\_ 2019 року

Голова НМКС

  
\_\_\_\_\_ (підпис)

(Цюцюра С.В.)  
(прізвище та ініціали)

**ВИТЯГ З НАВЧАЛЬНОГО ПЛАНУ 2019-2021 рр.**

шифр	Магістр ОПП	Форма навчання:										Форма контролю	Семестр	Відмітка про погодження
	Назва спеціальності	Кредитів на сем.	Обсяг годин					Кількість індивідуальних робіт						
			Всього	аудиторних										
				Разом	у тому числі			КП	КР	РГР	роб			
Л	Лр	Пз												
121	Інженерія програмного забезпечення	<b>6,0</b>	<b>180</b>	<b>60</b>	<b>30</b>	<b>30</b>				<b>1</b>		<b>Екз.</b>	<b>1</b>	

## Мета та завдання навчальної дисципліни

Програма та тематичний план дисципліни «Системна інженерія програмного забезпечення» орієнтована на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок, використання сучасних програмно-апаратних засобів для проектування, реалізації, верифікації, оптимізації та опису програмних систем з подальшим використанням набутих навичок під час розв'язання актуальних завдань.

Метою дисципліни є формування у студентів теоретичних знань та практичних навичок з розробки програмного забезпечення від етапу формування вимог до створення працездатного та якісного продукту.

### Компетенції здобувачів, що формуються в результаті засвоєння дисципліни

<b>Інтегральна компетентність (ІК)</b>	Здатність розв'язувати складні задачі і проблеми інженерії програмного забезпечення, що передбачає проведення досліджень з елементами наукової новизни та/або здійснення інновацій в умовах невизначеності вимог.
<b>Загальні компетентності (ЗК)</b>	<b>ЗК-1.</b> Здатність до абстрактного мислення, аналізу та синтезу. <b>ЗК-3.</b> Здатність проведення теоретичних та прикладних досліджень на відповідному рівні. <b>ЗК-6.</b> Здатність удосконалювати свої навички на основі аналізу попереднього досвіду.
<b>Фахові компетентності спеціальності (ФК)</b>	<b>ФК-1.</b> Здатність аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення. <b>ФК-2.</b> Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів. <b>ФК-3.</b> Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення. <b>ФК-6.</b> Здатність розробляти і координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення.

<b>Програмні результати навчання</b>	
<b>За загальними та загально-професійними компетентностями (ПРН)</b>	<p><b>ПРН-1.</b> Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення.</p> <p><b>ПРН-4.</b> Аналізувати, оцінювати і вибирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії.</p> <p><b>ПРН-5.</b> Обґрунтовано вибирати парадигми і мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення.</p> <p><b>ПРН-6.</b> Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення.</p> <p><b>ПРН-9.</b> Вміти визначати компоненти, їх функціональність та розміщення для побудови розподілених баз даних та сховищ даних; самостійно обирати відповідні параметри розміщення та реплікації; оптимізувати запити до розподілених БД; розуміти механізми обробки розподілених транзакцій; визначати джерела надходження даних та вміти їх агрегувати.</p>

### **Програма навчальної дисципліни**

#### **Змістовий модуль 1. Проектування та розробка ПЗ.**

##### **Теми 1. Основні поняття програмної інженерії.**

1. Програмне забезпечення.
2. Проектування ПЗ.
3. Фази проектування ПЗ.
4. Програмний продукт.
5. Критерії успішності проекту.

##### **Тема 2. Процес створення ПЗ.**

1. Методології проектування ПЗ.

2. Характеристики методологій проектування.
3. Стратегії конструювання ПЗ.
4. Адаптивність процесу конструювання.
5. Вибір методології проектування.

### **Тема 3. Міжнародні стандарти життєвого циклу ПЗ.**

1. Класична модель проектування ПЗ.
2. Каскадна модель проектування.
3. Прототипування.
4. Інкрементна модель.
5. Спіральна модель.
6. Інкрементна спіральна модель.
7. Швидка розробка програм (RAD).
8. Rational Unified Process.
9. Гнучкі методології (Agile).
10. Екстремальне програмування, його методи.
11. Методологія SCRUM.
12. Технологічні підходи до проектування ПЗ.

### **Тема 4. Інженерія вимог до ПЗ.**

1. Вимоги, їх властивості, види і обмеження.
2. Розробка вимог, їх виявлення і аналіз.
3. Специфікація вимог.
4. Документування і організація вимог.
5. Зміна вимог.
6. Планування і управління вимогами.
7. Аналіз, специфікація, верифікація та валідація вимог до ПЗ.
8. Шаблони специфікації вимог.
9. Стандарти вимог.

### **Тема 5. Проектування архітектури ПЗ.**

1. Основні поняття проектування ПЗ.
2. Структурування систем.
3. Моделі управління програм.
4. Модульна декомпозиція.
5. Проблемно-залежні архітектури ПЗ.
6. Шаблони проектування ПЗ.

### **Тема 6. Проектування інтерфейсу користувача.**

1. Інтерфейс користувача.
2. Принципи проектування інтерфейсів користувача.
3. Представлення інформації.
4. Засоби підтримки користувача.
5. Оцінювання інтерфейсу.

### **Змістовий модуль 2. Моделювання ПЗ.**

#### **Тема 7. Класичні методи аналізу і проектування ПЗ.**

1. Методологія моделювання SADT, IDEF, DFD, ELM, ООАД.
2. Мови моделювання.

3. Методи структурного проектування.
4. Діаграми потоків даних.
5. Методи, орієнтовані на структури даних.
6. Метод проектування Джексона.
7. Особливості класичних методів.

#### **Тема 8. Об'єктно-орієнтований аналіз і проектування.**

1. Основні поняття об'єктного підходу.
2. Об'єкти і класи об'єктів.
3. Ієрархія класів.
4. Процес об'єктно-орієнтованого проектування.
5. Визначення об'єктів.
6. Моделі архітектури.
7. Модифікація системної архітектури.

#### **Тема 9. Архітектура розподілених програмних систем.**

1. Розподілені системи.
2. Багатопроцесорна архітектура.
3. Архітектура клієнт/сервер.
4. Архітектура розподілених об'єктів. CORBA.

#### **Змістовий модуль 3. Менеджмент.**

#### **Тема 10. Управління програмними проектами.**

1. Задачі управління проектами.
2. Трикутник обмежень.
3. Ресурси в програмних проектах.
4. Співробітники, процеси і ролі.
5. Планування ресурсів.
6. Менеджмент проекту.
7. Особливості програмних проектів, їх задачі.
8. виправлення програмних дефектів.
9. Етапи і віхи програмних проектів.

#### **Тема 11. Управління ризиками в програмних проектах.**

1. Управління ризиками.
2. Ідентифікація ризиків.
3. Аналіз ризиків.
4. Планування управління ризиками.
5. Оцінка ризиків.
6. Методи управління ризиками.
7. Стандарти управління ризиками.

#### **Тема 12. Системи управління дефектами.**

1. Дефекти. Їх характеристики.
2. Ідентифікатор, стан і зміст дефекту.
3. Життєвий цикл дефекту.
4. Класифікація систем управління дефектами.
5. Їх основні характеристики.

## **Змістовий модуль 4. Методи забезпечення та контролю якості ПЗ.**

### **Тема 13. Контроль версій в програмних проектах.**

1. Групова робота з проектом.
2. Спільний доступ.
3. Збереження історії модифікацій.
4. Системи контролю версій.
5. Типи систем контролю версій.
6. Робота з системами контролю версій.
7. Система CVS, Subversion, SmartSVN.

### **Тема 14. Збір і випуск ПЗ.**

1. Задачі збору і випуску проектів.
2. Причини збору проектів.
3. Проблеми збору проектів.
4. Процедури і засоби збору.
5. Утиліти збору.
6. Неперервна інтеграція.
7. Дії для забезпечення неперервної інтеграції.
8. Умови запуску збору.
9. Промислові засоби неперервної інтеграції.

### **Тема 15. Якість програмного забезпечення.**

1. Тенденції індустрії розробки ПЗ.
2. Якість ПЗ: метрики і стандарти якості ПЗ.
3. Забезпечення якості ПЗ.
4. Методи забезпечення якості.
5. Верифікація та валідація ПЗ.
6. Тестування ПЗ.

## **Змістовий модуль 5. Засоби та середовища створення ПЗ.**

### **Тема 16. Інструментальні засоби інженерії програмного забезпечення.**

1. Інтегровані середовища розробки ПЗ.
2. Системи управління проектами.
3. Системи управління версіями документів, архітектурні особливості.
4. Інструменти автоматизації зборки проектів.
5. Інструменти автоматизації процесів тестування.
6. Сервери постійної інтеграції.

## **Тематика лабораторних занять**

### **Лабораторна робота №1.**

**Тема:** Аналіз предметної області та розробка технічного завдання.

**Мета:** Провести аналіз предметної області та отримати практичні навички по формуванню технічного завдання.

### **Лабораторна робота №2.**

**Тема:** Розробка діаграми класів.

**Мета:** Отримати практичні навички по створенню діаграми класів.

**Лабораторна робота №3.**

**Тема:** Розробка діаграми станів.

**Мета:** Отримати практичні навички по створенню діаграми станів.

**Лабораторна робота №4.**

**Тема:** Розробка діаграми послідовності.

**Мета:** Отримати практичні навички по створенню діаграми послідовності.

**Лабораторна робота №5.**

**Тема:** Розробка діаграми компонентів.

**Мета:** Отримати практичні навички по створенню діаграми компонентів.

**Лабораторна робота №6.**

**Тема:** Розробка діаграми кооперації.

**Мета:** Отримати практичні навички по створенню діаграми кооперації.

### **Індивідуальне завдання**

1. Комп'ютерна система розпізнавання статичних зображень.
2. Комп'ютерна система аналізу відеопотоків.
3. Комп'ютерна система аналізу змін напруги в електричній мережі.
4. Комп'ютерна система документообігу в бібліотеці.
5. Комп'ютерна система перевірки відвідування студентами занять на основі безконтактних карток.
6. Комп'ютерна система інвентаризації товарів в комп'ютерному магазині на основі штрих-кодів.
7. Комп'ютерна система обробки звукової інформації.
8. Комп'ютерна система проти пожежної охорони для приміщення.
9. Комп'ютерна система віртуального робочого місця для трейдера електронних бірж.
10. Комп'ютерна система розпізнавання автомобільних знаків.
11. Комп'ютерна система аналізу повітря на вміст шкідливих газів.
12. Комп'ютерна система аналізу води на вміст шкідливих речовин.
13. Комп'ютерна система обробки графічних зображень великого розміру.
14. Комп'ютерна система контролю тиску в резервуарах великого розміру.
15. Комп'ютерна система для моделювання 3-D об'єктів.
16. Комп'ютерна система для верстки документів.
17. Комп'ютерна система для створення анімаційних документів.
18. Комп'ютерна система аналізу друкованих текстів на плагіат.
19. Комп'ютерна система автоматичного детектування облич.
20. Комп'ютерна система для діагностики персонального комп'ютера.

## Самостійна робота студентів

№ п/п	Тематика
1	Історія розвитку програмної інженерії.
2	Розвиток модульного програмування.
3	Розвиток структурного програмування.
4	Основні положення об'єктного програмування.
5	Методи швидкої розробки програм.
6	Інструментальні засоби програмної інженерії.
7	Засоби тестування і наладки програм.
8	Комп'ютерна підтримка аналізу вимог до програмних виробів.
9	Інструментальне забезпечення життєвого циклу програм.
10	Мови розробки вимог до програмного виробу.
11	Мови підтримки структурного програмування.
12	Мови підтримки об'єктно-орієнтованого програмування.
13	Візуальні засоби в програмній інженерії.
14	Основні елементи графічного інтерфейсу користувача.
15	Основні етапи впровадження програмного забезпечення.

## Тренінг

№ п/п	Вид роботи	Порядок проведення тренінгу
1	Огляд сучасних середовищ моделювання комп'ютерних програм	1. розгляд сучасних середовищ моделювання програм фірм Visual Paradigm, Microsoft; 2. дослідження мови графічного опису для об'єктного моделювання на основі UML діаграм
2	Розгляд процесу моделювання комп'ютерних програм	1. постановка задачі; 2. опис технічного завдання; 3. моделювання програмної розробки на основі UML діаграм
3	Програмна реалізація проєктованого додатку	1. Реалізація розробленої програми на об'єктно-орієнтованій мові програмування; 2. реалізація графічного інтерфейсу користувача.
4	Тестування розробленого програмного додатку	1. вибір та обґрунтування тестової вибірки; 2. опис вхідних та вихідних даних; 3. перевірка правильності роботи реалізованого програмного додатку

## **Методи контролю та оцінювання знань здобувачів Політика щодо академічної доброчесності**

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) перевіряються на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку студента він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

### **Політика щодо відвідування**

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

Здобувач, який пропустив лекційне заняття, повинен законспектувати зміст цього заняття та продемонструвати конспект викладачу до складання заліку.

Здобувач, який пропустив практичне заняття, повинен законспектувати джерела, які були визначені викладачем як обов'язкові для конспектування, та продемонструвати конспект викладачу до складання заліку, а також виконати індивідуальне завдання, якщо його виконання було передбачене планом заняття.

За об'єктивних причин (хвороба, міжнародне стажування тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

### **Методи контролю**

Основні форми участі здобувачів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, запитання до виступаючого, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується здобувачами у тій чи іншій формі, наведеній вище. Обов'язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх семінарських занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

При оцінюванні рівня знань здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;
- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;
- ступінь сформованості уміння поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;
- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;
- досвід творчої діяльності: уміння виявляти проблеми, розв'язувати їх, формувати гіпотези;
- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, уміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

**Тестове опитування** може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

**Індивідуальне завдання** підлягає захисту здобувачем на заняттях, які призначаються додатково.

Індивідуальне завдання може бути виконане у різних формах. Зокрема, здобувачі можуть зробити його у вигляді реферату. Реферат повинен мати обсяг від 18 до 24 сторінок А4 тексту (кегель Times New Roman, шрифт 14, інтервал 1,5), включати план, структуру основної частини тексту відповідно до плану, висновки і список літератури, складений відповідно до ДСТУ 8302:2015. В рефераті можна також помістити словник базових понять до теми. Водночас індивідуальне завдання може бути виконане в інших формах, наприклад, у вигляді презентації у форматі Power Point. В цьому разі обсяг роботи визначається індивідуально – залежно від теми.

Література, що рекомендується для виконання індивідуального завдання, наведена у цій робочій програмі, а в електронному вигляді вона розміщена на Освітньому сайті КНУБА, на сторінці кафедри.

Також як виконання індивідуального завдання за рішенням викладача може бути зарахована участь здобувача у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту дисципліни, або публікація статті на одну з таких тем в інших наукових виданнях.

Текст індивідуального завдання подається викладачу не пізніше, ніж за місяць до початку залікової сесії. Заняття із захисту індивідуальних завдань призначаються не пізніше, ніж за 2 тижні до початку сесії. Викладач має право вимагати від здобувача доопрацювання індивідуального завдання, якщо воно не відповідає встановленим вимогам.

Результати поточного контролю заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності здобувачів за відсутності пропущених та невідпрацьованих семінарських занять та позитивні оцінки за індивідуальну роботу є підставою до підсумкової форми контролю – заліку. Бали за аудиторну роботу відпрацьовуються у разі пропусків.

**Підсумковий контроль** здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контролю. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою. Участь в роботі впродовж семестру – 100.

Форма підсумкового контролю – іспит.

Бали нараховуються за наступним співвідношенням:

- семінарські завдання 30% семестрової оцінки;
- індивідуальна робота 30 % семестрової оцінки;
- модульний: тестовий (заліковий) – 40 % семестрової оцінки.

#### Розподіл балів, які отримують здобувач

Змістовний модуль 1	Змістовний модуль 2	Змістовний модуль 3	Змістовний модуль 4	Змістовний модуль 5	Разом
20	20	20	20	20	100

#### Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 – 100	<b>A</b>	Зараховано
82-89	<b>B</b>	
74-81	<b>C</b>	
64-73	<b>D</b>	
60-63	<b>E</b>	
35-59	<b>FX</b>	Не зараховано з можливістю повторного складання
<u>0-34</u>	<b>F</b>	Не зараховано з обов'язковим повторним вивченням дисципліни

## Умови допуску до підсумкового контролю

Здобувачу, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Здобувач, який не здав та/або не захистив індивідуальне завдання, не допускається до складання заліку.

Здобувач, який не виконав вимог робочої програми по змістових модулях, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання по змісту відповідних змістових модулів в період між основною та додатковою сесіями.

Здобувач має право на опротестування результатів контролю (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться до здобувачів на початку вивчення дисципліни.

### Рекомендовані джерела

#### Основна

1. Sommerville I. Software Engineering, 10<sup>th</sup> ed. — Addison-Wesley / Pearson Education Limited, 2015. — 816 p. 2. Сомервилл И. Инженерия программного обеспечения, 6 изд. — М.: "Вильямс", 2002. — 624 с.

2. Pressman R. Software Engineering: A Practitioner's Approach, 7<sup>th</sup> ed. — McGraw-Hill, 2010. — 928p.

#### Додаткова

1. McConnell S., Code Complete: A Practical Handbook of Software Construction, Second Edition. — Microsoft Press, 2004. — 960 p.

2. Ghezzi C., Jazayeri M., Mandriol D. Fundamentals of Software Engineering, 2<sup>nd</sup> ed. — Prentice Hall, 2003. — 604p.

3. Лавріщева К.М. Програмна інженерія. — Київ, 2008. — 319с. 8. Липаев В. Программная инженерия. Методологические основы. — М.: Теис, 2006. — 608с.

4. Guide to Software engineering body of knowledge – 2004 Version / Exec. eds: Alain Abran, James W. Moore. — IEEE Computer Society, 2004. — 200p. — <http://www.computer.org/portal/web/swebok/htmlformat>

5. Wirth Niklaus. A Brief History of Software Engineering // IEEE Annals of the History of Computing. — v. 30, No. 3, 2008. — pp. 32–39.

6. ISO/IEC 12207:2008, Systems and software engineering – Software life cycle processes. — ISO, 2008. — 123p.

7. Software Engineering Code of Ethics and Professional Practice (Version 5.2). — <http://www.acm.org/about/se-code>

8. Viller S., Sommerville I. Ethnographically informed analysis for software engineers. // Int. J. HumanComputer Studies. — 53 (1), 2000. — pp.169–196.

9. Mellor S., Balcer M. Executable UML: A Foundation for Model-Driven Architecture. — AddisonWesley, 2002. — 416p.
10. Object Management Group: Semantics of a foundational subset for executable UML models (fUML), v1.0. — OMG, 2011. — 404p. — <http://www.omg.org/spec/FUML/>.
11. Object Management Group: Concrete Syntax For UML Action Language (Action Language For Foundational UML – ALF). — OMG, 2010. — 425p. — <http://www.omg.org/spec/ALF/>.
12. Schwaber K., Sutherland J. — The Scrum Guide: The Definitive Guide to Scrum – The Rules of the game. — 2011. — 17p. — <http://www.scrum.org/Scrum-Guides>.
13. Microsoft Application Architecture Guide, 2<sup>nd</sup> ed. / Microsoft Patterns & Practices Team. — Microsoft Press, 2009. — 496p. — <http://msdn.microsoft.com/en-us/library/dd673617.aspx>
14. Руководство Microsoft по проектированию архитектуры приложений, 2-е изд. / Microsoft Patterns & Practices Team. — Microsoft Press, 2010. — 529p. — <http://apparchguide.ms/>
15. Leffingwell D. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise. — Addison-Wesley, 2011. — 518p.

### **Інформаційні ресурси**

<http://library.knuba.edu.ua/>