

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО**

**Факультет обчислювальної техніки,  
інтелектуальних та управляючих систем**

**Кафедра програмного забезпечення  
автоматизованих систем**

# ***Аналіз вимог до програмного забезпечення***

**НАВЧАЛЬНА ПРОГРАМА**

**Черкаси 2012**

**УДК 004.3, 004.9**  
**ББК 32.97**

**Аналіз вимог до програмного забезпечення:** Навчальна програма /  
Розробник: О.О. Супруненко. – Черкаси: Вид. від. Черкаського національного  
університету імені Богдана Хмельницького, 2012. – 24 с.

**Розробник:**

**О.О. Супруненко**, кандидат технічних наук, доцент кафедри ПЗАС.

**Рецензенти:**

Профільна кафедра: кафедра обчислювальної техніки Національного  
технічного університету України «Київський політехнічний інститут»;

**В.В. Кузьмук**, доктор технічних наук, професор, Відділення гібридних  
модельючих і управляючих систем в енергетиці ІПМЕ НАН України,

**Б.О. Онищенко**, кандидат технічних наук, доцент кафедри програмного  
забезпечення автоматизованих систем ЧНУ.

*Затверджено Вченою радою Черкаського національного  
університету імені Богдана Хмельницького  
(протокол № 1 від 28 серпня 2012 р.)*

**ББК 32.97**

**УДК 004.3, 004.9**

© Черкаський національний університет, 2012

© О.О. Супруненко, 2012

## СТРУКТУРА ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### „Аналіз вимог до програмного забезпечення”

(за вимогами ECTS)

## ОПИС ПРЕДМЕТА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальна характеристика навчальної дисципліни	Напрямок, спеціальність, освітньо-кваліфікаційний рівень	Характеристика навчального курсу
<p>Кількість кредитів ECTS: <b>4</b></p> <p>Модуль: <b>(3 модулі) + (індивідуальне завдання).</b></p> <p>Змістових модулів: <b>3</b></p> <p>Загальна кількість годин: <b>144</b></p> <p>Тижневих годин: <b>8,5</b></p>	<p>Шифр та назва напрямку - <b>050103 Програмна інженерія</b></p> <p>Шифр та назва спеціальності: <b>8.05010300 – програмне забезпечення систем</b></p> <p>Освітньо-кваліфікаційний рівень: <b>бакалавр</b></p>	<p><b>Обов'язковий курс.</b></p> <p>Рік підготовки: <b>2012-2013 н.р.</b></p> <p>семестр: <b>6 семестр</b></p> <p>Лекції (теоретична підготовка): <b>24</b></p> <p>Семінари (практичні): <b>не передбачені</b></p> <p>Лабораторні заняття: <b>34</b></p> <p>Самостійна робота: <b>44</b></p> <p>Індивідуальна робота (індивідуальне навчально-дослідне завдання): <b>42</b></p> <p>Вид підсумкового контролю: <b>іспит</b></p>

## ВСТУП

**Дисципліна «Аналіз вимог до програмного забезпечення»** викладається для студентів третього курсу напряму підготовки «Програмна інженерія». Протягом вивчення дисципліни студенти освоюють основні способи виявлення і уточнення користувацьких вимог до програмного забезпечення, перетворення користувацьких вимог у технічні вимоги, формування нефункціональних вимог, що визначають якість сучасних складних програмних проектів. Вони мають навчитися використовувати освоєні методи формування, документування та аналізу технічних вимог для використання на початковій стадії програмного проекту та протягом всього проекту у робочому процесі управління вимогами. Для ефективного використання набутих знань студенти освоюють спеціалізоване програмне забезпечення, що застосовується у сучасних технологіях розробки програмного забезпечення.

**Мета викладання дисципліни** дати студентам знання про основні вимоги до програмного забезпечення та шляхи їх отримання, про специфікацію та документування вимог, мови написання специфікацій, про основи інженерії вимог до програмного забезпечення, про типи вимог та аналіз їх якості, узгодження вимог та управління ризиками програмних проектів, що викликані зміною та модифікацією вимог, про верифікацію та атестацію програмних продуктів, про основи моделювання аспектів досліджуваних програмних систем, типи моделей, про основні концепції уніфікованої мови програмування UML.

**Завдання вивчення дисципліни. Знання та вміння.** Забезпечити умови для освоєння студентами знань про основні вимоги до програмного забезпечення, про специфікації та документування вимог, мови написання специфікацій, про типи вимог, їх отримання, узгодження та аналіз їх якості, про верифікацію та атестацію програмних продуктів, як підґрунтя для формування проектних рішень, які створюються студентами для реалізації практичних завдань, що стосуються створення технічних завдань та початкової технічної документації для формування програмних проектів. Створити умови для розширення уявлення про сучасні ризики при створенні програмного забезпечення, про верифікацію та атестацію програмних продуктів з метою

підвищення якості проектних робіт, навичок реалізації початкових етапів програмних проектів.

Студент повинен *знати*: основні вимоги до програмного забезпечення, типи вимог, методи та джерела їх отримання, основи інженерії вимог до програмного забезпечення, специфікації та документування вимог, мови написання специфікацій, аналіз якості вимог до ПЗ, узгодження вимог та управління ризиками, верифікацію та атестацію програмних продуктів, основи моделювання досліджуваних об'єктів та програмних систем, типи моделей, основні концепції уніфікованої мови програмування UML.

Студент повинен *вміти*: визначати джерела вимог до програмного забезпечення та отримувати з них необхідну інформацію, формувати ясні, лаконічні та достатньо формалізовані описи вимог для створення нових і розширення існуючих програмних систем, розробляти специфікації вимог користувачів до програмного забезпечення, будувати моделі вимог до програмного забезпечення, здійснювати аналіз вимог, виконувати верифікацію та атестацію моделі вимог, моделювати різні аспекти систем, для яких створюється ПЗ, користуючись діаграмами уніфікованої мови програмування UML, проектувати простий графічний користувацький інтерфейс.

**Компетенції, які набуває студент при освоєнні дисципліни:** здатність здійснювати виявлення, документування та аналіз користувацьких вимог до програмного забезпечення, здатність перетворювати користувацькі вимоги в набір технічних вимог (функціональних та нефункціональних), створювати моделі вимог до програмного забезпечення та трансформувати їх у аналітичну та проектну моделі програмного проекту, здатність використати візуальні програмні засоби для формування вимог та управління ними на протязі виконання програмного проекту з метою зниження проектних ризиків та підвищення якості розроблюваного програмного продукту.

**Міждисциплінарні зв'язки. Перелік дисциплін із зазначенням розділів (тем), засвоєння яких студентам необхідно для вивчення даної дисципліни** Комп'ютерна дискретна математика, Основи програмування, Основи програмної інженерії, Об'єктно-орієнтоване програмування, Алгоритми та структури даних, Архітектура та проектування програмного забезпечення, Архітектура комп'ютерів, Операційні системи, Групова динаміка та комунікації, Конструювання програмного забезпечення, Людино-машинна взаємодія.

## **1.1. Тематика та зміст дисципліни**

### **ЗМІСТОВИЙ МОДУЛЬ 1**

#### **Визначення вимог до програмного забезпечення.**

***Тема 1. Класифікація програмних систем. Етапи розробки ПЗ. Роль вимог у плануванні програмних проектів.***

Предмет та задачі дисципліни «Аналіз вимог до програмного забезпечення». Класифікація програмних систем. Найбільш розповсюджені стандарти і технології розробки програмних систем (MRPII, CRM, ERP та ін.). Методологія структурного аналізу та проектування SADT. Методологія розробки програмного забезпечення RUP. Приклади вдалого і невдалого формування вимог до програмних проектів.

***Тема 2. Вимоги до програмного забезпечення. Класифікація вимог. Користувацькі вимоги. Методи виявлення та опрацювання вимог у технологіях MSF, RUP, XP.***

Типи вимог до програмного забезпечення. Класифікація вимог. Джерела вимог. Якість вимог. Користувацькі вимоги. Формалізація вимог. Функціональні та не функціональні вимоги. Методи виявлення та опрацювання вимог: інтерв'ювання, анкетування, активне і пасивне спостереження, опитування, вивчення технічної документації. Технологій повторного використання проектних рішень. Виявлення вимог за технологіями MSF, RUP, XP. Користувацькі історії, прототипування, сценарії використання, мозковий штурм.

***Тема 3. Функціональні й нефункціональні вимоги, особливості реалізації у програмних проектах.***

Функціональні вимоги. Вплив на структуру та реалізацію програмного проекту. Відображення нефункціональних вимог у структурі програмного проекту. Рекомендації до структури й методів опису програмних вимог – стандарт IEEE 830 (“Recommended Practice for Software Requirements Specifications”).

Специфікація вимог до програмного проекту. Типи специфікацій та їх застосування. Програмні засоби для ведення специфікацій та управління вимогами.

***Тема 4. Перевірка вимог. Змінення та модифікація вимог до програмного проекту. Моделі розробки програмного забезпечення.***

Перевірка вимог до програмного продукту: верифікація, валідація та тестування. Моделі розробки програмного забезпечення. Методології сімейства IDEF. Сучасні методології розробки програмного забезпечення SWAT, JAD, RAD. Гнучкі (agile) методології розробки програмного забезпечення.

***Тема 5. Вплив аналізу вимог по ПЗ на ітеративні цикли розробки проекту. Управління вимогами. Управління якістю. Управління рішеннями.***

Вплив аналізу вимог до ПЗ на ітеративні цикли розробки проекту. Управління вимогами (Requirement Management). Управління якістю розробки ПЗ. Стандарт ISO 9000. Управління рішеннями. Стандарт COBIT. Бізнес-рішення і управління рішеннями. Бібліотека інфраструктури інформаційних технологій ITIL.

## **ЗМІСТОВИЙ МОДУЛЬ 2**

### **Основи візуального моделювання програмних систем**

***Тема 6. Технологія RUP. Уніфікована мова моделювання UML 2.0. Три групи діаграм моделювання програмних систем. Шаблони проектування. Програмне середовище Rational Rose.***

Концепції проектування програмного забезпечення. Технологія RUP. Процес розробки ПЗ. Методи виявлення та документування вимог. Уніфікована мова моделювання UML, історія створення та основні концепції. Три групи діаграм моделювання програмних систем у UML 2.0. Типи класифікаторів та відношень. Засоби автоматизації проектування ПЗ компанії Rational Software. Програмне середовище Rational Rose. Шаблони проектування: абстрактне-конкретне (abstraction-occurrence), компоновщик (composite), гравець-роль (player-role), одинак (singleton), спостерігач (observer), делегування (delegation), фасад (facade), адаптер (adapter).

***Тема 7. Проектування, орієнтоване на користувача. Діаграми поведінки. Аналіз та коректування користувацьких вимог. Користувацькі інтерфейси.***

Проектування, орієнтоване на користувача. Модель аналізу вимог до програмного забезпечення. Діаграми поведінки при представленні моделей програмної системи: діаграма прецедентів (Use Case), діаграма діяльності (Activity), діаграма автомата (State Machine). Нотації та ідеологія. Діаграми бізнес-прецедентів та прецедентів, побудова, аналіз та деталізація. Мова обмежень OCL. Узгодження користувацьких вимог. Користувацькі інтерфейси. Клієнтський досвід. Стабілізація користувацьких вимог.

***Тема 8. Реалізація підсистем. Діаграми структур. Розробка архітектури програмної системи.***

Діаграми структур при моделюванні програмних систем: діаграма класів (Class), діаграма об'єктів (Object), діаграма компонентів (Component), діаграма композитних структур (Composite structure), діаграма розвертування (Deployment), діаграма пакетів (Package). Аналітична та програмна моделі програмної системи. Розробка діаграм бізнес-класів та класів. Типи зв'язків у діаграмі класів. Визначення операцій у класах. Документування операцій. Створення атрибутів та методів. Діаграма об'єктів. Деталізація при розробці моделі ПЗ. Розробка архітектури програмної системи: розподілені архітектури, «канали-та-фільтри» (pipe-and-filter), «модель-вигляд-контролер» (model-view-controller).

***Тема 9. Принципи розробки програмного забезпечення. Діаграми взаємодії.***

Загальні принципи розробки програмного забезпечення: декомпозиція (decomposition), незв'язність (decoupling), зачеплення (cohesion), повторне використання (reuse), можливість повторного використання (reusability), переносимість (portability), тестованість (testability), гнучкість (flexibility). Діаграми взаємодії при моделюванні програмних систем: діаграми послідовностей (Sequence) та комунікації (Communication), діаграма огляду взаємодії (Interaction Overview) та синхронізації (Timing).



## **ЗМІСТОВИЙ МОДУЛЬ 3**

### **Інженерія вимог до програмного забезпечення**

#### ***Тема 10. Розробка вимог. Нормативи та стандарти. ЕСПД.***

Етапи розробки вимог до програмних систем. Специфікація вимог. Стандарти і методології проектування бізнес-процесів. Єдина система програмної документації (ЕСПД). Стадії і етапи розробки. Оформлення текстових програмних документів. Технічне завдання. Оформлення експлуатаційних документів.

#### ***Тема 11. Технічне завдання на програмний проект. Складові та їх зміст.***

Технічне завдання, етапи розробки та зміст. Етапи реалізації технічного завдання на стадіях ескізного, технічного та робочого проекту. Керування пріоритетами. Технічне завдання на розробку програмної системи, роль глосарію у стабілізації технічних вимог.

#### ***Тема 12. Верифікація та тестування програмного забезпечення. Розробка програмного забезпечення через тестування. Впровадження програмної системи.***

Верифікація та тестування програмного забезпечення. Керування якістю розробки програмних продуктів. Технологія XP. Розробка програмного забезпечення через тестування (test-first programming). Впровадження програмних систем. Документація, що розробляється для впровадження програмного проекту.

## 1.2. Структура залікового кредиту навчальної дисципліни

Тема	Кількість годин, відведених на:				
	Лекції	Практично-семинарські заняття	Лабораторні заняття	Самостійну роботу	Індивідуальну роботу
1	2	3	4	5	6
<b>Змістовий модуль I. Визначення вимог до програмного забезпечення.</b>					
<b>Тема 1.</b> Класифікація програмних систем. Етапи розробки програмного забезпечення. Роль вимог у плануванні програмних проєктів.	2			4	2
<b>Тема 2.</b> Вимоги до програмного забезпечення. Класифікація вимог. Користувацькі вимоги. Методи виявлення та опрацювання вимог у технологіях MSF, RUP, XP.	2		4	4	4
<b>Тема 3.</b> Функціональні й нефункціональні вимоги, особливості реалізації у програмних проєктах.	2		2	2	2
<b>Тема 4.</b> Перевірка вимог. Змінення та модифікація вимог до програмного проєкту. Моделі розробки ПЗ.	2		2	4	4
<b>Тема 5.</b> Вплив аналізу вимог по ПЗ на ітеративні цикли розробки проєкту. Управління вимогами. Управління якістю. Управління рішеннями.	2			4	4

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<b>Змістовий модуль II. Основи візуального моделювання програмних систем.</b>					
<b>Тема 6.</b> Технологія RUP. Уніфікована мова моделювання UML 2.0. Три групи діаграм моделювання програмних систем. Шаблони проектування. Програмне середовище Rational Rose.	2		4	4	4
<b>Тема 7.</b> Проектування, орієнтоване на користувача. Діаграми поведінки. Аналіз та коректування користувацьких вимог. Користувацькі інтерфейси.	2		4	2	2
<b>Тема 8.</b> Розробка архітектури програмної системи. Реалізація підсистем. Діаграми структур.	2		6	6	4
<b>Тема 9.</b> Принципи розробки програмного забезпечення. Діаграми взаємодії.	2		4	4	4
<b>Змістовий модуль III. Інженерія вимог до програмного забезпечення.</b>					
<b>Тема 10.</b> Розробка вимог. Нормативи та стандарти. ЕСПД.	2		2	4	4
<b>Тема 11.</b> Технічне завдання на програмний проект. Складові та їх зміст.	2		2	2	2
<b>Тема 12.</b> Верифікація та тестування програмного забезпечення. Розробка програмного забезпечення через тестування. Впровадження програмної системи.	2		4	4	4
<b>Всього годин</b>	<b>24</b>		<b>34</b>	<b>44</b>	<b>42</b>

## **1.3. Теми лабораторних занять**

### **ЗМІСТОВИЙ МОДУЛЬ 1**

#### **Визначення вимог до програмного забезпечення**

***Тема 1. Методології розробки програмного забезпечення. Технологія Rational Unified Process (RUP). Технологія MSF.***

*Формування вимог до програмного забезпечення (ПЗ). Основні властивості вимог до ПЗ. Джерела вимог. Характеристики якості вимог. Методи виявлення вимог. Технологія Rational Unified Process, категорії вимог. Технологія MSF, користувацькі історії, аналіз та уточнення вимог. Роль глосарію у формуванні однозначних і зрозумілих вимог. Розробка набору вимог на прикладі реалізації проекту програмної системи – клієнт-серверного додатку (ролева гра).*

***Тема 2. Моделювання програмних систем. Візуальні моделі для уточнення користувацьких вимог.***

*Модель програмної системи у вигляді візуального змісту та HIPO-діаграми. Уточнення функціональних вимог на зустрічах з користувачем (ролева гра). Конкретизація моделі вимог.*

### **ЗМІСТОВИЙ МОДУЛЬ 2**

#### **Основи візуального моделювання програмних систем**

***Тема 3. Уніфікована мова моделювання UML та середовище моделювання Rational Rose. Діаграми для моделювання користувацьких вимог.***

*Уніфікована мова моделювання UML 2.0, система нотацій для моделювання програмних систем. Три групи діаграм моделювання програмних систем у UML 2.0.: діаграми поведінки, діаграми структур, діаграми взаємодії. Діаграма бізнес-прецедентів та прецедентів. Побудова графічного інтерфейсу користувача. Набір діаграм для роботи з замовником (користувачем).*

**Тема 4. Моделювання програмних систем у середовищі моделювання Rational Rose. Діаграми структур. Діаграми класів та об'єктів.**

Діаграми структур у UML 2.0. Система нотацій, яка використовується у діаграмах класів. Деталізація функцій підсистем при реалізації проекту програмної системи. Проведення аналізу діаграми класів на прикладі реалізації проекту програмної системи (ролева гра).

**Тема 5. Моделювання програмних систем у середовищі моделювання Rational Rose. Діаграми взаємодії.**

Діаграми взаємодії. Діаграми послідовностей та комунікації, система нотацій. Моделювання аспектів взаємодії потоків команд і даних у програмній системі. Аналіз та уточнення діаграм взаємодії на прикладі реалізації проекту програмної системи (ролева гра).

**Тема 6. Моделювання програмних систем у середовищі моделювання Rational Rose. Моделювання поведінки паралельних потоків у вигляді діаграми автомата.**

Ознайомлення з елементами середовища моделювання Rational Rose, що використовуються при побудові діаграм автоматів. Освоєння на прикладі реалізації проекту програмної системи побудови, аналізу та корегування діаграм станів при моделюванні функціональної взаємодії послідовних і паралельних потоків подій.

**Тема 7. Моделювання програмних систем у середовищі моделювання Rational Rose. Архітектура програмної системи.**

Відображення структурних аспектів програмної системи при реалізації розподілених програмних систем. Система нотацій, яка використовується у діаграмах розвертування (Deployment). Освоїти на прикладі реалізації проекту програмної системи побудову, аналіз та корегування діаграм розвертування.

## ЗМІСТОВИЙ МОДУЛЬ 3

### Інженерія вимог до програмного забезпечення

#### **Тема 8. Формування технічного завдання на розробку програмної системи.**

*Нормативні документи (ЄСПД), що регламентують склад проектної документації при розробці програмних проектів. Зміст та послідовність оформлення проектної документації. Технічне завдання, етапи та вимоги до його розробки. Техніко-економічне обґрунтування проекту. Створення технічного завдання на розроблюваний проект.*

#### **Тема 9. Тестові сценарії для перевірки програмної системи на відповідність вимогам. Технології повторного використання.**

*Розробка тестових сценаріїв для проміжного і підсумкового тестування програмних продуктів. Освоїти на прикладі реалізації проекту програмної системи розробку тестових сценаріїв (ролева гра). Застосування технологій повторного використання для підвищення ефективності програмних проектів..*

## 1.4. Самостійна робота студентів

**Мета та завдання самостійної роботи студентів.** Метою самостійної роботи студентів є набуття практичних навичок використанні уніфікованої мови моделювання UML та візуальних методів моделювання програмних систем, що застосовуються у сучасних технологій розробки програмних продуктів, використання середовища Rational Rose, Rational RequisitePro та SoDA для моделювання вимог у програмних проектах та підготовці специфікацій до тестування програмних продуктів.

**Форми організації самостійної роботи.** Самостійна робота організовується у вигляді реферативно-практичних та індивідуальних творчих завдань, які захищаються на підсумковому занятті.

## **1.4.1. Зміст самостійної роботи студентів**

### **ЗМІСТОВИЙ МОДУЛЬ 1**

#### **Визначення вимог до програмного забезпечення.**

***Тема 1. Класифікація програмних систем. Етапи розробки ПЗ. Роль вимог у плануванні програмних проектів.***

Стандарти і технології розробки програмних систем (MRP, CRP, MRPII, ERP, SCM, CRM, ERPPII, Workflow, OLAP, CALS, Project Management). Приклади формування вимог до складних програмних проектів.

***Тема 2. Вимоги до програмного забезпечення. Класифікація вимог. Користувацькі вимоги. Методи виявлення та опрацювання вимог у сучасних технологіях розробки ПЗ.***

Джерела вимог, та їх розширення у сучасних технологіях розробки програмних продуктів. Класифікація програмних систем за управлінням, особливості розробки. Аналіз вимог, дублювання та конфлікти у моделі вимог, матриця вимог. Бізнес-моделювання вимог, зв'язок з іншими етапами програмного проекту.

***Тема 3. Функціональні й нефункціональні вимоги, особливості реалізації у програмних проектах.***

Специфікація вимог до програмного забезпечення (Software Requirements Specification). Типи специфікацій та їх застосування. Програмні засоби для розробки та ведення специфікацій.

***Тема 4. Перевірка вимог. Змінення та модифікація вимог до програмного проекту. Моделі розробки програмного забезпечення.***

Перевірка вимог до програмного продукту, особливості проведення верифікації програмної системи. Моделі розробки програмного забезпечення. Методології сімейства IDEF. Методології розробки програмного забезпечення JAD та RAD. Методологія Agility. Особливості розробки ПЗ у методології Application Lifecycle Management (ALM).

***Тема 5. Вплив аналізу вимог по ПЗ на ітеративні цикли розробки проекту. Управління вимогами. Управління якістю. Управління рішеннями.***

Програмні продукти для управління вимогами та управління якістю розробки програмних систем. Бізнес-рішення та управління рішеннями. Стандарт COBIT. Формальні моделі рішень.

## **ЗМІСТОВИЙ МОДУЛЬ 2**

### **Основи візуального моделювання програмних систем**

***Тема 6. Технологія RUP. Уніфікована мова моделювання UML 2.0. Три групи діаграм моделювання програмних систем. Шаблони проектування. Програмне середовище Rational Rose.***

Варіації застосування технології RUP у сучасних технологіях розробки ПЗ. Документування вимог та проектних рішень на етапах ескізного та аналітичного проектування. Відмінність моделювання програмного проекту у UML 1.0 та UML 2.0: статичне та динамічне подання у UML 1.0.; три групи діаграм моделювання програмних систем у UML 2.0. Додаткові нотації мови UML для бізнес-моделювання. Розширення мови UML для побудови моделей бізнес-систем.

***Тема 7. Проектування, орієнтоване на користувача. Діаграми поведінки. Аналіз та коректування користувацьких вимог. Користувацькі інтерфейси.***

Проектування, орієнтоване на користувача. Узагальнена модель предметної області, її зв'язок з моделлю аналізу вимог до програмного забезпечення. Застосування бізнес-діаграм для уточнення вимог. Інтерфейси, їх реалізація за допомогою компонентів.

***Тема 8. Реалізація підсистем. Діаграми структур. Розробка архітектури програмної системи.***

Діаграми структур при моделюванні програмних систем: діаграма компонентів (Component), діаграма композитних структур (Composite structure), діаграма розвертування (Deployment), діаграма пакетів (Package). Нотації та ідеологія.



Розробка архітектури програмної системи: розподілені архітектури, «канали-та-фільтри» (pipe-and-filter), «модель-вигляд-контролер» (model-view-controller).

### ***Тема 9. Принципи розробки програмного забезпечення. Діаграми взаємодії.***

Загальні принципи розробки програмного забезпечення. Керування розгалуженими потоками управління у візуальних середовищах моделювання програмних систем. Створення моделей паралельних програм та їх аналіз. Моделювання паралельних програм на етапах проектування, реалізації та тестування. Створення бібліотек шаблонів у сучасних методологіях розробки програмних систем.

## **ЗМІСТОВИЙ МОДУЛЬ 3**

### **Інженерія вимог до програмного забезпечення**

#### ***Тема 10. Розробка вимог. Нормативи та стандарти, ЕСПД.***

Розробка вимог у програмних проектах з неперервною ідеологією постачання. Управління вимогами у проектах з неперервною ідеологією постачання. Специфікація вимог, формування технічного завдання у відповідності з ЕСПД.

#### ***Тема 11. Технічне завдання на програмний проект. Складові та їх зміст.***

Типові технічні завдання на реалізацію програмних проектів. Технічне завдання, глосарій та однозначність користувацьких вимог.

#### ***Тема 12. Верифікація та тестування програмного забезпечення. Розробка програмного забезпечення через тестування. Впровадження програмної системи.***

Валідація та тестування програмних продуктів. Розробка програмного забезпечення через тестування (test-first programming). Особливості впровадження розподілених програмних систем.

## 1.4.2. Індивідуальні завдання

**Індивідуальні завдання** виконуються у вигляді рефератів, творчих завдань та реферативно-практичних робіт, які стосуються практичного застосування новітніх методологій виявлення та розробки вимог до програмного проекту, моделювання та проектування програмних систем, аналізу вимог та управління процесом проектування програмних систем. Реферати та реферативно-практичні роботи складаються з двох частин: теоретичного огляду за темою роботи та практичного виконання завдання, яке ілюструє теоретичні положення. В якості пропонованих програмних продуктів можуть використовуватися Rational Rose чи Rational Software Architect, Rational RequisitePro, SoDA та інші програмні засоби. Творчі завдання спрямовані на формування вимог та створення мініпроєкту програмної системи, яка розробляється за обраною методологією, розробка користувацького інтерфейсу, який задовольняє вимогам замовника. За результатами роботи оформляється звіт, який складається з двох частин: опис ідеї та границь програмного проєкту, реалізація програмного проєкту у вигляді набору діаграм UML чи інших візуальних засобів проектування програмних проєктів. Завдання на індивідуальну роботу студент отримує у викладача, що проводить лабораторні завдання. Опрацьовані матеріали та створені проєкти можуть використовуватися у курсових роботах, кваліфікаційних та магістерських студентів на наступних курсах навчання.

**Методичні вказівки та завдання** до виконання творчих та реферативно-практичних робіт містять вимоги до виконання робіт, структуру, зміст, перелік літератури та Internet-посилання на електронні ресурси [ММ 3].

### **Типові завдання для індивідуальних робіт студентів:**

1. Провести аналіз користувацьких вимог для розробки автоматизованої інформаційної системи дистанційної освіти. Провести формалізацію вимог, скласти специфікацію вимог.
2. Розробити вимоги до тестування програмної системи автоматизації документообігу факультету. Скласти специфікацію вимог до тестування.
3. Скласти технічне завдання на розробку програмного модуля адміністрування автоматизованої системи продажу авіаквитків.
4. Розробити модель програмної системи фінансово-економічної звітності малого підприємства «Черевички».
5. Розробити технічні вимоги до програмної системи Internet-провайдера міста (до 10000 користувачів).

## 2. Форми та методи контролю знань, критерії оцінювання

Контроль знань, умінь і навичок студентів здійснюється з метою встановлення рівня засвоєння студентами теоретичного матеріалу та оволодіння практичними навичками, що передбачені навчальною програмою дисципліни «Аналіз вимог до програмного забезпечення».

Контроль знань студентів проводиться в таких формах:

- 1) поточний контроль (на лекціях, на лабораторних заняттях при захисті лабораторних робіт, рефератів, індивідуальних творчих завдань)
- 2) модульний контроль (при виконанні контрольних робіт),
- 3) підсумковий контроль (іспит),
- 4) ректорський контроль (комплексна контрольна робота).

### РОЗПОДІЛ БАЛІВ, ПРИСВОЮВАНИХ СТУДЕНТАМ ЗА СЕМЕСТР

№	Об'єкт контролю	Кількість на семестр	Максимальна кількість балів за одиницю об'єкту контролю	Сума балів	Примітка
1.	Захист лабораторної роботи	8	9	72	Студент отримує додатково бали за творчий підхід до виконання завдання, або за виконання додаткового завдання
2.	Модульна контрольна робота	1	28	28	
3.	Індивідуальне завдання (реферат, реферативно-практична робота, індивідуальне творче завдання)	1	12	12	
4.	Максимальна кількість балів за обов'язкові види поточного контролю			100	

Підсумковий контроль, що проводиться наприкінці шостого семестру і передбачає проведення іспиту. На іспиті проводиться контроль теоретичних знань, практичних навичок та компетенцій, що оцінюється сумарно до 50 балів. Інші від 0 до 50 балів студент набирає під час навчання: у шостому семестрі – 50 балів (100 балів × 0,5). Підсумкова оцінка виставляється за шкалою оцінювання у сумі до 100 балів.

### ШКАЛА ОЦІНЮВАННЯ ЗНАНЬ

Рейтинговий показник	Оцінка у національній шкалі	Оцінка ECTS
90 – 100	Зараховано	A (відмінно)
82 – 89		B (добре)
75 – 81		C (добре)
68 – 74		D (задовільно)
60 – 67		E (задовільно)
35 – 59	Незараховано	FX (незадовільно) з можливістю повторного складання
1 – 34		F (незадовільно) з обов'язковим повторним вивченням

**Методи навчання:** лекції, на яких застосовується проблемно-інформаційний метод навчання, навчання на прикладах і прототипах; лабораторні заняття, на яких застосовуються активні методи навчання (дослідницько-пошуковий метод, особистісно-орієнтовані методики та ін.).

**Методи оцінювання:** поточне опитування, опитування при захисті лабораторних робіт, тестування, захист індивідуальних творчих завдань, модульні контрольні роботи.

#### Методичне забезпечення:

1. Персональний комп'ютер.
2. Операційна система Windows XP і подальші версії.
3. Системи візуального моделювання Rational Rose та Rational Software Architect, система управління вимогами Rational RequisitePro, система автоматизованого документування і звітності SoDA.

## ІНФОРМАЦІЙНІ ДЖЕРЕЛА

### Основна література (О)

1. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам. – М.: БИНОМ, Лаборатория знаний, 2008. – 200 с.
2. Мацяшек Л.А. Анализ и проектирование информационных систем с помощью UML 2.0, 3-е изд. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2008. – 816 с. : ил.
3. Карл И. Вигерс. Разработка требований к программному обеспечению. — Русская редакция, 2004. – 576 с. : ил.
4. Ларман Крег. Применение UML и шаблонов проектирования. 2-е издание. / Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 624 с.
5. Бабич А.В. UML. Первое знакомство. – М.: БИНОМ, Лаборатория знаний, 2008. – 176 с.
6. Рамбо Дж., Якобсон А., Буч Г. UML: специальный справочник. – СПб.: Питер, 2002. – 656 с.
7. Леоненков А.В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. – М.: БИНОМ, Лаборатория знаний, 2006. – 320 с.
8. Котляров В.П. Основы тестирования программного обеспечения. – М.: БИНОМ, Лаборатория знаний, 2006. – 360 с.
9. Карпов Ю. Г. Model Chekcking. Верификация параллельных и распределённых программных систем. – СПб.: БХВ-Петербург, 2010. – 560 с.
10. Рекомендации по преподаванию программной инженерии в университетах. Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering Computing Curricula 2001: Computer Science, перевод с английского. М.: Интернет-Университет Информационных Технологий, 2007. – 472 с.

### Додаткова література (Д)

1. Сухомлин В.А. Введение в анализ информационных технологий: Курс лекций. Серия: Высшая компьютерная школа МГУ. – М.: Изд-во Горячая линия – Телеком, 2003. – 432 с.
2. Сеницын С.В., Налютин Н.Ю. Верификация программного обеспечения. – М.: БИНОМ, Лаборатория знаний, 2008. – 368 с.
3. Бирюков А.Н. Лекции о процессах управления информационными технологиями. – М.: БИНОМ, Лаборатория знаний, 2010. – 216 с.
4. Сперанский Д.В. Лекции по теории экспериментов с конечными автоматами. – М.: БИНОМ, Лаборатория знаний, 2010. – 287 с.
5. Добров Б.В., Иванов В.В., Лукашевич Н.В., Соловьев В.Д. Онтологии и тезаурусы: модели, инструменты, приложения. – М.: БИНОМ, Лаборатория знаний, 2010. – 176 с.
6. Галатенко В.А. Программирование в стандарте POSIX. Часть 1. – М.: БИНОМ, Лаборатория знаний, 2006. – 560 с.
7. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. – М.: БИНОМ, Лаборатория знаний, 2008. – 304 с.

8. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Управление внедрением информационных систем. – М.: БИНОМ, Лаборатория знаний, 2008. – 224с.
9. Кулямин В.В. Технологии программирования. Компонентный подход. – М.: БИНОМ, Лаборатория знаний, 2006. – 464 с.
10. Терри Кватрани. Rational Rose и UML. Визуальное моделирование. Пер. с англ. – М.: ДМК Пресс, 2001. – 176 с.

### **Перелік методичних матеріалів (ММ)**

1. Электронный конспект лекцій «Аналіз вимог до програмного забезпечення» / Укладач – к.т.н, доц. Супруненко О.О. Черкаси: Рукопис, 2010. – 54 с.
2. Методичні вказівки та завдання до лабораторних робіт з курсу „Аналіз вимог до програмного забезпечення” для студентів денної форми навчання спеціальності 6.05010301 „Програмне забезпечення систем”, Частина 1, 2. / Укладач – к.т.н, доц. Супруненко О.О. Черкаси: Рукопис, 2011. – 68 с.
3. Методичні вказівки та індивідуальні завдання до самостійної роботи студентів з дисципліни «Аналіз вимог до програмного забезпечення» для студентів, які навчаються за напрямом підготовки 050103 – «Програмна інженерія» усіх форм навчання. / Укладач – к.т.н, доц. Супруненко О.О. Черкаси: Рукопис, 2012. – 22 с.

### **Internet – посилання**

1. Академик. Rational Software [Электронный документ]. Режим доступа: <http://dic.academic.ru/dic.nsf/ruwiki/112007>. Проверено 24.03.2012.
2. Статьи и материалы по Rational – одному из брендов IBM Software Group. [Электронный документ]. Режим доступа: [http://www.ibm.com/developer-works/ru/rational/?S\\_TACT=105AGX99&S\\_CMP=GR01](http://www.ibm.com/developer-works/ru/rational/?S_TACT=105AGX99&S_CMP=GR01). Проверено 30.03.12.
3. Требования к программному обеспечению [Электронный ресурс]. Режим доступа: [http://ru.wikipedia.org/wiki/Требования\\_к\\_программному\\_обеспечению](http://ru.wikipedia.org/wiki/Требования_к_программному_обеспечению). Проверено: 04.04.2012.
4. Леоненков А.В. Визуальное моделирование в среде IBM Rational Rose 2003 [Электронный документ]. Режим доступа: <http://www.intuit.ru/department/se/ibmrrrose/>. Проверено 03.04.2012.
5. Марчуков А.В., Савельев А.О. Работа в Microsoft Visual Studio. [Электронный документ]. Режим доступа: <http://www.intuit.ru/department/itmngt/workinmsvistudio/>. Проверено 03.04.2012.
6. Model Maker 7. Available at: [www.modelmakertools.com](http://www.modelmakertools.com) (accessed 1 July 2012).
7. Together 2005. Available at: [www.borland.com](http://www.borland.com) (accessed 1 July 2012).
8. Леоненков А.В. Паттерны проектирования. [Электронный документ]. Режим доступа: <http://www.intuit.ru/department/pl/umlbasics/14/>. Проверено 03.04.12.

## Зміст навчальної програми

Структура програми навчальної дисципліни „Аналіз вимог до ПЗ" .....	3
ВСТУП.....	4
1.1. Тематика та зміст дисципліни.....	6
1.2. Структура залікового кредиту навчальної дисципліни.....	10
1.3. Теми лабораторних занять.....	12
1.4. Самостійна робота студентів .....	14
1.4.1. Зміст самостійної роботи студентів.....	15
1.4.2. Індивідуальні завдання .....	18
2. Форми та методи контролю знань, критерії оцінювання.....	19
Інформаційні джерела .....	21
Основна література (О) .....	21
Додаткова література (Д).....	21
Перелік методичних матеріалів (ММ) .....	22
Internet – посилання.....	22

Навчальне видання

**Супруненко Оксана Олександрівна**

# **Аналіз вимог до програмного забезпечення**

**Навчальна програма**

*Комп'ютерна верстка О.О. Супруненко*

Підписано до друку 21.12.2012. Формат 60×84/16. Гарнітура Times  
Папір офсет. Ум. друк. арк. 0,85. Тираж 50 прим. Зам. № 3 \_\_\_\_.

Видавець і виготовник – видавничий відділ  
Черкаського національного університету  
Імені Богдана Хмельницького.

Адреса: 18000, м. Черкаси, бул. Шевченка, 81, кімн. 117,  
тел. (0472) 37-13-16, факс (0472) 37-22-33,  
e-mail: [vydav@cdu.edu.ua](mailto:vydav@cdu.edu.ua), <http://www.cdu.edu.ua>  
Свідоцтво про внесення до державного реєстру  
суб'єкт видавничої справи ДК № 3427 від 17.03.2009 р.