

Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Спеціальність **121 Інженерія програмного забезпечення**

Освітня програма «**Інженерія програмного забезпечення**»

Рівень вищої освіти **перший (бакалаврський)**

Факультет **математики та інформатики**

2024–2025 навчальний рік

Робоча програма навчальної дисципліни «Аналіз вимог до програмного забезпечення» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньою програмою «Інженерія програмного забезпечення» для спеціальності 121 Інженерія програмного забезпечення

Мова навчання: українська

Розробник: Бичков О.С., професор кафедри інформаційних технологій та моделювання, докт. тех. наук, професор.

Робоча програма затверджена на засіданні кафедри інформаційних технологій та моделювання.

Протокол від 3 жовтня 2024 року № 9-А.

Завідувач кафедри



Мороз І. П.

Робочу програму схвалено навчально-методичною комісією факультету математики та інформатики.

Протокол від 4 жовтня 2024 року № 7-Б.

Голова навчально-методичної комісії



Білецький В. В.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна ф.н.	заочна ф.н.
Кількість кредитів: 5 Модулів: 1 Змістових модулів: 3 Індивідуальне науково-дослідне завдання: Загальна кількість годин : 150 Тижневих годин для денної форми навчання: аудиторних: 3 самостійної роботи студента: 6	Галузь знань: 12 Інформаційні технології Спеціальність: 121 Інженерія програмного забезпечення Освітня програма: «Інженерія програмного забезпечення» Рівень вищої освіти: перший (бакалаврський)	Обов'язкова	
		Рік підготовки:	
		2-й	2-й
		Семестр:	
		4-й	4-й
		Лекції:	
		26 год.	8 год.
		Практичні, семінарські:	
		-	-
		Лабораторні:	
		24 год.	8 год.
		Самостійна робота:	
		100 год.	134 год.
Індивідуальні завдання:			
-	-		
Вид контролю:			
Екзамен	Екзамен		

Передумови для вивчення дисципліни: «Програмування», «Архітектура комп'ютера», «Об'єктно-орієнтоване програмування», «Основи інженерії програмного забезпечення»

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Аналіз вимог до програмного забезпечення» відноситься до обов'язкових компонентів професійної підготовки здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю 121 Інженерія програмного забезпечення. Робоча програма навчальної дисципліни складена у відповідності до освітньо-професійної програми «Інженерія програмного забезпечення» підготовки бакалаврів за названою спеціальністю.

Аналіз вимог до програмного забезпечення – один із важливих етапів процесу розробки програмного забезпечення. Він включає збирання, документування та аналіз вимог до програмного проекту для забезпечення його відповідності потребам зацікавлених сторін і користувачів.

Метою викладання навчальної дисципліни «Аналіз вимог до програмного забезпечення» є засвоєння теоретичних основ та набуття практичних умінь аналізу функціональних, експлуатаційних, системних та користувацьких вимог до програмного забезпечення в процесі його проектування і супроводження.

Основними **завданнями** дисципліни «Аналіз вимог до програмного забезпечення» є:

- формування системи теоретичних знань методології аналізу і управління вимогами до ПЗ, класифікацій та властивостей вимог, стандартів управління вимогами;
- формування комплексу практичних умінь і навичок формулювати вимоги до програмних продуктів, систематизувати та специфікувати їх в сфері сучасного програмного забезпечення, вміння користуватися різними джерелами;
- розвивання розуміння сутності вимог до ПЗ і процесу управління вимогами, умінь складати нотації аналізу і управління вимогами, роботи з замовниками та користувачами для вдосконалення вимогам;

- формування уміння виконувати розробку документів різноманітного призначення, використовувати діаграмну технологію аналізу вимог до ПЗ;
- формування і розвивання умінь виконувати аналіз варіантів використання ПЗ згідно з потребами користувачів, аналізувати вимоги в залежності від сфери застосування ПЗ;
- формування цілісної системи знань про моделювання аспектів досліджуваних програмних систем, типи моделей, про основні концепції уніфікованої мови програмування UML.

Згідно з освітньо-професійною програмою навчальна дисципліна «Аналіз вимог до програмного забезпечення» має забезпечити формування у здобувачів вищої освіти відповідних **компетентностей**.

Загальні компетентності

K02. Здатність застосовувати знання у практичних ситуаціях.

Фахові компетентності

K13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

K16. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

3. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті освоєння повного курсу навчальної дисципліни «Аналіз вимог до програмного забезпечення» у здобувачів вищої освіти формуються глибокі, міцні і системні знання, які передбачають вільне володіння понятійним апаратом, розуміння основних задач предмету, його мети та завдання, а також здатність до практичного застосування цих знань при реалізації прикладних застосувань. Згідно з освітньо-професійною програмою мають бути досягнуті наступні **програмні результати навчання**:

ПР02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.

ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.

ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР24. Вміти проводити розрахунок економічної ефективності програмних систем.

Здобувачі вищої освіти повинні

знати:

- основні вимоги до програмного забезпечення, типи вимог, методи та джерела їх отримання;
- основи інженерії вимог до програмного забезпечення, специфікації та документування вимог;
- 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;

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.

4. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. «Основи інженерії вимог до програмного забезпечення»

Тема 1. Вступ. Предмет та задачі дисципліни. Класифікація програмних систем. Етапи розробки програмного забезпечення. Найбільш розповсюджені стандарти і технології розробки програмних систем (MRPII, CRM, ERP та ін.). Методології структурного аналізу та проектування.

Тема 2. Виявлення вимог. Вимоги до програмного забезпечення. Джерела вимог. Стратегії виявлення вимог.

Тема 3. Вимоги до ПЗ в залежності від моделей його створення. Типи вимог до програмного забезпечення. Класифікація 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. Характеристики «хороших» вимог. Системні та користувацькі вимоги до ПЗ.

Тема 4. Функціональні й нефункціональні вимоги. Функціональні та експлуатаційні вимоги до ПЗ. Вплив на структуру та реалізацію програмного проекту. Відображення нефункціональних вимог у структурі програмного проекту. Рекомендації до структури й методів опису програмних вимог.

Тема 5. Документування вимог, специфікація вимог. Специфікація вимог до програмного проекту. Типи специфікацій та їх застосування. Програмні засоби для ведення специфікацій та управління вимогами.

Змістовий модуль 2. «Основи візуального моделювання програмних систем»

Тема 6. Технологія RUP. Уніфікована мова моделювання UML 2.0. Групи діаграм моделювання програмних систем. Технологія RUP. Уніфікована мова моделювання UML: основні концепції. Три групи діаграм моделювання програмних систем у UML 2.0. Типи класифікаторів та відношень. Шаблони проектування: абстрактнеконкретне (abstraction-occurrence), компоновщик (composite), гавець-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. 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.

Тема 8. Реалізація підсистем. Діаграми структур. 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.

Тема 9. Розробка архітектури програмної системи. Розробка архітектури програмної системи: розподілені архітектури, «канали-та-фільтри» (pipe-and-filter), «модель-вигляд-контролер» (model-viewcontroller).

Тема 10. 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 11. 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 12. Технічне завдання на програмний проект. Технічне завдання, етапи розробки та зміст. Етапи реалізації технічного завдання на стадіях ескізного, технічного та робочого проекту. Керування пріоритетами.

Тема 13. Верифікація та тестування програмного забезпечення. Верифікація та тестування програмного забезпечення. Керування якістю розробки програмних продуктів. Атрибути якості програмного забезпечення.

5. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		лек	пр	лаб	інд	с.р.		лек	пр	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Основи інженерії вимог до програмного забезпечення												
Тема 1. Вступ. Предмет та задачі дисципліни.	6	2				4	6	1				5
Тема 2. Виявлення вимог.	12	2		2		8	12			1		11
Тема 3. Вимоги до ПЗ в залежності від моделей його створення.	12	2		2		8	12	1		1		10
Тема 4. Функціональні й нефункціональні вимоги.	12	2		2		8	12			1		11
Тема 5. Документування вимог, специфікація вимог.	12	2		2		8	12	1				11
Разом за змістовим модулем 1	54	10		8		36	54	3		3		48
Змістовий модуль 2. Основи візуального моделювання програмних систем												
Тема 6. Технологія RUP.	12	2		2		8	12	1				11

Уніфікована мова моделювання UML 2.0. Групи діаграм моделювання програмних систем.												
Тема 7. Проектування, орієнтоване на користувача. Діаграми поведінки. Аналіз та коректування користувацьких вимог.	12	2		2		8	12			1		11
Тема 8. Реалізація підсистем. Діаграми структур.	12	2		2		8	12	1		1		10
Тема 9. Розробка архітектури програмної системи.	12	2		2		8	12			1		11
Тема 10. Принципи розробки програмного забезпечення. Діаграми взаємодії.	12	2		2		8	12	1				11
Разом за змістовим модулем 2	60	10		10		40	60	3		3		54
Змістовий модуль 3. Інженерія вимог до програмного забезпечення												
Тема 11. Розробка вимог. Нормативи та стандарти. ЕСПД.	12	2		2		8	12	1				11
Тема 12. Технічне завдання на програмний проект.	12	2		2		8	12			1		11
Тема 13. Верифікація та тестування програмного забезпечення.	12	2		2		8	12	1		1		10
Разом за змістовим модулем 3	36	6		6		24	36	2		2		32
Усього годин	150	26	-	24	-	100	120	8	-	8	-	134

6. ТЕМИ СЕМІНАРСЬКИХ ЗАНЯТЬ

Не передбачено навчальним планом

7. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено навчальним планом

8. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин
1.	Методології розробки програмного забезпечення	2
2.	Моделювання програмних систем. Візуальні моделі для уточнення користувацьких вимог.	2
3.	Уніфікована мова моделювання UML. Середовище моделювання.	2
4.	Діаграми структур. Діаграми класів та об'єктів.	4
5.	Діаграми взаємодії.	4
6.	Моделювання поведінки паралельних потоків.	4
7.	Моделювання програмних систем. Архітектура програмної системи.	4
8.	Формування технічного завдання на розробку програмної системи.	2
	Разом	24

9. САМОСТІЙНА РОБОТА

№ з/п	Назва теми	Кількість годин
1.	Поняття «програмна система» і «програмний продукт». Поняття предметної області. Цілі аналізу предметної області. Особливості структурного і об'єктного аналізу.	5
2.	Поняття «вимога». Класифікація вимог. Вимоги до продукту і до процесу. Функціональні вимоги до ПЗ. Експлуатаційні вимоги до ПЗ.	5
3.	Системні та користувацькі вимоги до ПЗ. Особливості вимог до ПЗ для користувачів і для розробників. Основні властивості вимог та взаємозв'язок властивостей. Джерела виникнення вимог.	5
4.	Документування вимог, специфікація вимог. Правила документування вимог, призначених для користувача і для розробників. Категорії системних вимог, що мають бути описані в специфікації.	5
5.	Джерела, стратегії та методи виявлення вимог до ПЗ. Процес розробки вимог до ПЗ, формування та аналіз.	5
6.	Концепції, принципи, методи системного моделювання для аналізу вимог до ПЗ. Особливості та характеристики системних моделей. Типи моделей для уточнення вимог, призначених для користувача і для розробників. Цілі та засоби функціонального і об'єктного моделювання.	5
7.	Розширений аналіз вимог, застосування комп'ютерних засобів. Методика прототипування (експериментального і еволюційного), пов'язана з видом прототипу.	5
8.	Призначення мови UML та її загальна структура.	4
9.	Пакети UML.	4
10.	Базові семантичні конструкції мови, їх опис за допомогою спеціальних позначень.	4
11.	Особливості графічного зображення діаграм мови UML.	4
12.	Елементи графічної нотації діаграми варіантів використання.	4
13.	Формалізація функціональних вимог до системи за допомогою діаграми варіантів використання.	4
14.	Розширення мови UML для побудови моделей програмного забезпечення та бізнес-систем.	4
15.	Відношення та їх графічне зображення на діаграмі класів.	4
16.	Внутрішні дії стану і діяльність.	4
17.	Події та їх специфікація на діаграмах станів.	4
18.	Особливості моделювання поведінки об'єктів у вигляді діаграм станів.	5
19.	Особливості моделювання паралельного поведінки об'єктів у формі діаграм станів.	5
20.	Особливості фізичного представлення програмних систем.	5
21.	Використання діаграми компонентів для проектування залежностей між компонентами	5
22.	Верифікація та валідація вимог до ПЗ. Повнота і узгодженість вимог. Якість специфікацій та їх характеристики. Способи перевірки якості специфікацій. Експертиза специфікації. Тестування вимог та атестації вимог. Управління вимогами. Основні функції управління вимогами.	5
	Разом	100

Самостійна робота є основним засобом засвоєння здобувачем вищої освіти навчального матеріалу у час, вільний від обов'язкових навчальних занять.

Самостійна робота здобувача вищої освіти над засвоєнням навчального матеріалу з навчальної дисципліни може виконуватися у бібліотеці, навчальних кабінетах, лабораторіях і комп'ютерних класах, а також у домашніх умовах та передбачає:

- вивчення лекційного матеріалу по темі;
- опрацювання літератури по темі;
- підготовка до лабораторних робіт;
- підготовка до самостійної роботи;
- робота за персональним комп'ютером по темі;
- робота в глобальній мережі Інтернет.

10. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Не передбачено

11. МЕТОДИ НАВЧАННЯ

МН1 – словесний метод (лекція, дискусія, обговорення досліджуваного явища чи процесу, аналіз проблемних ситуацій);

МН2 – практичний метод (лабораторні заняття);

МН3 – наочний метод (ілюстрації, демонстрації);

МН4 – робота з навчально-методичною літературою (конспектування, складання реферату);

МН5 – інтерактивний метод (із застосуванням аудіо, відео, новітніх інформаційних технологій та комп'ютерних засобів навчання);

МН6 – самостійна робота (самостійний аналіз, проектування та програмна реалізація завдань).

12. МЕТОДИ ОЦІНЮВАННЯ

МО1 – екзамен;

МО2 – усне або письмове опитування під час лабораторних занять;

МО4 – тестування;

МО7 – презентація та обговорення результатів виконаних завдань;

МО8 – перевірка контрольних робіт;

МО9 – захист лабораторних робіт.

13. ЗАСОБИ ДІАГНОСТИКИ РЕЗУЛЬТАТІВ НАВЧАННЯ

- завдання та рекомендації для виконання лабораторних робіт;
- тести поточного контролю знань;
- тести підсумкового контролю знань;
- модульна контрольна робота;
- комплексна контрольна робота;
- презентація (звіти) результатів виконання завдань;
- підсумковий контроль у формі екзамену.

Види та методи навчання і оцінювання

Код компетентності (за ОПШ)	Назва компетентності	Код ПРН	Назва програмного результату навчання	Методи навчання	Методи оцінювання
K02	Здатність застосовувати	ПР02	Знати кодекс професійної етики, розуміти соціальну значимість та	МН1, МН2, МН3, МН4,	МО1, МО2, МО4, МО7,

	знання у практичних ситуаціях.		культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	МН6	МО8, МО9
		ПР09	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР14	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР24	Вміти проводити розрахунок економічної ефективності програмних систем.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
K13	Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.	ПР09	Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР10	Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проєктування.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
K14	Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.	ПР11	Вибирати вихідні дані для проєктування, керуючись формальними методами опису вимог та моделювання.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР14	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
K16	Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.	ПР04	Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР14	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
K17	Здатність дотримуватись специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів	ПР02	Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9
		ПР04	Знати і застосовувати професійні	МН1, МН2,	МО1, МО2,

	життєвого циклу.		стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.	МН3, МН4, МН6	МО4, МО7, МО8, МО9
K25	Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.	ПР14	Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	МН1, МН2, МН3, МН4, МН6	МО1, МО2, МО4, МО7, МО8, МО9

14. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Результат освітньої діяльності здобувача вищої освіти оцінюється згідно Положення про оцінювання знань і умінь здобувачів вищої освіти РДГУ за такими критеріями оцінювання та рівнями компетентності:

Суми балів за 100-бальною шкалою	Оцінка в ЄКТС	Значення оцінки ЄКТС	Критерії оцінювання	Рівень компетентності	Оцінка за національною шкалою
90-100	A	відмінно	здобувач вищої освіти виявляє особливі творчі здібності, вміє самостійно здобувати знання, без допомоги викладача знаходить і опрацьовує необхідну інформацію, вміє використовувати набуті знання і вміння для прийняття рішень у нестандартних ситуаціях, переконливо аргументує відповіді, самостійно розкриває власні здібності	високий (творчий)	відмінно
82-89	B	добре	здобувач вищої освіти вільно володіє теоретичним матеріалом, застосовує його на практиці, вільно розв'язує вправи і задачі у стандартних ситуаціях, самостійно виправляє допущені помилки, кількість яких незначна	достатній (конструктивно-варіативний)	добре
74-81	C	добре	здобувач вищої освіти вміє зіставляти, узагальнювати, систематизувати інформацію під керівництвом викладача, загалом самостійно застосовувати її на практиці; контролювати власну діяльність; виправляти помилки, з-поміж яких є суттєві, добирати аргументи для підтвердження думок		
64-73	D	задовільно	здобувач вищої освіти	середній	задовільно

			відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, за допомогою викладача може аналізувати навчальний матеріал, виправляти помилки, з-поміж яких є значна кількість суттєвих	(репродуктивний)	
60-63	E	задовільно	здобувач вищої освіти володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні		
35-59	FX	незадовільно з можливістю повторного складання семестрового контролю	здобувач вищої освіти володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу	низький (рецептивно-продуктивний)	незадовільно
1-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	здобувач вищої освіти володіє матеріалом на рівні елементарного розпізнавання і відтворення окремих фактів, елементів, об'єктів	низький (рецептивно-продуктивний)	незадовільно

Підсумкова (загальна) оцінка з навчальної дисципліни є сумою рейтингових оцінок (балів), одержаних за окремі оцінювальні форми навчальної діяльності: поточне і підсумкове оцінювання рівня засвоєння теоретичного та практичного матеріалу під час аудиторних занять і самостійної роботи; оцінка (бали) за виконання лабораторних завдань; оцінка (бали) за індивідуальну науково-дослідну роботу; оцінка (бали) за участь у наукових конференціях, олімпіадах, підготовку наукових публікацій, рефератів тощо.

15. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ ВИЩОЇ ОСВІТИ

В університеті діє накопичувальна кредитно-трансферна система оцінювання програмних результатів навчання студентів, що реалізується в ході виконання і захисту лабораторних робіт та модульного контролю, для яких визначено мінімальну кількість балів, яку слід набрати для формування рейтингового балу студента та виставлення його у залікову книжку і відомість успішності студентів з відповідними оцінками за національною та європейською кредитно-трансферною системами (ЄКТС).

Поточне тестування та самостійна робота													Екзамен	Сума
Змістовий модуль №1					Змістовий модуль №2					Змістовий модуль №3				
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13		
2	4	4	4	4	4	4	4	4	4	4	4	4		
Модульн. контр. – 5					Модульн. контр. – 5									
23					25					12			40	100

T1, T2, ..., T13 – теми змістових модулів.

16. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- рекомендована література для вивчення дисципліни;
- інструкції до виконання лабораторних робіт;

- дистанційний курс «Аналіз вимог до програмного забезпечення» на сервісі дистанційної освіти Рівненського державного гуманітарного університету;
- відкриті дистанційні курси на платформах Prometheus, Coursera;
- ОС Windows XP/7/8/10;
- MS Office;

17. ПИТАННЯ ДЛЯ ПІДГОТОВКИ ДО ПІДСУМКОВОГО КОНТРОЛЮ

1. Визначення поняття вимоги. Класифікації вимог.
2. Класифікація за масштабом.
3. Класифікація з архітектури.
4. Класифікація за характером використання інформації.
5. Класифікація за системою подання даних.
6. Класифікація за підтримуваним стандартом управління і технологіям комунікації.
7. Класифікація за ступенем автоматизації.
8. Рівні вимог. Повнота вимог.
9. Ясність вимог (недвозначність, визначеність, однозначність специфікацій).
10. Коректність і узгодженість вимог (несуперечність).
11. Верифікація вимог (придатність до перевірки).
12. Необхідність і корисність при експлуатації вимог.
13. Впорядкованість за важливістю і стабільністю вимог.
14. Хто створює і використовує вимоги. Організація роботи з вимогами.
15. Аналіз вимог, бізнес-аналіз, аналіз проблемної області.
16. Роль глосарію при аналізі вимог.
17. Аналіз вимог та інші робочі потоки програмної інженерії.
18. Джерела вимог. Стратегії виявлення вимог.
19. Специфікація варіанту використання. Вільний формат.
20. Специфікація варіанту використання. Шаблон повного опису.
21. Специфікація варіанту використання за А. Коберн.
22. Специфікація варіанту використання. Табличні представлення варіанту використання.
23. Специфікація функціональних вимог. Атрибути вимог.
24. Цілі прототипування.
25. Класифікація прототипів, горизонтальний прототип.
26. Класифікація прототипів, вертикальний прототип.
27. Класифікація прототипів, одноразовий прототип.
28. Класифікація прототипів, еволюційний прототип.
29. Класифікація прототипів, паперовий прототип.
30. Класифікація прототипів, розкадровка.
31. Ілюстровані сценарії прецедентів.
32. Документування вимог.
33. Вимоги до зовнішнього інтерфейсу. Інтерфейси користувача. Інтерфейси обладнання.
34. Вимоги до зовнішнього інтерфейсу. Інтерфейси ПЗ. Інтерфейси передачі інформації.
35. Методи і засоби перевірки вимог. Неофіційні перегляди вимог.
36. Методи і засоби перевірки вимог. Інспекції.
37. Методи і засоби перевірки вимог. Розробка тестів.
38. Методи і засоби перевірки вимог. Визначення критеріїв прийнятності.
39. Двозначність вимог.
40. Мінімальна специфікація.
41. Технологія RUP.
42. Уніфікована мова моделювання UML: основні концепції.
43. Три групи діаграм моделювання програмних систем у UML 2.0.
44. Типи класифікаторів та відношень.

45. Шаблони проектування: абстрактнеконкретне (abstraction-occurrence), компоновщик (composite), гравець-роль (player-role), одинак (singleton), спостерігач (observer), делегування (delegation), фасад (facade), адаптер (adapter).
46. Проектування, орієнтоване на користувача.
47. Модель аналізу вимог до програмного забезпечення.
48. Діаграми поведінки при представленні моделей програмної системи: діаграма прецедентів (Use Case), діаграма діяльності (Activity), діаграма автомата (State Machine).
49. Нотації та ідеологія. Діаграми бізнес-прецедентів та прецедентів, побудова, аналіз та деталізація.
50. Мова обмежень OCL. Узгодження користувацьких вимог.
51. 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).
52. Аналітична та програмна моделі програмної системи.
53. Розробка діаграм бізнес-класів та класів.
54. Типи зв'язків у діаграмі класів.
55. Визначення операцій у класах.
56. Документування операцій.
57. Створення атрибутів та методів.
58. Діаграма об'єктів.
59. Деталізація при розробці моделі ПЗ.
60. Розробка архітектури програмної системи: розподілені архітектури, «канали-та-фільтри» (pipe-and-filter), «модель-вигляд-контролер» (model-viewcontroller).
61. 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).
62. Діаграми взаємодії при моделюванні програмних систем: діаграми послідовностей (Sequence) та комунікації (Communication), діаграма огляду взаємодії (Interaction Overview) та синхронізації (Timing).

18. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Козак О.Л. Опорний конспект лекцій з курсу «Аналіз вимог до програмного забезпечення» для студентів напрямку підготовки Програмна інженерія. Тернопіль, 2011. 56 с.
2. Лавріщева К.М. Програмна інженерія. Київ, 2008. 319 с.
3. Смірнов В.В., Смірнова Н.В. Технологія проектування програмних систем. Лекції. Кіровоград : КНТУ, 2012. 95 с.

Допоміжна:

4. Unified Modeling Language User Guide. The (2 ed.). Addison-Wesley, 2005. 496 p.
5. Ambler S. The Object Primer: Agile Model Driven Development with UML 2. Cambridge University Press, 2004.
6. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley, 2004.
7. Alexander I., Beus-Duekic L. Discovering Requirements: How to Specify Products and Services. Wiley, 2009.
8. Wiegers K.E. Software Requirements, 2nd ed. Microsoft Press, 2003.
9. INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.2, International Council on Systems Engineering, 2012.
10. Van Lamsweerde A. Requirements Engineering: From System Goals to UML Models to

19. ІНФОРМАЦІЙНІ (ІНТЕРНЕТ) РЕСУРСИ

- UML Tutorial. – Режим доступу: <https://www.tutorialspoint.com/uml/index.htm>.
- <https://www.uml.org/>
- <http://www.undef.com/IDEF3.htm> – Process Description Capture Method
- <http://www.itl.nist.gov/fipspubs/idef1x.doc> – IDEF1X
- <https://www.microsoft.com/Rus/Download.aspx>
- <https://mon.gov.ua/ua> – сайт Міністерства освіти і науки України
- <http://naps.gov.ua/> – Національна академія педагогічних наук
- <http://nbuv.gov.ua> – Національна бібліотека України імені В.І.Вернадського
- <http://iteach.com.ua> – сайт проекту Intel® «Навчання для майбутнього»
- <http://library.rshu.edu.ua/> – сайт наукової бібліотеки РДГУ