

Рівненський державний гуманітарний університет
Кафедра інформаційних технологій та моделювання

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОСНОВИ ІНЖЕНЕРІЇ 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 «Основи інженерії програмного забезпечення» для здобувачів першого (бакалаврського) рівня вищої освіти зі спеціальності 121 Інженерія програмного забезпечення за освітньою програмою «Інженерія програмного забезпечення»

Мова навчання: українська

Розробник: Петренко С. В., доцент кафедри інформаційних технологій та моделювання,
канд. пед. наук, доцент

Робоча програма затверджена на засіданні кафедри інформаційних технологій та моделювання.

Протокол від 27 серпня 2024 року № 8.

Завідувач кафедри



Мороз І. П.

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

Протокол від 3 вересня 2024 року № 7.

Голова навчально-методичної комісії



Гнедко Н. М.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
		Обов'язкова	Обов'язкова
Кількість кредитів:	5	Рік підготовки:	
Модулів:		2	2
Змістових модулів:	2	Семестр:	
Індивідуальне науково-дослідне завдання:		3	3
Загальна кількість годин:	150	Лекції:	
		26 год.	8 год.
Тижневих годин:	3	Практичні:	
		-	-
Аудиторних самостійної роботи студента	6	Лабораторні:	
		24 год.	8 год.
		Самостійна робота:	
		100	134
		Індивідуальні завдання:	
		-	-
		Вид контролю:	
		екзамен	екзамен

Передумови для вивчення дисципліни: «Іноземна мова (за професійним спрямуванням)», «Програмування», «Об'єктно-орієнтоване програмування».

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Навчальна дисципліна «Основи інженерії 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.

Мета дисципліни полягає у ознайомленні здобувачів вищої освіти з основними поняттями, засобами та методами інженерії програмного забезпечення, а також формування у знань основних принципів розробки ефективного програмного забезпечення та набуття навичок використання основних принципів реалізації етапів життєвого циклу ПЗ.

У результаті освоєння повного курсу навчальної дисципліни «Основи інженерії програмного забезпечення» у здобувачів вищої освіти мають сформуватися визначені освітньою програмою **компетентності**.

Загальні компетентності

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

Фахові компетентності

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

K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

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

3. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

У результаті освоєння повного курсу навчальної дисципліни «Основи інженерії 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.

ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПР08. Вміти розробляти людино-машинний інтерфейс.

ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

Здобувачі вищої освіти мають

знати:

- фундаментальні концепції, парадигми і основні принципи функціонування засобів інженерії програмного забезпечення;
- основний інструментарій для розробки ПЗ;
- принципи та моделі розробки ПЗ;
- методології розробки програмного забезпечення;
- архітектуру ПЗ;
- підходи в керуванні розробкою ПЗ;
- принципи і методи тестування ПЗ;

вміти:

- проводити комплексну розробку програмного забезпечення;
- формулювати вимоги до програмного продукту;
- розробляти структуру проекту програмного забезпечення;
- проводити декомпозицію комплексних задач;
- будувати структурні і функціональні діаграми;
- ефективно використовувати сучасний інструментарій та застосовувати кращі практики при розробці ПЗ;
- оцінювати, тестувати та забезпечувати якість ПЗ;
- керувати версіями програмного проекту;
- ефективно підтримувати та оновлювати вже існуючі продукти.

4. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. «Визначальні принципи розробки ПЗ»

Тема 1. Життєвий цикл та підходи до розробки програмного забезпечення. Методологія Agile та Waterfall. Методика Scrum: ролі, процеси та артефакти. Практика CI/CD. Системи відслідковування помилок.

Тема 2. Системи контролю версій. Специфіка функціонування SCM. Основні принципи роботи та команди. Інструментарій та екосистема. Кращі практики використання.

Тема 3. Принципи розробки ПЗ. Принципи SOLID, DRY, YAGNI, KISS, MVP.

Тема 4. Моделювання систем. Мова моделювання та їх види. Роль моделювання в розробці ПЗ та інженерії систем. Графічна мова UML. Види UML-діаграм.

Тема 5. Клієнт-серверна архітектура. Роль клієнта та сервера. Основні компоненти клієнтської та серверної частини. Протоколи та комунікація. Масштабування та оптимізація.

Змістовий модуль 2. «Проектування та реалізація ПЗ»

Тема 6. Протокол HTTP/HTTPS. Архітектурний стиль REST. Основні принципи протоколу HTTP. Структура запиту та відповіді HTTP. Сесії та куки. Протокол HTTPS. Підтримка методів версіонування HTTP. Підхід до архітектури мережеских протоколів.

Тема 7. Проектування та реалізація ПЗ. Відмінності проектування з нуля та існуючого проекту. Збір вимог та визначення технічний стеку. Налаштування середовища. Інтеграція в діючий проект. Типи розробки.

Тема 8. Підтримка та покращення ПЗ. Підбір інструментарію. Конвенції написання коду. Командні правила. Аналіз існуючого продукту. Функціональні та нефункціональні вимоги. Покращення коду шляхом рефакторингу.

Тема 9. Тестування ПЗ. Процес тестування ПЗ. Етапи тестування. Методи і рівні тестування ПЗ. Тестувальник та його роль. QA, QC і тестування. Інструментарій для тестування.

Тема 10. Типи тестування програмного забезпечення. Класифікація тестування. Функціональне тестування. Тестування навантаження. Тестування безпеки.

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
Модуль 1												
Змістовий модуль 1. «Визначальні принципи розробки ПЗ»												
Тема 1. Життєвий цикл та підходи до розробки програмного забезпечення.	14	4		2		8	14	1				13
Тема 2. Системи контролю версій.	12	2		2		8	24	1		2		21
Тема 3. Парадигми розробки ПЗ.	12	2		2		8						
Тема 4. Моделювання систем.	12	2		2		8	24	1		2		21
Тема 5. Клієнт-серверна архітектура.	12	2		2		8						
Разом за змістовим модулем 1	62	12	-	10	-	40	62	3	-	4	-	55
Змістовий модуль 2. «Проектування та реалізація ПЗ»												

Тема 6. Протокол HTTP/HTTPS. Архітектурний стиль REST.	14	2		2		10	14	1				13
Тема 7. Проектування та реалізація ПЗ.	23	4		4		15	40	2		2		36
Тема 8. Підтримка та покращення ПЗ.	23	4		4		15						
Тема 9. Тестування ПЗ.	14	2		2		10	34	2		2		30
Тема 10. Типи тестування програмного забезпечення.	14	2		2		10						
Разом за змістовим модулем 2	88	14	-	14	-	60	88	5	-	4	-	79
Усього годин	150	26	-	24	-	100	150	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 год. (денна ф.н.)
1.	Життєвий цикл та підходи до розробки програмного забезпечення	2
2.	Системи контролю версій	2
3.	Парадигми розробки ПЗ	2
4.	Моделювання систем	2
5.	Клієнт-серверна архітектура	2
6.	Протокол HTTP/HTTPS. Архітектурний стиль REST	2
7.	Проектування та реалізація ПЗ	4
8.	Підтримка та покращення ПЗ	4
9.	Тестування ПЗ	2
10.	Типи тестування програмного забезпечення	2
	Разом	24

9. САМОСТІЙНА РОБОТА

№ з.п.	Назва теми, питання для опрацювання	Кількість год. (денна ф.н.)
1.	Життєвий цикл та підходи до розробки програмного забезпечення. Методологія Agile та Waterfall. Методика Scrum: ролі, процеси та артефакти. Практика CI/CD. Системи відслідковування помилок.	8

2.	Системи контролю версій. Специфіка функціонування SCM. Основні принципи роботи та команди. Інструментарій та екосистема. Кращі практики використання.	8
3.	Принципи розробки ПЗ. Принципи SOLID, DRY, YAGNI, KISS, MVP.	8
4.	Моделювання систем. Мова моделювання та їх види. Роль моделювання в розробці ПЗ та інженерії систем. Графічна мова UML. Види UML-діаграм.	8
5.	Клієнт-серверна архітектура. Роль клієнта та сервера. Основні компоненти клієнтської та серверної частини. Протоколи та комунікація. Масштабування та оптимізація.	8
6.	Протокол HTTP/HTTPS. Архітектурний стиль REST. Основні принципи протоколу HTTP. Структура запиту та відповіді HTTP. Сесії та куки. Протокол HTTPS. Підтримка методів версіонування HTTP. Підхід до архітектури мережевих протоколів.	10
7.	Проектування та реалізація ПЗ. Відмінності проектування з нуля та існуючого проєкту. Збір вимог та визначення технічний стеку. Налаштування середовища. Інтеграція в діючий проєкт. Типи розробки.	15
8.	Підтримка та покращення ПЗ. Підбір інструментарію. Конвенції написання коду. Командні правила. Аналіз існуючого продукту. Функціональні та нефункціональні вимоги. Покращення коду шляхом рефакторингу.	15
9.	Тестування ПЗ. Процес тестування ПЗ. Етапи тестування. Методи і рівні тестування ПЗ. Тестувальник та його роль. QA, QC і тестування. Інструментарій для тестування.	10
10.	Типи тестування програмного забезпечення. Класифікація тестування. Функціональне тестування. Тестування навантаження. Тестування безпеки.	10
	Разом	100

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 рефератів);
- 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;
- МО7 – презентації результатів виконаних завдань та досліджень;
- МО9 – захист лабораторних і практичних робіт.

13. ЗАСОБИ ДІАГНОСТИКИ РЕЗУЛЬТАТІВ НАВЧАННЯ

- екзамен;
- аналітичні звіти;
- презентації результатів виконаних завдань та досліджень;
- лабораторні роботи.

Види та методи навчання і оцінювання

Код компетентності (за ОПШ)	Назва компетентності	Код ПРН	Назва програмного результату навчання	Методи навчання	Методи оцінювання результатів навчання
K01	Здатність до абстрактного мислення, аналізу та синтезу	ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення	MН1 MН2 MН3 MН4 MН5 MН6	MO1 MO2 MO7 MO9
K02	Здатність застосовувати знання у практичних ситуаціях	ПР02	Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.	MН1 MН2 MН3 MН4 MН5 MН6	MO1 MO2 MO7 MO9
		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	MН1 MН2 MН3 MН6	MO1 MO2 MO7 MO9
		ПР18	Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.	MН1 MН2 MН3 MН4 MН5 MН6	MO1 MO2 MO7 MO9
K13	Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення	ПР08	Вміти розробляти людино-машинний інтерфейс.	MН1 MН2 MН3 MН6	MO1 MO7 MO9
K20	Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення	ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	MН1 MН2 MН3 MН4 MН5 MН6	MO1 MO7 MO9

K23	Здатність реалізувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.	ПР03	Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.	МН1 МН2 МН3 МН4 МН5 МН6	МО1 МО7 МО9
		ПР07	Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	МН1 МН2 МН3 МН4 МН5 МН6	МО1 МО7 МО9
K25	Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення	ПР18	Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.	МН1 МН2 МН3 МН4 МН5 МН6	МО1 МО2 МО7 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 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	середній (репродуктивний)	задовільно
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, виконання ІНДЗ 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						
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10		
4	9	9	9	4	12	7	10	10	6		

Модульний контроль – 5	Модульний контроль – 5		
40	50	10	100

T1, T2, ..., T10 – теми змістових модулів.

Розподіл балів за видами освітньої діяльності

№ з.п.	Вид навчальної діяльності	Оціночні бали	Кількість балів
T1	Виконання завдань лабораторного заняття	2	4
	Презентації результатів виконаних завдань	2	
T2	Виконання завдань лабораторного заняття	4	9
	Презентації результатів виконаних завдань	5	
T3	Виконання завдань лабораторного заняття	4	9
	Презентації результатів виконаних завдань	5	
T4	Виконання завдань лабораторного заняття	4	9
	Презентації результатів виконаних завдань	5	
T5	Виконання завдань лабораторного заняття	2	4
	Презентації результатів виконаних завдань	2	
Модульний контроль		5	5
T6	Виконання завдань лабораторного заняття	6	12
	Презентації результатів виконаних завдань	6	
T7	Виконання завдань лабораторного заняття	3	7
	Презентації результатів виконаних завдань	4	
T8	Виконання завдань лабораторного заняття	5	10
	Презентації результатів виконаних завдань	5	
T9	Виконання завдань лабораторного заняття	5	10
	Презентації результатів виконаних завдань	5	
T10	Виконання завдань лабораторного заняття	2	6
	Презентації результатів виконаних завдань	4	
Модульний контроль		5	5
Екзамен		10	10
Разом		100	

16. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Самостійна робота студентів над теоретичним та практичним матеріалом навчальної дисципліни здійснюється в таких формах:

- вивчення теоретичного матеріалу, що викладений на лекційних заняттях та призначеного для самостійного опрацювання;
- індивідуальне виконання навчальних завдань, розв'язування алгоритмічних задач та завдань по розробці алгоритмів та програмуванню.

В якості навчально-методичного забезпечення самостійної роботи студентів використовується основна та допоміжна література з дисципліни (підручники, навчальні посібники, монографії, словники, довідники, енциклопедії, журнали, статті у наукових виданнях), Інтернет-ресурси, матеріал лекцій, представлений у електронному вигляді та інтерактивних презентацій, методичні рекомендації для виконання лабораторних робіт.

17. ПИТАННЯ ДЛЯ ПІДГОТОВКИ ДО ПІДСУМКОВОГО КОНТРОЛЮ

1. SWEBOOK
2. Тестування програмного продукту
3. Життєвий цикл програмного забезпечення.
4. Методології розробки програмного забезпечення.
5. Agile.
6. Scrum.
7. Backlog та Sprint проєкту.
8. Обов'язкові церемонії Scrum.
9. Допоміжне ПЗ для роботи за принципом Agile.
10. Системи контролю версій.
11. Git команди branch, merge, commit, pull, push, status, add та ін.
12. Локальний та глобальний репозиторії Git.
13. Git-flow.
14. Утиліту з графічним інтерфейсом для роботи з Git.
15. Шаблонами об'єктно-орієнтованого програмування та проєктування.
16. Принципами об'єктно-орієнтованого програмування та проєктування.
17. Принцип SOLID.
18. KISS, DRY та YAGNI принципи.
19. Шаблони GRASP.
20. ПЗ для моделювання систем.
21. CASE-Інструментарій.
22. UML діаграми.
23. Структурні діаграми.
24. Діаграми поведінки.
25. Діаграми взаємодії.
26. Види UML діаграм.
27. Програми для створення діаграм.
28. Клієнт-серверна архітектура.
29. Client (клієнт) та server (сервер).
30. Load balancer (лоад балансер).
31. Тонкий та товстий клієнт.
32. Database (база даних), replica server (репліка сервер) та master (майстер).
33. Request (запит) та response (відповідь).
34. Session (сесія).
35. Public/private network (публічна/приватна мережа).
36. Хостинг та доменне ім'я.
37. DNS.
38. Протокол HTTP.
39. Протокол HTTPS та безпека.
40. Методи HTTP.
41. Базова авторизація.
42. Коди статусу відповіді http.
43. HTTP/2.
44. Заголовки HTTP запиту.
45. Параметри в запиті та тіло запиту.
46. Архітектурний стиль REST. Як він відрізняється від інших стилів.
47. Методи HTTP використовуються в RESTful сервісах і які їхні типові задачі.
48. Ресурси в контексті REST і як вони ідентифікуються.
49. Як структурується URL для доступу до ресурсів у RESTful сервісах.
50. Роль статус-кодів HTTP у RESTful веб-сервісах.
51. Поняття "stateless" у контексті архітектури REST.
52. Безпека RESTful веб-сервісів.

53. Основні компоненти HTTP запиту використовуються у RESTful сервісах.
54. Що таке API та яку роль воно відіграє в REST.
55. Покращення (refactoring) програмного коду.
56. Забезпечення якості програмного забезпечення (quality assurance).
57. Контроль якості програмного забезпечення (quality control).
58. Рівні тестування.
59. Тестова документація.
60. Автоматизація тестування.
61. Поняття інженерії програмного забезпечення.
62. Інструменти та види тестування.
63. Засоби контролю помилок (bug tracking systems).
64. Функціональні та нефункціональні вимоги.
65. Стандарти оформлення коду.
66. Процес інспекції коду.
67. Стратегії випуску програмного забезпечення.
68. Підтримка програмного забезпечення.
69. Unit, Integration та API тестування.

18. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Gruhn V., Striemer R. The Essence of Software Engineering. Saint Philip Street Press, 2020. 248 p.
2. Robert C. Martin The Clean Coder: A Code of Conduct for Professional Programmers. Addison-Wesley; 1. Edition, 2011. 205 p.
3. Software Engineering – Guide to the Software Engineering Body of Knowledge (SWEBOK) 2005-09-15.
4. Бородкіна І., Бородкін Г. Інженерія програмного забезпечення. Центр навчальної літератури, 2018. 204 с.
5. Мельник Н., Левус Є. Вступ до інженерії програмного забезпечення. Львівська політехніка, 2018. 248 с.
6. Піхлер Р. Agile продукт-менеджмент за допомогою Scrum. Створення продуктів, що подобаються клієнтам. Фабула, 2019. 128 с.
7. Табунщик Г., Каплієнко Г., Петрова О. Проектування та моделювання програмного забезпечення сучасних інформаційних систем. Запоріжжя: Дике Поле, 2016. 250 с.

Допоміжна:

1. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Professional, 2018. 208 p.
2. Robertson S., Robertson J. Mastering the requirements process: Getting requirements right (3rd edition). Addison-Wesley, 2013. 541 p.
3. Sommerville I. Software engineering; tenth edition. Harlow: Pearson Education Limited, 2016. 811 p.
4. Державний стандарт України. Основні напрямки оцінювання та відбору CASE-інструментів. ДСТУ 3919–1999. 2000 р.

19. ІНФОРМАЦІЙНІ (ІНТЕРНЕТ) РЕСУРСИ

- <https://classroom.google.com/c/NTI3MjA1NDM0MjYw>
- <https://www.atlassian.com/ru/agile>
- <https://git-scm.com/>
- <https://www.atlassian.com/ru/git/tutorials>
- <https://developer.mozilla.org/ru/docs/Web>
- http://www.tutorialspoint.com/software_engineering/

- <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
- <http://swe.org.ua/>
- <https://softwareengineeringdaily.com/>