

**ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«КИЇВСЬКИЙ ФАХОВИЙ КОЛЕДЖ МІСЬКОГО
ГОСПОДАРСТВА ТАВРІЙСЬКОГО НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ ІМЕНІ В. І. ВЕРНАДСЬКОГО»**

Циклова комісія: комп'ютерно-інтегрованих технологій

ЗАТВЕРДЖУЮ

Заступник директора коледжу з
навчально-виховної роботи

Л.А. Пустовойт

«30» серпня 2023 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

НПП02.07 Основи програмної інженерії

галузь знань:

12 Інформаційні технології

спеціальність:

123 Комп'ютерна інженерія

відділення:

екології, комп'ютерних систем та автоматизації

Київ – 2023 рік

Робоча програма **НПП02.07 «Основи програмної інженерії»** складена відповідно до освітньо-професійної програми підготовки фахових молодших бакалаврів для студентів **IV** курсу галузі **12 «Інформаційні технології»** спеціальності **123 «Комп'ютерна інженерія»**

Розробники: Науменко П.В., викладач-спеціаліст

Робочу програму схвалено на засіданні циклової комісії комп'ютерно-інтегровані технології

Протокол № 1 від «28» серпня 2023 року

Голова циклової комісії: _____  Л.М.Глушко

Розглянуто і рекомендовано до затвердження навчально-методичною радою коледжу

Протокол № 1 від «30» серпня 2023 року

Голова НМР: _____  А.М.Одинець

Зміст

<u>1</u>	<u>ПОЯСНЮВАЛЬНА ЗАПИСКА</u>	4
<u>2</u>	<u>НАВЧАЛЬНО-ТЕМАТИЧНИЙ ПЛАН</u>	6
<u>3</u>	<u>КАЛЕНДАРНО-ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ</u>	7
<u>4</u>	<u>ТЕМИ І ПЛАНИ ЛЕКЦІЙНИХ ЗАНЯТЬ</u>	9
<u>5</u>	<u>ТЕМИ І ПЛАНИ ПРАКТИЧНИХ ЗАНЯТЬ</u>	13
<u>6</u>	<u>ТЕМИ І ПИТАННЯ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТА</u>	16
<u>7</u>	<u>МЕТОДИ АКТИВІЗАЦІЇ НАВЧАЛЬНОГО ПРОЦЕСУ</u>	19
<u>8</u>	<u>СИСТЕМА ПОТОЧНОГО І ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ</u>	21
<u>9</u>	<u>КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ</u>	27
<u>10</u>	<u>РЕКОМЕНДОВАНА ЛІТЕРАТУРА</u>	29

1. ПОЯСНЮВАЛЬНА ЗАПИСКА

Програма вивчення дисципліни Основи програмної інженерії складена для підготовки фахових молодших бакалаврів за освітньо-професійною програмою Обслуговування комп'ютерних систем і мереж для студентів 4 курсу галузі 12 Інформаційні технології спеціальності 123 Комп'ютерна інженерія.

Метою викладання дисципліни «Основи програмної інженерії» є:

- розвинення у студентів фахового стилю мислення;
- надання знань з основ програмної інженерії, необхідних для подальшого вивчення спеціальних дисциплін та для практичної інженерної діяльності;
- ознайомлення студентів з базовими процесами розробки програм;
- підготовки фахівців з розробки інформаційних технологій за допомогою сучасних інструментальних засобів проектування та реалізації програм.

Завдання вивчення дисципліни «Основи програмної інженерії» полягає:

- вивчення проблем та напрямків розвитку технології програмування;
- оволодіння основними методами та засобами автоматизації проектування, виробництва, випробувань та оцінки якості програмного забезпечення;
- вивчення напрямків розвитку методів та програмних засобів колективної розробки програмного забезпечення;
- оволодіння стандартним підходом до розробки програмних проектів, який полягає у використанні моделей життєвого циклу, в процесі яких вбудовані методи проектування, верифікації, тестування та оцінювання проміжних робочих продуктів для того, щоб виявити імовірні ризики та недоліки.

Після вивчення дисципліни " Основи програмної інженерії " студент має **ЗНАТИ**:

- методології та технології програмування;
- методи формування вимог до програмного забезпечення;
- моделі життєвого циклу розробки програмного забезпечення;

- принципи проектування програмного забезпечення;
- методи тестування програм;
- методи оцінки ефективності програм.

У результаті вивчення тем студенти повинні **ВМІТИ**:

- формувати вимоги до програмного забезпечення;
- вибирати найбільш адекватну методологію та технологію програмування;
- обирати метод проектування програмного забезпечення;
- вибирати методи та засоби реалізації програм;
- вибирати метод тестування програм;
- вибирати методи оцінки ефективності програм.

Запропонована послідовність вивчення тем зумовлена логічною схемою формування систематичних знань, вмінь та навичок з даної дисципліни та їх практичного застосування.

Систематичну увагу необхідно приділяти розвитку у студентів навичок самостійної роботи з конспектом, підручниками, навчальними посібниками, таблицями та іншою довідковою літературою.

Процес вивчення дисципліни **НПП02.07 Основи програмної інженерії** спрямований на формування елементів наступних компетентностей:

а) загальні компетентності (ЗК):

К31. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.

PH2. Мати спеціалізовані концептуальні знання, набуті у процесі навчання. Вміти розв'язувати складні задачі і проблеми, які виникають у професійній діяльності. Зрозуміле і недвозначне донесення власних висновків, знань та пояснень, що їх обґрунтовують, до фахівців та нефахівців. Відповідати за прийняття рішень у складних умовах.

К33. Здатність до абстрактного мислення, аналізу та синтезу.

К34. Здатність спілкуватися державною мовою як усно, так і письмово.

КЗ5. Здатність спілкуватися іноземною мовою.

КЗ7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел та практичного її застосування.

КЗ8. Здатність вчитися і бути сучасно навченим.

б) спеціальні (фахові) компетентності (СК):

КФ1. Здатність застосовувати законодавчу та нормативно-правову бази, а також вимоги відповідних, в тому числі і міжнародних, стандартів та практик щодо здійснення професійної діяльності в галузі інформаційних технологій.

КФ4. Здатність до використання сучасних методів побудови та аналізу ефективних алгоритмів, основ теорії чисельних методів, і вміння їх реалізувати в конкретних застосуваннях.

КФ8. Здатність використовувати знання сучасних технологій та інструментальних засобів розробки складних програмних систем (інженерії програмного забезпечення), уміння їх застосовувати на всіх етапах життєвого циклу розробки.

КФ10. Здатність здійснювати вибір, розробляти, розгортати, інтегрувати, діагностувати, адмініструвати та експлуатувати комп'ютерні системи та мережі, мережеві ресурси, сервіси та інфраструктуру організації.

КФ11. Здатність до ділових комунікацій у професійній сфері, знання основ ділового спілкування, навички роботи в команді.

КФ13. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.

КФ14. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати прийняті рішення.

Очікувані результати навчання.

РН1. Знати способи аналізу, синтезу та подальшого сучасного навчання. Вміти проводити аналіз інформації, приймати обґрунтовані рішення, вміти придбати сучасні знання. Встановлювати відповідні зв'язки для досягнення цілей. Нести відповідальність за своєчасне набуття сучасних знань.

PH2. Мати спеціалізовані концептуальні знання, набуті у процесі навчання. Вміти розв'язувати складні задачі і проблеми, які виникають у професійній діяльності. Зрозуміле і недвозначне донесення власних висновків, знань та пояснень, що їх обґрунтовують, до фахівців та нефахівців. Відповідати за прийняття рішень у складних умовах.

PH3. Мати глибокі знання із структури професійної діяльності. Вміти здійснювати професійну діяльність, що потребує оновлення та інтеграції знань. Здатність ефективно формувати комунікаційну стратегію у професійній діяльності. Нести відповідальність за професійний розвиток, здатність до подальшого професійного навчання з високим рівнем автономності.

PH6. Мати досконалі знання державної мови та базові знання іноземної мови. Вміти застосовувати знання державної мови, як усно так і письмово, вміти спілкуватись іноземною мовою. Використовувати при фаховому та діловому спілкуванні та при підготовці документів державну мову. Використовувати іноземну мову у професійній діяльності.

PH11. Володіти базовими знаннями фундаментальних наук, в обсязі, необхідному для освоєння навчальних дисциплін професійної підготовки.

PH13. Діяти на основі законодавчої, нормативно-правової баз України та вимог відповідних стандартів, тому числі міжнародних.

PH14. Володіти навиками аналізу навчальної і спеціальної літератури, нормативних положень, технічної документації для вирішення проблем, що виникають у професійній діяльності.

PH23. Володіти теоретичними (логічними та арифметичними) основами побудови сучасних комп'ютерів.

PH28. Вміти опановувати та розробляти документацію на системи, продукти і сервіси інформаційних технологій, професійно спілкуватись рідною та англійською мовою.

- PH29. Вміти використовувати існуючі та розробляти нові математичні методи, використовувати відповідні закони фізики для вирішення задач, пов'язаних з проектуванням та використанням КСМ.
- PH34. Уміти застосовувати засоби сучасних мов програмування під час програмної реалізації алгоритмів професійних задач.
- PH35. Вміти використовувати математичний апарат при розв'язанні прикладних і наукових завдань в області комп'ютерної інженерії.
- PH36. Вміти застосовувати теоретичні (логічні та арифметичні) основи побудови сучасних комп'ютерів при вирішенні професійних завдань.
- PH43. Вміти економічно мислити, орієнтуватися у конкретних виробничих ситуаціях, аналізувати показники виробничої діяльності підприємства.
- PH45. Практично володіти рідною та однією з іноземних мов в обсязі тематики, зумовленої професійними потребами.
- PH46. Використовувати відповідну термінологію у власних дослідженнях та професійній діяльності державною мовою та/або іноземною; спілкуватися в діалоговому режимі в галузі професійної діяльності; вміти презентувати результати власних досліджень та описувати їх у фахових публікаціях, використовуючи сучасні інформаційні та комунікативні технології.
- PH48. Вдосконалювати професійний та особистісний розвиток протягом усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.
- PH49. Критично осмислювати основні теорії, принципи, методи і поняття у навчанні та професійній діяльності.
- PH50. Дотримуватися етичних норм, враховуючи авторське право та норми академічної доброчесності при проведенні досліджень, розробці програмних продуктів, проектів, презентацій результатів роботи.

2. НАВЧАЛЬНО-ТЕМАТИЧНИЙ ПЛАН

№ тем и	Назва теми	Кількість годин			
		Всього	в тому числі		
			Л	П	СРС
Розділ 1. Проектування та розробка ПЗ.		98	30	18	50
1.1	Основні поняття та проблеми розробки ПЗ.		4	2	4
1.2	Життєвий цикл ПЗ. Міжнародні стандарти життєвого циклу ПЗ.		4	2	10
1.3	Моделі та методології розробки ПЗ.		8	4	12
1.4	Аналіз вимог до програмного забезпечення.		6	4	8
1.5	Документування моделей предметної області з використанням UML.		4	2	8
1.6	Шаблони проектування ПЗ.		4	4	8
Розділ 2. Менеджмент програмних проєктів.		34	10	8	16
2.1	Введення в управління проєктами.		10	8	16
Розділ 3. Методи забезпечення та контролю якості ПЗ.		48	12	16	20
3.1	Якість ПЗ. Метрики і стандарти якості ПЗ.		4	10	10
3.2	Введення в тестування програмного забезпечення.		8	6	10
Всього		180	52	42	86

3. КАЛЕНДАРНО-ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№	Тема	Форма занять, кількість годин		
		Л.	Пр.	С.р.
Розділ 1. Проектування та розробка ПЗ.				
1.	Основні поняття та проблеми розробки ПЗ.	2		2
2.	Поняття, означення та проблеми розробки ПЗ. Програмне забезпечення та інструментарій технології програмування.		2	
3.	Програмне забезпечення та інструментарій технології програмування.	2		2
4.	Життєвий цикл ПЗ (програмного забезпечення).	2		6
5.	Міжнародні стандарти життєвого циклу ПЗ.	2		
6.	Життєвий цикл ПЗ (програмного забезпечення). Міжнародні стандарти життєвого циклу ПЗ.		2	
7.	Основні типи моделей життєвого циклу.	2		6
8.	Моделі та методології розробки ПЗ.	2		
9.	Основні типи моделей життєвого циклу. Моделі та методології розробки ПЗ.		2	
10.	Загальні підходи до моделювання ПЗ.	2		4
11.	Аналіз вимог до програмного забезпечення.	2		
12.	Аналіз вимог до програмного забезпечення.		2	
13.	Експлуатація та супроводження готової програмної системи. Інженерія вимог як процес.	2		8
14.	Експлуатація та супроводження готової програмної системи.		2	
15.	Інженерія вимог як процес		2	
16.	Документування моделей предметної області з використанням UML.	2		8
17.	Концептуальна модель UML.	2		
18.	Зв'язки в UML.	2		6
19.	Діаграми та правила UML.	2		
20.	Зв'язки в UML. Діаграми та правила UML.		2	
21.	Шаблони (патерни) проектування ПЗ. Патерни програмування.	2		4
22.	Шаблони (патерни) проектування ПЗ.		2	
23.	Архітектурні шаблони програмного забезпечення.	2		4
24.	Шаблони проектування та програмування ПЗ.		2	
	Всього по розділу	30	18	50
Розділ 2. Менеджмент програмних проектів.				
25.	Введення в управління проектами.	2		4
26.	Методи управління проектами.	2		4

№	Тема	Форма занять, кількість годин		
		Л.	Пр.	С.р.
27.	Методи та структура управління проектами.		2	
28.	Програмне забезпечення для управління проектами.	2		4
29.	Програмне забезпечення для управління проектами.		2	
30.	Системи контролю проектів.	2		4
31.	Теми з управління проектами. Міжнародні стандарти.	2		
32.	Системи контролю проектів.		2	
33.	Теми з управління проектами. Міжнародні стандарти управління проектами.		2	
	Всього по розділу	10	8	16
	Розділ 3. Методи забезпечення та контролю якості ПЗ.			
34.	Якість ПЗ. Метрики і стандарти якості ПЗ.	2		2
35.	Якість ПЗ.		2	
36.	Метрики і стандарти якості ПЗ.		2	
37.	Система управління якістю. Види тестування.	2		4
38.	Процес сертифікації програм на базі інформації про їх використання.		2	
39.	Види тестування.		2	
40.	Сертифікація ПЗ з участю користувачів, її переваги.		2	
41.	Введення в тестування програмного забезпечення.	2		2
42.	Особливості вимог програмного забезпечення.	2		6
	Методи і фази тестування.	2		
43.	Необхідні і достатні умови для проведення тестування.	2		6
44.	Цілі і завдання процесу тестування та необхідні і достатні умови для проведення тестування.		2	
45.	Методи тестування. Вимоги програмного забезпечення.		2	
46.	Фази тестування.		2	
	Всього по розділу	12	16	20
	Всього по предмету	52	42	86

4. ТЕМИ І ПЛАНИ ЛЕКЦІЙНИХ ЗАНЯТЬ

№	Назва теми	Кількість годин
	Розділ 1. Проектування та розробка ПЗ.	
	Тема 1.1 Основні поняття та проблеми розробки ПЗ.	
Л-1	<i>План.</i> 1. Основні характеристики програмних продуктів. 2. Класи програмних продуктів. 3. Системне програмне забезпечення. 4. Інструментарій технології програмування.	2
Л-2	<i>План.</i> 5. Пакети прикладних програм. 6. Класифікація прикладного програмного забезпечення. 7. Технологія програмування в історичному аспекті.	2
	Тема 1.2 Життєвий цикл ПЗ. Міжнародні стандарти життєвого циклу ПЗ.	
Л-3	<i>План.</i> 1. Сучасні підходи до розроблення і впровадження інформаційних систем. 2. Основні розділи програмної інженерії. 3. Поняття життєвого циклу програмного забезпечення.	2
Л-4	<i>План.</i> 4. Основні процеси життєвого циклу ПЗ. 5. Життєвий цикл програмного забезпечення. 6. Основні типи моделей ЖЦ. 7. Міжнародні стандарти та методології розробок ПЗ.	2
	Тема 1.3 Моделі та методології розробки ПЗ.	
Л-5	<i>План.</i> 1. Поняття технології розробки програми. 2. Основа розробки програмного забезпечення. 3. Моделі життєвого циклу. 4. Rational Objectory Process – модель життєвого циклу.	2
Л-6	<i>План.</i> 5. Життєвий цикл UML (Rational Objectory Process). 6. Специфікування і планування. 7. Процес розробки. 8. Випуск продукту і механізми зворотного зв'язку.	2
Л-7	<i>План.</i>	2

№	Назва теми	Кількість годин
	9. <i>Загальні підходи до моделювання програмного забезпечення.</i> 10. <i>Основні підходи та концепції моделювання.</i>	
Л-8	<i>План.</i> 11. <i>Принципи моделювання.</i> 12. <i>Об'єктне моделювання.</i>	2
	Тема 1.4 Аналіз вимог до програмного забезпечення.	
Л-9	<i>План.</i> 1. <i>Вимоги до програмного забезпечення</i> 2. <i>Проектування програмного забезпечення.</i> 3. <i>Визначення вимог.</i>	2
Л-10	<i>План.</i> 4. <i>Експлуатація та супроводження готової програмної системи.</i> 5. <i>Інженерія вимог як процес.</i>	2
Л-11	<i>План.</i> 6. <i>Складання вимог.</i> 7. <i>Аналіз та класифікація вимог.</i>	2
	Тема 1.5 Документування моделей предметної області з використанням UML.	
Л-12	<i>План.</i> 1. <i>Введення в UML</i> 2. <i>Поняття та означення UML.</i> 3. <i>Концептуальна модель UML.</i>	2
Л-13	<i>План.</i> 1. <i>Зв'язки в UML. Діаграми UML.</i> 2. <i>Правила UML.</i> 3. <i>Загальні механізми UML.</i>	2
	Тема 1.6 Шаблони проектування ПЗ.	
Л-14	<i>План.</i> 1. <i>Історія виникнення шаблонів проектування програмного забезпечення.</i> 2. <i>Поняття патерну програмування.</i> 3. <i>Види патернів програмування</i> 4. <i>Патерни проектування</i>	2
Л-15	<i>План.</i> 5. <i>Шаблони програмного забезпечення: опис, призначення, використання, недоліки.</i> 6. <i>Проблеми вибору архітектурних шаблонів програмних систем.</i>	2
	Розділ 2. Менеджмент програмних проєктів.	

№	Назва теми	Кількість годин
	Тема 2.1 Введення в управління проектами.	
Л-16	<i>План.</i> 1. <i>Поняття проекту та історія виникнення управління проектами.</i> 2. <i>Методи управління проектами.</i> 3. <i>Критичний шлях управління проектом.</i>	2
Л-17	<i>План.</i> 4. <i>Екстремальне управління проектами.</i> 5. <i>Принципи управління послідовністю подій.</i> 6. <i>Проекти в контрольованому середовищі.</i>	2
Л-18	<i>План.</i> 7. <i>Процесне управління.</i> 8. <i>Групи процесів управління проектами: ініціювання, планування та розробка, виконання, моніторинг та контроль, завершення.</i>	2
Л-19	<i>План.</i> 9. <i>Системи контролю проектів.</i> 10. <i>Теми з управління проектами.</i>	2
Л-20	<i>План.</i> 11. <i>Міжнародні стандарти.</i> 12. <i>Програмне забезпечення для управління проектами.</i>	2
	Розділ 3. Методи забезпечення та контролю якості ПЗ.	
	Тема 3.1 Якість ПЗ. Метрики і стандарти якості ПЗ.	
Л-21	<i>План.</i> 1. <i>Оцінка якості процесів створення програмного забезпечення.</i> 2. <i>Серія стандартів ISO 9000.</i> 3. <i>Вибір показників якості.</i> 4. <i>Загальна схема процесів оцінки характеристик якості програм.</i>	2
Л-22	<i>План.</i> 5. <i>Система управління якістю.</i> 6. <i>Процес сертифікації програм на базі інформації про їх використання.</i> 7. <i>Модель обмеженої гарантії на ПЗ.</i> 8. <i>Сертифікація ПЗ з участю користувачів.</i>	2
Л-23	<i>План.</i> 9. <i>Переваги моделі сертифікації з участю користувачів.</i>	2

№	Назва теми	Кількість годин
	10. <i>Тестування як забезпечення якості ПЗ.</i> 11. <i>Класифікація та види тестування за різними показниками.</i>	
	Тема 3.2 Введення в тестування програмного забезпечення.	
Л-24	<i>План.</i> 1. <i>Етапи життєвого циклу програмного забезпечення.</i> 2. <i>Основні поняття в тестуванні програмного забезпечення.</i>	2
Л-25	<i>План.</i> 3. <i>Цілі і завдання процесу тестування.</i> 4. <i>Необхідні і достатні умови для проведення тестування.</i>	2
Л-26	<i>План.</i> 5. <i>Основні визначення та атрибути тестування.</i> 6. <i>Вимоги програмного забезпечення. Методи та фази тестування. Класи еквівалентності.</i> 7. <i>Рівні тестування.</i> 8. <i>Види тестування ПЗ.</i>	2
	Всього годин	52

5. ТЕМИ І ПЛАНИ ПРАКТИЧНИХ ЗАНЯТЬ

№	Назва теми	Кількість годин
	Розділ 1. Проектування та розробка ПЗ.	
	Тема 1.1 Основні поняття та проблеми розробки ПЗ.	
П-1	<p><i>План.</i></p> <ol style="list-style-type: none"> 1. <i>Класи та характеристики програмних продуктів.</i> 2. <i>Системне програмне забезпечення.</i> 3. <i>Інструментарій технології програмування.</i> 4. <i>Пакети прикладних програм.</i> 5. <i>Класифікація прикладного програмного забезпечення.</i> 6. <i>Сучасні підходи до розроблення і впровадження інформаційних систем.</i> 7. <i>Розділи програмної інженерії.</i> 8. <i>Життєвий цикл програмного забезпечення.</i> 	2
	Тема 1.2 Життєвий цикл ПЗ. Міжнародні стандарти життєвого циклу ПЗ.	
П-2	<p><i>План.</i></p> <ol style="list-style-type: none"> 1. <i>Процеси життєвого циклу програмного забезпечення.</i> 2. <i>Типи моделей ЖЦ (життєвого циклу).</i> 3. <i>Міжнародні стандарти ПЗ.</i> 4. <i>Технології розробки програми.</i> 5. <i>Основа розробки програмного забезпечення.</i> 6. <i>Моделі життєвого циклу.</i> 	2
	Тема 1.3 Моделі та методології розробки ПЗ.	
П-3	<p><i>План.</i></p> <ol style="list-style-type: none"> 1. <i>Rational Objectory Process – модель життєвого циклу.</i> 2. <i>Життєвий цикл UML (Rational Objectory Process).</i> 3. <i>Загальні підходи до моделювання програмного забезпечення.</i> 4. <i>Основні принципи та концепції моделювання.</i> 5. <i>Об'єктне моделювання.</i> 	2
П-4	<p><i>План.</i></p> <ol style="list-style-type: none"> 6. <i>Загальні підходи до моделювання програмного забезпечення.</i> 7. <i>Основні принципи та концепції моделювання.</i> 8. <i>Об'єктне моделювання.</i> 	2
	Тема 1.4 Аналіз вимог до програмного забезпечення.	
П-5	<p><i>План.</i></p> <ol style="list-style-type: none"> 1. <i>Вимоги до програмного забезпечення.</i> 	2

№	Назва теми	Кількість годин
	2. <i>Проектування програмного забезпечення.</i> 3. <i>Визначення вимог до ПЗ.</i>	
П-6	<i>План.</i> 4. <i>Експлуатація та супроводження готової програмної системи.</i> 5. <i>Процес інженерії вимог.</i> 6. <i>Аналіз та класифікація та складання вимог.</i>	2
	Тема 1.5 Документування моделей предметної області з використанням UML.	
П-7	<i>План.</i> 1. <i>Поняття та означення UML.</i> 2. <i>Концептуальна модель UML.</i> 3. <i>Зв'язки UML.</i> 4. <i>Види та типи діаграм UML.</i> 5. <i>Правила UML.</i> 6. <i>Загальні механізми UML.</i>	2
	Тема 1.6 Шаблони проектування ПЗ.	
П-8	<i>План.</i> 1. <i>Шаблон проектування програмного забезпечення.</i> 2. <i>Поняття та види патернів програмування.</i>	2
П-9	<i>План.</i> 3. <i>Шаблони програмного забезпечення: опис, призначення, використання, недоліки.</i> 4. <i>Проблеми вибору архітектурних шаблонів програмних систем.</i>	2
	Розділ 2. Менеджмент програмних проектів.	
	Тема 2.1 Введення в управління проектами.	
П-10	<i>План.</i> 1. <i>Методи управління проектами.</i> 2. <i>Критичний шлях управління проектом.</i>	2
П-11	<i>План.</i> 3. <i>Екстремальне управління проектами.</i> 4. <i>Принципи управління послідовністю подій.</i>	2
П-12	<i>План.</i> 5. <i>Проекти в контрольованому середовищі.</i> 6. <i>Процесне управління.</i>	2
П-13	<i>План.</i> 7. <i>Управління проектами: ініціювання, планування та розробка, виконання, моніторинг та контроль, завершення.</i>	2

№	Назва теми	Кількість годин
	8. Системи контролю та теми з управління проектами. 9. Міжнародні стандарти. 10. Програмне забезпечення для управління проектами.	
	Розділ 3. Методи забезпечення та контролю якості ПЗ.	
	Тема 3.1 Якість ПЗ. Метрики і стандарти якості ПЗ.	
П-14	План. 1. Оцінка якості процесів створення програмного забезпечення. 2. Серія стандартів ISO 9000.	2
П-15	План. 3. Показники якості та загальна схема процесів оцінки характеристик якості програм. 4. Система управління якістю.	2
П-16	План. 5. Процес сертифікації програм на базі інформації про їх використання. 6. Модель обмеженої гарантії на ПЗ.	2
П-17	План. 7. Сертифікація програмного забезпечення з участю користувачів та переваги цієї моделі сертифікації. 8. Тестування як забезпечення якості ПЗ. 9. Класифікація та види тестування за різними показниками.	2
	Тема 3.2 Введення в тестування програмного забезпечення.	
П-18	План. 1. Етапи життєвого циклу програмного забезпечення. 2. Цілі і завдання процесу тестування.	2
П-19	План. 3. Поняття в тестуванні програмного забезпечення. 4. Умови для проведення тестування.	2
П-20	План. 5. Основні визначення та атрибути тестування. 6. Вимоги програмного забезпечення.	2
П-21	План. 7. Методи та фази тестування. 8. Класи еквівалентності. 9. Рівні та види тестування ПЗ.	2
	Всього годин	42

6. ТЕМИ І ПИТАННЯ ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТА

№	Назва теми	Кількість годин
	Розділ 1. Проектування та розробка ПЗ.	
	Тема 1.1 Основні поняття та проблеми розробки ПЗ.	
С-1	Дерево характеристик якості програмних продуктів представлено. Характеристика пакетів прикладних програм. Системне програмне забезпечення.	2
С-2	Класифікація програм за категоріями користувачів. Базове та сервісне програмне забезпечення. Програмні продукти для створення додатків. Програмні засоби мультимедіа. Системи штучного інтелекту.	2
	Тема 1.2 Життєвий цикл ПЗ. Міжнародні стандарти життєвого циклу ПЗ.	
С-3	Стандартизація життєвого циклу ПЗ. Розподіл процесів життєвого циклу за стандартом стандарту ISO/IEC 12207.	6
С-4	Процеси життєвого циклу ПЗ. Моделі ЖЦ. Процеси розроблення програмного забезпечення. Міжнародні стандарти розробок ПЗ.	6
С-5	Проектування архітектури ПЗ. Детальне проектування ПЗ. Методи XP-програмування.	4
	Тема 1.3 Моделі та методології розробки ПЗ.	
С-6	Етапи та технології розробки програм. Каскадна модель життєвого циклу ПЗ. Ітераційна модель життєвого циклу ПЗ. Життєвий цикл UML (Rational Objectory Process). Основні властивості ROP-технології.	8
С-7	Процес розробки . Специфікування і планування. Випуск продукту і механізми зворотного зв'язку. Моделювання програмного забезпечення. Об'єктне моделювання. Найважливіші моделі програмних систем. Об'єктно-орієнтоване моделювання.	8
	Тема 1.4 Аналіз вимог до програмного забезпечення.	
С-8	Збір та аналіз вимог замовника. Проектування, реалізація, тестування. Схема водоспадної (каскадної) моделі життєвого циклу програмних систем, її цінність та недоліки.	6

№	Назва теми	Кількість годин
	Схема спіральної моделі життєвого циклу розробки. Дослідження джерел відомостей про вимоги. Встановлення пріоритетності вимог. Інженерія складання вимог.	
	Тема 1.5 Документування моделей предметної області з використанням UML.	
С-9	Діаграма класів (Class diagram) UML. Механізми UML. Моделювання поведінки системи. Побудова програмної системи (ПС) методом UML. Візуалізація системи та опис деталей. Ключові абстракції.	4
	Тема 1.6 Шаблони проектування ПЗ.	
С-10	Категорії патернів програмування. Типи шаблонів GOF. Патерни, які входять до складу шаблонів GRASP. Схема класифікації шаблонів проектування інформаційних систем. Опис шаблону програмного забезпечення, його призначення, використання та недоліки. Приклади архітектурних шаблонів програмних систем та проблеми їх вибору.	4
	Розділ 2. Менеджмент програмних проектів.	
	Тема 2.1 Введення в управління проектами.	
С-11	Мережева діаграма PERT для семимісячного проекту з п'ятьма ключовими подіями. Схема та 5-ти складових проекту в «традиційному методі» розвитку проекту.	4
С-12	Цикли планування та зворотного зв'язку в екстремальному програмуванні з часовими межами повторюваних циклів.	4
С-13	Принципи управління послідовністю подій. Теорії обмежень (Theory of Constraints - TOC) для проектів.	4
С-14	Модель процесів PRINCE2. Програмного забезпечення Microsoft Project, Time Line 6.5.	4
	Розділ 3. Методи забезпечення та контролю якості ПЗ.	
	Тема 3.1 Якість ПЗ. Метрики і стандарти якості ПЗ.	
С-15	Серія стандартів якості ПЗ ISO 9000.	2

№	Назва теми	Кількість годин
	<p>Означення з уточненнями у стандарті ISO 9126-1 для кожної характеристики програмного засобу. Стандарти – ISO 9126-2, ISO 9126-3 та ISO 9126-4 . Міжнародний стандарт ISO 14598.</p>	
C-16	<p>Реалізація процесів оцінки у відповідності з адаптованою версією стандарту ISO 12207. Стандарти CMM на базі моделі Capability Maturity Model for Software (CMM). Компонентна розробка програм (CBSE – component based software engineering). Види та класифікація тестування за різними показниками.</p>	4
	Тема 3.2 Введення в тестування програмного забезпечення.	
C-17	<p>Цикли тестування web-сторінки. Тестування в реальних умовах. Тестування якості web-сторінок.</p>	2
C-18	<p>Розробки тестів з реальними вхідними даними для перевірки правильності роботи ПЗ. Виявлення різних аномальних станів у ПЗ незалежними тестувальниками.</p>	6
C-19	<p>Використання Selenium WebDriver для автоматизованого тестування. Можливості роботи WebDriver. Патерни автоматизованого функціонального тестування. Основні тестові випадки. Запуск тестів.</p>	6
	Всього годин	86

7. МЕТОДИ АКТИВІЗАЦІЇ НАВЧАЛЬНОГО ПРОЦЕСУ

Для активізації навчально-пізнавальної діяльності студентів при вивченні дисципліни "Основи програмної інженерії" застосовуються наступні методики:

Міні-лекції передбачають виклад навчального матеріалу з використанням мультимедійних засобів за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції, як правило, проводяться як частина заняття-дослідження.

Робота в малих групах — використовується з метою активізації роботи студентів при проведенні практичних занять. Це так звані групи психологічного комфорту, де кожен учасник відіграє свою особливу роль і певними своїми якостями доповнює інших. Використання цієї технології створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

В процесі проведення **практичних занять** доцільно використовувати бригадний метод роботи. Студенти поділяються на **індивідуально-типологічні групи**. Всі студенти отримують аналогічні завдання, однак різним типологічним групам викладач надає різний вид допомоги з метою розвитку їх індивідуальних здібностей. Вид допомоги (надання опорного конспекту, алгоритму розв'язування завдання даного типу, визначення типових помилок в процесі розв'язування аналогічних завдань тощо) залежить від того до якої типологічної групи входить студент. Після завершення розв'язування задачі в кожній групі проходить обговорення отриманого розв'язку, допущених помилок, проводиться оцінювання результатів навчальної діяльності кожного студента.

Самостійна пізнавальна діяльність студентів, що проводиться в позаурочний час, повинна бути спрямована на розвиток творчих здібностей та практичних навичок студентів. З цією метою створюється комплект

завдань підвищеного рівня складності. В залежності від рівня підготовки студентів робота з завданнями організовується індивідуально або в малих групах студентів. Студенти з високим рівнем підготовки працюють самостійно, інші студенти використовують бригадний метод роботи. Результати виконання цих завдань презентуються на практичних заняттях.

Презентації – виступи перед аудиторією – використовуються для представлення певних досягнень, результатів роботи студентів, звіту про виконання індивідуальних завдань з використанням технічних засобів.

Підсумкове заняття, що передбачає використання кейс-методу (метод аналізу конкретних ситуацій): предметна область пропонується студентами, кейс (ситуація та задачі до неї) формулюється викладачем, розробка проекту та його презентація відбувається на занятті.

Творчий проект: предметна область пропонується студентами, створення інформаційної моделі, проведення аналізу та розробка проекту відбувається поза заняттями під керівництвом викладача, презентація проекту відбувається на занятті.

Застосування таких форм організації навчально-пізнавальної діяльності студентів передбачає роботу над завданнями, що спрямовані на розвиток креативності. Тематика завдань є поглибленням або розширенням тематики програми дисципліни і має практичне підґрунтя, пов'язане з реаліями життя студента. Завдання носять або виключно проблемний характер, і тоді студент самостійно знаходить шляхи розв'язання, або містять алгоритм виконання. Але в будь-якому випадку ці завдання спрямовані не стільки на набуття якихось навичок, скільки на розвиток вміння аналізувати проблему, формалізувати її, шукати власні шляхи її вирішення за допомогою вивченого інструментарію, отримувати та аналізувати результат, проводити порівняльні характеристики тощо.

8. СИСТЕМА ПОТОЧНОГО І ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

Для визначення рівня засвоєння студентами навчального матеріалу використовуються наступні методи оцінювання знань:

1. Поточне оцінювання

Поточне оцінювання студентів з основ програмної інженерії проводиться безпосередньо під час навчальних занять або за результатами виконання домашніх завдань, усних відповідей, презентацій, тощо.

2. Тематичне оцінювання

Тематичному оцінюванню підлягають основні результати вивчення теми (розділу). Тематична оцінка виставляється на підставі результатів опанування студентами матеріалу теми впродовж її вивчення з урахуванням поточних оцінок, різних видів навчальних робіт (самостійних, творчих, контрольних робіт) та навчальної активності студентів.

3. Оцінки за індивідуальну самостійну роботу

Самостійна робота студентів організовується у відповідності з навчально-тематичним планом і передбачає самостійне опрацювання на основі навчальної та наукової літератури окремих питань з кожної теми дисципліни, опрацювання, прослуханого лекційного матеріалу і підготовку до практичних та контрольних занять, виконання запланованих викладачем самостійних письмових індивідуальних робіт і презентацій.

4. Семестрове оцінювання.

Семестрове оцінювання здійснюється за результатами тематичного оцінювання з урахуванням динаміки особистих навчальних досягнень студентів з предмета протягом семестру, важливість теми, тривалість її вивчення, складність змісту тощо. Семестрова оцінка може підлягати коригуванню. У разі підвищення оцінки виставляється скоригована оцінка.

5. Залік/Іспит.

Формою підсумкового контролю є складання заліку.

Питання до заліку з дисципліни «Основи програмної інженерії»

1. Поняття «програми», «програмного забезпечення», «програмного продукту». Розподіл всіх програм за характером використання і категоріям користувачів (на 2 класи).
2. Основні характеристики програм.
3. Навести дерево характеристик якості програмних продуктів.
4. Надати означення мобільності, надійності роботи та ефективності програмного продукту.
5. Надати означення обліку людського фактора, модифікованості та комунікативності програмних продуктів.
6. Важливі характеристики програмних продуктів в умовах існування ринку.
7. Класифікація програмних продуктів, у якій основною ознакою є сфера (область) використання.
8. Поняття системного програмного забезпечення, поняття пакету прикладних програм та поняття інструментарія технології програмування.
9. Структура системного програмного забезпечення.
10. Поняття «утиліти». Навести приклади утиліт.
11. Поняття CASE-технології. Навести приклади програмних продуктів для створення додатків.
12. Перерахувати класи пакетів прикладних програм (ППП). Поняття проблемно-орієнтованих ППП, ППП автоматизованого проектування, ППП загального призначення, методо-орієнтованих ППП.
13. Поняття офісних ППП, настільних видавничих систем, програмних засобів мультимедіа, систем штучного інтелекту.
14. Модульне програмування. Його архітектура, відміна та перевага.
15. Архітектура програми при об'єктно-орієнтованому підході. Його переваги.
16. Поняття *ActiveX* – технології, її основні переваги.
17. Поняття операційної оболонки.
18. Поняття мови програмування та її умовний розділ на класи.
19. Компоненти системи програмування. Поняття інструментального середовища користувача та його компоненти.
20. Поняття та означення системи управління базами даних (СУБД), серверу БД, генератору звітів (серверу звітів), текстового процесора,

табличного процесора, засобу створення презентаційної графіки, інтегрованих пакетів.

21. Перерахувати основні розділи програмної інженерії, фактори еталонної моделі програмної інженерії, типи інформації для економічного аналізу.
22. Поняття життєвого циклу програмного забезпечення. Три групи процесів ЖЦПЗ.
23. Які процеси входять до основних процесів ЖЦПЗ.
24. Які етапи включає в себе процес розроблення ПЗ.
25. Поняття інтеграції ПЗ, кваліфікаційної вимоги, кваліфікаційного тестування ПЗ, інтеграції системи, приймання ПЗ, процесу експлуатації.
26. Означення підготовчої роботи, експлуатаційного тестування, експлуатації системи, підтримки користувачів, процесу супроводу. Які задачі включає в себе підготовча робота служби супроводу.
27. Які задачі включає аналіз проблем і запитів на модифікацію ПЗ, що виконуються службою супроводу. В чому полягають перевірка і приймання. Як виконується зняття ПЗ з експлуатації.
28. Поняття моделі ЖЦ (життєвого циклу) та основні типи моделей ЖЦ.
29. Перерахувати прогнозовані та адаптивні моделі ЖЦ.
30. Поняття та основна ідея XP-програмування. 12 методів XP-програмування.
31. Схема та особливості каскадної моделі життєвого циклу ПЗ.
32. Три основні підходи до розроблення ПЗ, їх означення.
33. Чотири принципи моделювання ПЗ та їх означення.
34. Надати означення алгоритмічного та об'єктно-орієнтованого підходів при об'єктному моделюванні.
35. Джерела відомостей про вимоги при складанні вимог.
36. Методи складання вимог. Означення продукту процесу складання вимог.
37. Класифікація та означення вимог на першому кроці аналізу вимог.
38. Означення кроків аналізу вимог.
39. Поняття та означення UML.
40. Поняття концептуальної моделі UML.
41. Означення чотирьох видів сутностей UML.
42. Поняття варіанту використання (use case), активного класу (active class), компоненту (component), артефакту (artifact), вузла (node) в моделі UML.
43. Поняття поведінкових сутностей (Behavioral things), взаємодії (Interaction), автомату (state machine), діяльності (activity) в моделях UML.

44. Поняття пакету (package), сутності анотування в моделях UML. Чотири типи зв'язків у UML.
45. Означення залежності (dependence), асоціації (association), узагальнення (generalization), реалізації (realization) в типах зв'язків у UML.
46. Поняття діаграми UML.
47. Навести 13 видів найчастіше використовуваних для розроблення моделей та архітектурних виглядів ПЗ UML-діаграм.
48. Означення правил UML. Ссинтаксичні й семантичні правила, які входять до UML.
49. Поняття загальних механізмів UML та його види.
50. Означення специфікації (Specifications), доповнення (Adornments), загальноприйнятих поділів (Common divisions), механізмів розширення (Extensibility mechanisms) в UML-механізмах.
51. Означення стереотипу (Stereotype), присвоєного значення (Tagged value), обмеження (Constraints) в UML.
52. Поняття шаблону проектування програмного забезпечення (software design patterns).
53. Типи шаблонів GOF та GRASP.
54. Ряд інших назв патернів програмування. Хто і в якому році склав перелік шаблонів проектування.
55. Види та означення патернів програмування.
56. Поняття патерну проектування. Категорії патернів проектування програмних систем.
57. Означення архітектурних патернів та ідіом. Завдання кожного патерну.
58. Елементи опису патерну, та їх поняття.
59. Схема класифікації шаблонів проектування інформаційних систем.
60. Групи, у які об'єднуються архітектурні патерни.
61. Поняття архітектурних шаблонів програмного забезпечення. Їх порівняння із шаблонами проектування. Приклади архітектурних шаблонів.
62. Поняття структурних шаблонів, їх рівні. Перелік структурних шаблонів.
63. Поняття та означення шаблону проектування Model-View-ViewModel.
64. Призначення, використання та опис шаблону MVVM (Model-View-ViewModel).
65. Поняття та способи формулювання закону Деметри (Law of Demeter, LoD).
66. Переваги та недоліки закону Деметри.

67. Поняття command-query separation (CQS) або command-query responsibility segregation (CQRS). Його недоліки.
68. Застосування CQRS (command-query responsibility segregation) у контрактному програмуванні.
69. Поняття породжуючих шаблонів (Creational patterns).
70. Перелік шаблонів, що породжують (Creational patterns).
71. Проблеми вибору архітектурних шаблонів програмних систем.
72. Поняття проекту та управління проектами (Project Management).
73. Методи управління проектами. Типові етапи виконання проекту при традиційном методі.
74. Схема 5 складових проекту в «традиційному методі».
75. Поняття критичного шляху управління проектом (Critical Chain Project Management - CCPM).
76. Означення екстремального управління проектами (екстремальне проектне управління).
77. Поняття та принципи управління послідовністю подій (Event chain methodology).
78. Поняття управління проектами в контрольованому середовищі (Projects in controlled environments - PRINCE).
79. Групи процесів управління проектами.
80. Схема та поняття групи процесів ініціювання.
81. Склад групи процесів планування.
82. Означення групи процесів виконання.
83. Означення групи процесів моніторингу та контролю та що вона в себе включає.
84. Склад та означення групи процесів завершення проекту. Навести схему.
85. Системи контролю проектів. Поняття та означення контролю проектів. Характеристики гарного формального плану впровадження системи.
86. Теми з управління проектами. Поняття менеджера проекту, схема та означення трикутника управління проектом.
87. Означення структури декомпозиції робіт (Work breakdown Structure - WBS) та структури управління проектами.
88. Приклади міжнародних стандартів управління проектами.
89. Програмне забезпечення для управління проектами, навести приклади.
90. Означення ключових понять (у різні періоди життєвого циклу проекту): планування, контролю, управління.
91. План проекту (7 пунктів) при управлінні проектом.

92. Стандарти для оцінки якості процесів створення програмного забезпечення.
93. Три групи характеристик різних категорій метрик стандарту атрибутів якості програмних засобів ISO 9126–1.
94. Етапи вибору показників якості програмних засобів.
95. Поняття функціональної пригодності, оцінки коректності програмних засобів, оцінки здатності до взаємодії, оцінки захищеності програмних засобів, оцінки надійності, оцінки практичності, супроводжуваності, оцінки мобільності.
96. Означення характеристик і субхарактеристик якості ПЗ (ISO 9126-1).
97. Процес сертифікації програм на базі інформації про їх використання.
98. Схема якості ПЗ та її характеристики.
99. Класифікація видів тестувань за різними показниками.
100. Основна мета процесу тестування. Основне завдання тестування ПЗ. Необхідні і достатні умови для проведення тестування.
101. Означення вимоги (Requirements) до програмного забезпечення та її складові.
102. Означення фаз тестування та класів еквівалентності (Equivalence class).

9. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ

Відповідно до ступеня оволодіння зазначеними знаннями і способами діяльності виокремлюються такі рівні навчальних досягнень студентів з архітектури комп'ютерів:

- **«незадовільно»** - студент не опанував навчальний матеріал з дисципліни «Основи програмної інженерії», не знає технічних термінів, наукових фактів, визначень, майже не орієнтується в рекомендованій літературі, відсутні наукове мислення, практичні навички не сформовані.

- **«задовільно»** - студент володіє навчальним матеріалом на рівні елементарного розпізнавання та відтворення окремих схем, таблиць, понять, алгоритмів, елементів, об'єктів, що становлять незначну частину навчального матеріалу чи за допомогою викладача виконує елементарні завдання і повторює інформацію, операції, дії, засвоєні ним у процесі навчання, здатний вирішувати завдання за зразком.

- **«добре»** - коли студент самостійно застосовує знання в стандартних ситуаціях, володіє знаннями про основні методи та засоби автоматизації проектування, виробництва, випробувань та оцінки якості програмного забезпечення, знає напрямки розвитку методів та програмних засобів колективної розробки програмного забезпечення, знає методи проектування, верифікації, тестування та оцінювання програмних продуктів для виявлення імовірних ризиків та недоліків програмного забезпечення та здатний вирішувати завдання за зразком.

- **«відмінно»** - студент здатний самостійно орієнтуватися в нових для нього ситуаціях, скласти план дій і виконувати його, пропонувати нові,

невідомі йому раніше рішення, тобто його діяльність має дослідницький характер.

Оцінювання якості підготовки студентів з дисципліни «Основи програмної інженерії» здійснюється в двох аспектах: *рівень володіння теоретичними знаннями*, який можна виявити в процесі усного опитування, та *якість практичних умінь і навичок*, тобто здатність до застосування вивченого матеріалу під час приведення прикладів проектування програмного забезпечення, методів тестування і верифікації програмного забезпечення та оцінки ефективності програм.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Бейзер Б. Тестирование черного ящика / Б. Бейзер. – Москва: Русское слово, 2005. – 448 с.
2. Блэк Р. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование / Р. Блэк. – Москва: Лори, 2006. – 544 с.
3. Брауде Э. Д. Технология разработки программного обеспечения / Э. Дж Брауде. – Санкт-Петербург: Питер, 2004. – 655 с.
4. Гагарина Л. Г. Технология разработки программного обеспечения: учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – Москва: Форум, 2008. – 400 с. – (ИНФРА-М).
5. Дастин Э. Автоматизированное тестирование программного обеспечения / Э. Дастин, Д. Рэшка, Д. Пол. – Москва: Лори, 2003. – 592 с. – (Перевод с английского).
6. Канер С. Тестирование программного обеспечения: Фундаментальные концепции менеджмента бизнес-приложений / С. Канер, Д. Фолк, Е. Нгуен. – Киев: ДиаСофт, 2001. – 544 с.
7. Лаврищева Е. М. Методы и средства инженерии программного обеспечения: Учебник / Е. М. Лаврищева, В. А. Петрухин. – Москва: МФТИ(ГУ), 2006. – 304 с.
8. Лавріщева К. І. Програма інженерія / К. І. Лавріщева. – Київ: Видавничий дім «Академперіодика» НАН України, 2008. – 319 с.
9. Основы инженерии качества программных систем / [Ф. И. Андон, Г. И. Коваль, Т. М. Коротун та ін.]. – Киев: Академперіодика, 2007. – 672 с.
10. Савин Р. Тестирование Dot Com / Р. Савин. – Москва: Дело, 2007. – 312 с.
11. Соммервилл И. Инженерия программног обеспечения / И. Соммервилл. – Москва: Издательский дом "Вильямс", 2002. – 624 с. – (6-е издание: Перевод с английского).

12. Тамре Л. Введение в тестирование программного обеспечения / Л. Тамре. – Москва: Издательский дом "Вильямс", 2003. – 368 с. – (Перевод с английского).
13. Шафер Д. Управление программными проектами: достижение оптимального качества при минимуме затрат / Д. Шафер, Р. Фарелл, А. Шафер. – Москва: Издательский дом "Вильямс", 2003. – 1136 с. – (Перевод с английского).

Додаткова:

14. Калбертсон Р. Быстрое тестирование / Р. Калбертсон, К. Браун, Г. Кобб. – Москва: Издательский дом "Вильямс", 2002. – 384 с. – (Перевод с английского).
15. Ковалюк Т. В. Основи програмування / Т. В. Ковалюк. – Київ: Видавнича група ВНУ, 2005. – 384 с.
16. Липаев В. В. Обеспечение качества программных средств. Методы и стандарты / В. В. Липаев. – Москва: СИНТЕГ, 2001. – 380 с.
17. Макгрегор Д. Тестирование объектно-ориентированного программного обеспечения. Практическое пособие / Д. Макгрегор, Д. Сайкс. – Киев: ДиаСофт, 2002. – 432 с.

Інформаційні ресурси

18. Підручник "Програмна інженерія" [Електронний ресурс] – Режим доступу до ресурсу: <http://www.cyb.univ.kiev.ua/library/books/lavrishcheva-6.pdf>.
19. Новости - ITnews [Електронний ресурс] – Режим доступу до ресурсу: <http://itnews.com.ua/news/>.
20. Журнал "Інформаційні технології. Аналітичні матеріали" [Електронний ресурс] – Режим доступу до ресурсу: <http://it.ridne.net/>.

21. MSDN – сеть разработчиков Microsoft [Электронный ресурс] – Режим доступа до ресурсу: <https://msdn.microsoft.com/>.
22. Google App Engine Documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://cloud.google.com/appengine/docs/>.
23. Hibernate. Everything data. - Hibernate [Электронный ресурс] – Режим доступа до ресурсу: <http://hibernate.org/>.
24. Android Developers [Электронный ресурс] – Режим доступа до ресурсу: <https://developer.android.com/index.html>.
25. iXBT.com - оперативные новости, обзоры и тесты смартфонов, планшетов, ноутбуков и проекторов [Электронный ресурс] – Режим доступа до ресурсу: <http://www.ixbt.com/>.

Зразок тестових завдань

1. Тестування сумісності - це:
 - 1) перевірка коректної роботи продукту в певному середовищі;
 - 2) позитивне та негативне тестування;
 - 3) тестування інтерфейсу користувача.
2. Навести схему каскадної моделі ЖЦПЗ (життєвого циклу програмного забезпечення).
3. Моделювання ПЗ (програмного забезпечення) може бути:
 - 1) неформальне;
 - 2) формальне;
 - 3) тестове;
 - 4) вибіркове.

Зразок аналітично-розрахункової задачі.

Задача. *Вихідні дані:*

Керування тестуванням має такі складові:

- планування процесу тестування(складання планів, тестів, наборів даних);
- валідація;
- проведення тестування компонентів повторного використання і патернів як основних об'єктів продукту;
- верифікація;
- генерація необхідних тестових сценаріїв, що відповідають середовищу виконання продукту;
- баг/дефект репорт;
- верифікація правильності реалізації системи і валідація реалізації вимог до готового продукту;
- функціональна придатність;
- збирання даних про відмови, помилки і виявлені непередбачені ситуації при виконанні продукту;
- відповідність стандартам та правилам;
- підготовка звітів за результатами тестування й оцінка характеристик продукту.

Необхідно: визначити лише ті складові, які реально відносяться до керування тестуванням.

Приклад різнорівневого завдання.

Початковий рівень – по 0,5 бала

1. Які основні характеристики програм.
2. Надати класифікацію програмних продуктів, у якій основною ознакою є сфера (область) використання.

Середній рівень – по 0,5 балів.

1. Навести дерево характеристик якості програмних продуктів.
2. Навести поняття «утиліти». Навести приклади утиліт.
3. Що таке ЖЦПЗ?

Достатній рівень – 1 бал

1. Які процеси входять до основних процесів ЖЦПЗ.
2. Навести поняття та означення системи управління базами даних (СУБД), серверу БД, генератору звітів (серверу звітів), текстового процесора, табличного процесора, засобу створення презентаційної графіки, інтегрованих пакетів.
3. Навести основну мету процесу тестування. Яке основне завдання тестування ПЗ? Навести необхідні і достатні умови для проведення тестування.

Високий рівень – по 0,5 балів

1. Перерахувати основні розділи програмної інженерії, фактори еталонної моделі програмної інженерії, типи інформації для економічного аналізу.
2. Які етапи включає в себе процес розроблення ПЗ (програмного забезпечення)?
3. Які задачі включає аналіз проблем і запитів на модифікацію ПЗ, що виконуються службою супроводу. В чому полягають перевірка і приймання. Як виконується зняття ПЗ з експлуатації?

Питання для самоконтролю на тему «Життєвий цикл ПЗ (програмного забезпечення)»:

1. Що таке «життєвий цикл ПЗ (ЖЦПЗ)»?
2. Що в себе включає «розробка ПЗ»?
3. Що в себе включає «експлуатація ПЗ»?
4. Назвати два основних типи моделей ЖЦ (життєвого циклу).
5. Назвати «прогнозовані моделі ЖЦ».
6. Назвати «адаптивні моделі ЖЦ».
7. Навести схеми «каскадної» та «V-подібної» моделей ЖЦ.
8. Що таке «XP-програмування» (eXtreme Programming)?
9. Що таке «адаптивна розробка ПЗ» (Adaptive Software development - ASD)?