

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**Робоча програма
навчальної дисципліни
"ТЕХНОЛОГІЇ СТВОРЕННЯ
ПРОГРАМНИХ ПРОДУКТІВ"
для студентів напряму підготовки
6.050101 "Комп'ютерні науки"
денної форми навчання**

**Харків
ХНЕУ ім. С. Кузнеця
2016**

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 5 від 24.12.2015 р.

Самостійне електронне текстове мережеве видання

Укладач Ю. І. Скорін

Робоча програма навчальної дисципліни "Технології створення Р 58 програмних продуктів" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної форми навчання : [Електронне видання] / уклад. Ю. І. Скорін. – Харків : ХНЕУ ім. С. Кузнеця, 2016. – 73 с.

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами. Вміщено плани лекцій, лабораторних занять, матеріали для закріплення знань (завдання для самостійної роботи, контрольні запитання), критерії оцінювання знань студентів, професійні компетентності, якими повинен володіти студент після вивчення дисципліни.

Рекомендовано для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної форми навчання.

Вступ

Навчальну дисципліну "Технології створення програмних продуктів" віднесено до групи освітньо-професійних дисциплін для студентів напряму підготовки 6.050101 "Комп'ютерні науки".

Дисципліна "Технології створення програмних продуктів" є невід'ємною частиною циклу комп'ютерних дисциплін, необхідних фахівцям-аналітикам з вивчення понять, принципів, методології та технологій створення програмних продуктів (ПП), призначення засобів об'єктного підходу до проектування ПП, стандартів програмної інженерії, методів створення вимог при розробленні ПП, методів та засобів тестування ПП, впровадження стандартів якості при створенні ПП, інструментальних засобів програмної інженерії та їх практичне застосування при проектуванні програмного забезпечення (ПЗ) інформаційних систем (ІС).

Виробництво і використання комп'ютерних програм у даний час є масовою діяльністю. У зв'язку з постійно зростаючими обсягами програмних розробок потрібно готувати фахівців, здатних вирішувати проблеми створення нових програмних продуктів на інженерній основі, використовуючи накопичений запас знань в області програмування та управління системами.

Сформовану структуру та зміст підготовки фахівців треба розширити методами управління, планування і регулювання робіт, адаптуючи їх до умов колективної розробки програмних систем з гарантованою якістю.

Передумовами цього є становлення дисципліни "Технології створення програмних продуктів", яка базується на теорії програмної інженерії або інженерії програмного забезпечення, що увібрала в себе накопичений запас знань в практиці і теорії програмування за останні десятиліття, збагаченою інженерної дисципліною виконання процесів життєвого циклу програмного забезпечення.

У зв'язку з цим предметом вивчення сучасних студентів, майбутніх розробників програмного забезпечення, менеджерів програмних проектів, тестерів, контролерів якості тощо, повинні стати не тільки теоретичні та прикладні методи проектування, а й інженерні методи управління колективом, планування та оцінювання якості виконуваних робіт і укладання в задані терміни і вартість проекту.

Дана дисципліна призначена для оволодіння студентами знань та навичок у сфері програмування модулів інформаційних систем на основі сучасних методології та засобів об'єктно-орієнтованого підходу для проектування інформаційних систем.

Актуальність використання методології визначається необхідністю підвищення рівня ефективності проектних рішень щодо програмних проектів, побудови, експлуатації та супроводу програмного забезпечення інформаційних систем, зокрема тих, які використовуються у стандартах промислових інформаційних систем.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітній ступінь	Характеристика навчальної дисципліни
		денна форма навчання
Кількість кредитів – 4	Галузь знань 0501 "Інформатика і обчислювальна техніка"	за вибором
Змістових модулів – 2	Напрямок підготовки 6.050101 "Комп'ютерні науки"	Рік підготовки
Загальна кількість годин – 144		3-й
		Семестр
		6-й
		Лекції
Тижневих годин для денної форми навчання: аудиторних – 2; самостійної роботи студента – 3	Освітній ступінь: бакалавр	16 год
		Лабораторні
		44 год
		Самостійна робота
		84 год
		Вид контролю
		диференційований залік

Примітка. Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить – 71 %.

2. Мета та завдання навчальної дисципліни

Метою викладання навчальної дисципліни "Технологія створення програмних продуктів" є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методології та технологій розроблення, тестування та експлуатації програмних продуктів, та для моделювання і проектування інформаційних систем, документування та маркетингу програмного забезпечення.

Для досягнення мети поставлені такі основні **завдання**:

вивчення понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем;

вивчення призначення засобів об'єктного підходу до проектування програмного забезпечення інформаційних систем;

вивчення стандартів програмної інженерії;

вивчення методів створення вимог під час розроблення програмних продуктів;

вивчення методів та засобів тестування програмне забезпечення сучасними засобами тестування програмних продуктів;

вивчення стандартів якості під час створення програмного забезпечення інформаційних систем;

вивчення інструментальних засобів програмної інженерії та їх практичне застосування під час проектування програмне забезпечення інформаційних систем.

Об'єктом навчальної дисципліни є процеси, що відбивають різні аспекти створення та супроводу програмних продуктів.

Предметом навчальної дисципліни є технології розроблення, тестування та експлуатації програмних модулів інформаційних систем на основі сучасних методології та засобів об'єктно-орієнтованого підходу для проектування інформаційних систем, вивчення методології щодо розроблення програмного забезпечення інформаційних систем, застосування мови *Universal Modeling Language* (UML) для моделювання і проектування інформаційних систем, застосування програмного інструментарію – IBM Rational Rose, вивчення основних положень з проектування інформаційних систем в економіці на базі сучасних технічних, програмних, інструментальних та комунікаційних засобів.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з літературою з питань базисної технології створення програмних продуктів як сукупності процесів розроблення програмних систем.

Вивчення даної навчальної дисципліни студент розпочинає, прослухавши більшість навчальних дисциплін гуманітарного та професійного циклів. Теоретико-методологічною базою вивчення цієї дисципліни є знання

та навички в галузі аналізу та проектування систем, що отримані при вивченні дисциплін "Системний аналіз та проектування систем обробки інформації", "Основи баз даних та знань", "CASE-технології", "Проектування інформаційних систем", "Технології комп'ютерного проектування".

У свою чергу, знання з даної дисципліни забезпечують успішне виконання комплексних курсових робіт, бакалаврських та магістерських дипломних робіт тощо.

У результаті вивчення навчальної дисципліни студент повинен:

знати:

етапи життєвого циклу програмного продукту;

як аналізувати, створювати та обирати склад та типи вимог до програмного продукту;

сучасні процеси проектування та розроблення програмних продуктів;

принципи управління якістю програмних продуктів;

методи тестування програмного продукту;

методології та засоби програмної інженерії;

методи моделювання для опису об'єктів інформатизації;

основні підходи та поняття до об'єктно-орієнтованого проектування інформаційних систем;

як ідентифікувати об'єкти системи, що проектується;

методи спостереження, опису об'єктів проектування;

сучасні підходи до маркетингу програмних продуктів.

вміти:

аналізувати предметну область на основі об'єктно-орієнтованої методології проектування;

застосовувати основні методи та інструменти розроблення програмних продуктів;

будувати діаграми у мові UML для формалізації опису предметної області, для якої розроблюється програмний продукт;

складати специфікації щодо вимог різних рівнів (вимоги замовника та розробника) на основі аналізу предметної області з використанням стандартної специфікації вимог до програмного продукту, що розробляється;

проводити порівняльний аналіз процесів проектування і розробки програмних продуктів і робити обґрунтований вибір;

виконувати формування та аналіз вимог для розроблення програмних продуктів;

виконувати тестування програмного продукту;

розробляти документацію, необхідну для тестування програмного продукту;

працювати з замовником для виявлення вимог до програмного продукту;

складати технічне завдання на розробку програмного продукту;

застосовувати методи моделювання для опису об'єктів інформатизації;

аналізувати та моделювати бізнес-процеси, будувати регламенти зі створення комп'ютеризованих бізнес-процесів;

ідентифікувати об'єкти системи, що проектується;

розробляти діаграми динамічних та статичних аспектів інформаційної системи;

розрізняти і використовувати стандарти групи IDEF;

будувати діаграми потоків даних;

розробляти діаграми взаємодії об'єктів інформаційної системи;

розробляти технічну документацію на програмне забезпечення;

застосовувати сучасні підходи до маркетингу програмних продуктів.

У процесі викладання навчальної дисципліни основна увага приділяється оволодінню студентами професійними компетентностями, що наведені в табл. 2.1.

Структуру складових професійних компетентностей та їх формування відповідно до Національної рамки кваліфікацій України наведено в табл. А.1 додатка А.

Таблиця 2.1

Професійні компетентності, які отримують студенти після вивчення навчальної дисципліни

Код компетентності	Назва компетентності	Складові компетентності
1	2	3
ТСПП* 1	Використовувати теоретичні основи сучасних методологій та технологій розроблення, тестування та експлуатації ПП як сукупності процесів розроблення програмних систем на засадах життєвого циклу ПЗ	Знати етапи життєвого циклу програмного продукту
		Аналізувати, створювати та обирати склад та типи вимог до програмного продукту
		Знати сучасні процеси проектування та розроблення програмних продуктів

1	2	3
ТСПП 2	Реалізовувати принципи управління якістю програмних продуктів використовуючи сучасні методи тестування програмних продуктів	Знати теоретичні основи управління якістю програмних продуктів
		Знати принципи управління якістю програмних продуктів
		Застосовувати сучасні методи тестування програмного продукту
ТСПП 3	Здійснювати реалізацію основних підходів до об'єктно-орієнтованого проектування інформаційних систем	Знати та застосовувати методології та засоби програмної інженерії
		Знати та застосовувати методи моделювання для опису об'єктів інформатизації
		Знати та застосовувати основні підходи до об'єктно-орієнтованого проектування інформаційних систем

* Технології створення програмних продуктів.

3. Програма навчальної дисципліни

Змістовий модуль 1

Загальні питання та методи ТСПП

Тема 1. Базові поняття створення програмних продуктів

1.1. Вступ.

Предмет, мета та завдання навчальної дисципліни, її місце у навчальному процесі. Організаційно-методичні вказівки щодо вивчення дисципліни.

1.2. *Поняття моделі життєвого циклу програмного забезпечення та інформаційної системи.*

Моделі життєвого циклу життєвого циклу програмного забезпечення та інформаційної системи.

1.3. *Основні етапи життєвого циклу інформаційної системи – аналіз, розроблення вимог, проектування, тестування, ввід у експлуатацію, супровід.*

Етапи життєвого циклу, а саме: аналіз, розроблення вимог, проектування, тестування, ввід у експлуатацію, супровід. Склад завдань за етапами.

1.4. Причини виникнення технології програмної інженерії.

Причини виникнення технології програмної інженерії: повторне використання коду, модульне програмування, модифікація програм. Історична довідка.

1.5. Структура життєвого циклу за міжнародним стандартом ISO/IEC 12207.

Основний зміст та галузь застосування міжнародного стандарту ISO/IEC 12207.

Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення

2.1. Зміст основних етапів розроблення програмного забезпечення.

Зміст основних етапів розроблення програмного забезпечення: визначення вимог, проектування, кодування, тестування, супровід.

2.2. Особливості каскадної, спіральної, інкрементальної моделей.

Особливості каскадної, спіральної, інкрементальної моделей. Моделі швидкого розроблення програмних продуктів.

2.3. Зміст процесу розроблення програмного забезпечення.

Уніфікований процес розроблення програмного забезпечення (USDP).

2.4. Основні стандарти програмної інженерії.

ISO/IEC 12207 Information Technology Software Life Cycle Processes, SEI CMM Capability Maturity Model (for Software), ISO/IEC 15504 Software Process Assessment, PMBOK Project Management Body of Knowledge, SWEBOK Software Engineering Body of Knowledge, ACM/IEEE CC2001 Computing Curricula 2001. Стандарт ISO/IEC 12207-95. Міжнародні організації-розробники стандартів.

Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення

3.1. Принципи і методи проектування програмного забезпечення.

Принципи і методи проектування програмного забезпечення. Основні поняття програмного процесу: життєвий цикл, процес, основні процеси. Сутність моделі програмного процесу, моделі організації робіт.

3.2. Методи програмної інженерії.

Методи програмної інженерії: опис моделей, правила та обмеження, рекомендації по застосуванню методу. Типова схема розроблення програми.

3.3. Рівні управління вимогами.

Рівні управління вимогами: рівень замовника та розробника.

3.4. Особливості застосування та формулювання вимог.

Поняття С-вимоги та D-вимоги. Особливості застосування та формулювання.

Тема 4. Типова схема аналізу вимог

4.1. Джерела виникнення С-вимог.

Джерела виникнення С-вимог. Визначення складу осіб, які впливають на вимоги.

4.2. Методології та інструментальні засоби для С-вимог.

Методології та інструментальні засоби для С-вимог. Використання діаграм варіантів використання, класів та переходів станів для С-вимог.

4.3. Типова схема розроблення D-вимог.

Типова схема розроблення D-вимог та їх зв'язок із С-вимогами. Метрики визначення та оцінювання D-вимог. Склад D-вимог. Послідовність дій для збирання й документування D-вимог.

4.4. Типи D-вимог.

Типи D-вимог: функціональні, нефункціональні, зворотні. Сутність типів вимог. Метрики оцінювання. Метрики аналізу вимог згідно зі стандартом IEEE 982.2-1988. Властивості D-вимог. Організація D-вимог. Засоби опису детальної вимоги.

Змістовий модуль 2

Реалізація, тестування та супровід ПЗ

Тема 5. Якість програмного забезпечення

5.1. Поняття і складові якості програмного забезпечення.

Поняття якості програмного забезпечення. Основні аспекти якості ПЗ, рівні уявлення моделі якості ПЗ. Складові якості програмного забезпечення: надійність, супровід, практичність, ефективність, мобільність, функціональність.

5.2. Метрики якості програмного забезпечення.

Поняття метрики якості програмного забезпечення. Організація контролю якості ПЗ на етапах життєвого циклу програмного продукту.

5.3. Стандартне оцінювання значень показників якості.

Чотирьохрівнева модель якості ПЗ. Типи заходів. Стандарт, використовуваний для оцінювання значень показників якості. Методи оцінювання значень показників якості: вимірjuвальний, реєстраційний, розрахунковий та експертний, а також їх комбінації.

Тема 6. Моделі якості та надійності в програмній інженерії

6.1. Управління якістю програмного забезпечення.

Мета процесу управління якістю ПЗ. Два процеси забезпечення якості на етапах ЖЦ ПЗ. Вимоги стандарту до організації системи якості.

6.2. Моделі оцінки надійності програмного забезпечення.

Поняття надійності ПЗ. Основні поняття в проблематиці надійності ПЗ. Класифікація моделей надійності ПЗ. Моделі надійності в програмній інженерії.

6.3. Марківські і пуассонівська моделі надійності.

Марківський процес. Моделі зростання надійності ПЗ. Модель Джелінського-Моранді. Модель Шика-Вулвертон. Модель Гоело-Окумото.

Тема 7. Тестування програмного забезпечення

7.1. Мета тестування та основні принципи тестування.

Мета тестування. Основні принципи тестування: динамічність, кінцевість, вибір, очікувана поведінка. Область знань "Тестування програмного забезпечення" (відповідно до стандарту SWEBOOK, 2004).

7.2. Рівні тестування.

Рівні тестування. Модульне та системне тестування.

7.3. Типовий план модульного тестування.

Типовий план модульного тестування. Класифікація типів тестів: чорний, сірий, білий ящики. Тестування "ящиків", аналіз межових значень для "чорного" ящика. Розгляд рішень для тестування "білого ящика".

7.4. Методи планування тестування.

Методи планування тестування. Стандарти документації з тестування.

Тема 8. Супровід програмного забезпечення

8.1. Сутність та методи супроводу програмних систем.

Супровід програмних систем: мета і основні завдання. Організація процесу супроводу. Методи супроводу: аналіз впливу факторів, зворотне проектування.

8.2. Завдання супроводу програмних систем.

Визначення завдань супроводу згідно зі стандартом IEEE 1219-1992.

8.3. Якість супроводу програмних систем.

Визначення якості супроводу: метрики супроводу та їх використання.

8.4. Управління супроводом програмних систем.

Принципи управління супроводом програмних систем.

4. Структура навчальної дисципліни

Із самого початку вивчення навчальної дисципліни кожен студент має бути ознайомлений як з робочою програмою навчальної дисципліни і формами організації навчання, так і зі структурою, змістом та обсягом кожної з тем, а також з усіма видами контролю та методикою оцінювання сформованих професійних компетентностей.

Вивчення студентом навчальної дисципліни відбувається шляхом послідовного і ґрунтовного опрацювання тем дисципліни. Структура залікового кредиту навчальної дисципліни наведена в табл. 4.1.

Таблиця 4.1

Структура залікового кредиту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин			
	денна форма			
	Усього	у тому числі		
		Лекційні	Лабораторні	самостійна робота
			Підготовка до занять	
1	2	3	4	5
Змістовий модуль 1. Загальні питання та методи ТСПП				
Тема 1. Базові поняття створення програмних продуктів	19	2	6	11

1	2	3	4	5
<i>Тема 2.</i> Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення	19	2	6	11
<i>Тема 3.</i> Класифікація та сутність процесів розроблення програмного забезпечення	19	2	6	11
<i>Тема 4.</i> Типова схема аналізу вимог	19	2	6	11
Разом за змістовим модулем 1	76	8	24	44
Змістовий модуль 2. Реалізація, тестування та супровід ПЗ				
<i>Тема 5.</i> Якість програмного забезпечення	19	2	6	11
<i>Тема 6.</i> Моделі якості та надійності в програмній інженерії	19	2	6	11
<i>Тема 7.</i> Тестування програмного забезпечення	19	2	6	11
<i>Тема 8.</i> Супровід програмного забезпечення	11	2	2	7
Разом за змістовим модулем 2	68	8	20	40
Усього годин	144	16	44	84

5. Теми лабораторних занять

Лабораторне заняття – форма навчального заняття, за якої студент під керівництвом викладача особисто проводить імітаційні експерименти чи досліди з метою практичного підтвердження окремих теоретичних положень навчальної дисципліни. У ході лабораторних робіт студент набуває професійних компетентностей та практичних навичок роботи з комп'ютерним обладнанням, відповідними програмними продуктами. За результатами виконання завдання на лабораторному занятті студенти оформляють індивідуальні звіти про його виконання та захищають ці звіти перед викладачем (табл. 5.1).

У даний час сфера інформаційних технологій характеризується в основному промисловою розробкою складних програмних комплексів і систем, причому великими територіально розподіленими колективами [21; 41].

Навчання технологіям створення програмних продуктів в рамках традиційної освітньої технології не сприяє формуванню у студентів компетенцій, необхідних для роботи на великих проектах, де поряд з навичками самостійної розробки програм необхідно вміння працювати в команді, володіння засобами підтримки такої роботи і наявність навичок програмування.

Таблиця 5.1

Перелік тем лабораторних занять

Програмні питання	Кількість годин	Література
<i>Лабораторна робота 1.</i> Дослідження методики розробки специфікації вимог до програмного продукту	8	Основна: [2; 4; 7; 9; 12]. Додаткова: [13 – 50]
<i>Лабораторна робота 2.</i> Дослідження основних елементів мови програмування <i>Ruby</i>	10	Основна: [6; 8; 10; 12]. Додаткова: [45 – 49]
<i>Лабораторна робота 3.</i> Ознайомлення з фреймворком <i>Rails</i> . Аналіз систем контролю версій, основи роботи з GIT	8	Основна: [1; 6; 8; 10; 12]. Додаткова: [36 – 54]
<i>Лабораторна робота 4.</i> Розроблення додатку відповідно до індивідуального завдання, аналіз підходів до реалізації аутентифікації та авторизації	10	Основна: [1; 5; 9; 11]. Додаткова: [23 – 52]
<i>Лабораторна робота 5.</i> Дослідження основних можливостей фреймворку <i>Twitter Bootstrap</i> , основи роботи з мовою програмування <i>JavaScript</i> та фреймворком <i>JQuery</i>	8	Основна: [1; 3; 10; 12]. Додаткова: [29 – 55]
Усього годин	44	

Підготовка професіоналів для роботи у великих ІТ компаніях потребує принципово інших методах, в порівнянні з традиційними, необхідність їх розробки ставати все більш гострою.

Для вирішення означеної проблеми доцільно використовувати при проведенні лабораторних занять, які застосовані в промисловій розробці програмного забезпечення та асоційовані в даний час з поняттям "програмна інженерія".

У даному підході застосовуються активні форми навчання, а саме імітується процес розробки програмного забезпечення, прийнятий в ІТ галузі, тим самим процес навчання наближається до умов реального ринку праці.

Для опису даної моделі процесу навчання необхідно ввести такі поняття.

1. Учасник процесу навчання. Учасниками процесу навчання є студент і викладач.

2. Роль. Під роллю розуміється сукупність дій і функцій, виконуваних учасниками процесу. Студент і викладач в процесі навчання з використанням методів програмної інженерії можуть перебувати в різних ролях, причому одна і та ж роль може виконуватися як студентом, так і викладачем.

3. Робочий продукт. Результат роботи учасника процесу в тій чи іншій ролі.

Враховуючи, що основним об'єктом в даній моделі є завдання щодо створення програмного продукту, над якою йде колективна робота, необхідно ввести наступні ролі учасників цього колективного освітнього процесу.

1. Автор. У завдання автора входить формулювання завдання для розробки студентом.

2. Модератор. У завдання модератора входить призначення завдання на виконавців, контроль процесу роботи над завданням і винесення остаточного рішення про результат її вирішення.

3. Кодувальник. Завдання кодувальника – написання програми для вирішення призначеної йому завдання.

4. Інспектор. Інспектор займається аналізом коду розробленої програми на предмет її коректності та якості кодування, адекватності застосовуваних програмних рішень. Інспектор виносить один з трьох можливих вердиктів:

а) прийняти задачу без зауважень. Це означає, що інспектору абсолютно зрозумілі всі програмні рішення, код написаний правильно, текст програми забезпечений всіма необхідними коментарями;

б) прийняти задачу за умови усунення зауважень. Це означає, що програма в цілому написана вірно, однак деякі моменти представляються незрозумілими або некоректними. Інспектор формулює свої зауваження і перевіряє завдання знову після їх усунення кодувальником. Процес усунення зауважень триває до тих пір, поки інспектор не винесе рішення "Прийняти без зауважень";

в) відхилити задачу. Це означає, що програма не виконує необхідних функцій або кількість зауважень настільки велике, що вона не може бути прийнята без суттєвого доопрацювання.

5. Тестер. У завдання тестера входить розроблення тесту, який дозволить перевірити якість написання програми шляхом пошуку в ній помилок і власне перевірка. Робота тестера починається після того, як інспектор виніс вердикт "Прийняти без зауважень". При цьому тестер не повинен аналізувати код програми, він лише повинен запускати її і проганяти на сценаріях, адекватних умовою задачі.

Робочими продуктами розглянутої моделі процесу навчання є:

у автора формулювання завдання;

у модератора оцінювання роботи виконавців;

у кодувальника код програми (текст);

у інспектора зауваження і набір рекомендацій з доопрацювання коду;

у тестера сценарій тестів.

Послідовність виконання ролей в процесі роботи над програмою:

автор формулює завдання, для вирішення якої необхідно написати програму і передає її кодувальнику;

модератор призначає ролі;

кодувальник пише програму і передає її інспектору;

інспектор аналізує програмний код і виносить своє рішення. Якщо код приймається без зауважень, то програмний продукт передається тестеру. Якщо у інспектора є зауваження, він повідомляє їх кодувальнику. Далі відбувається взаємодія кодувальника з інспектором з усунення зауважень до винесення рішення "Прийняти без зауважень";

тестер розробляє сценарій і перевіряє якість програмного продукту;

модератор виносить остаточне рішення про якість програмного продукту і оцінює результати роботи.

Можливо кілька варіантів розподілу ролей між учасниками процесу. У табл. 5.2 показано розподіл в рамках традиційних форм процесу навчання.

Таблиця 5.2

Розподіл ролей між учасниками освітнього процесу за умови традиційної форми навчання

Роль	Учасник	Функція
Автор	Викладач	Формулювання завдання
Модератор	Викладач	Розподіл завдань між студентами (видача завдань)
Кодувальник	Студент	Написання програмного коду
Інспектор	Викладач	Перевірка коду програми, оцінювання роботи студента
Тестер	Викладач	Запуск програми і її тестування, оцінювання роботи студента

У традиційному варіанті ролі автора, модератора, інспектора та тестера виконує викладач, а студент виконує лише роль кодувальника. Застосування такого підходу в даний час виправдане лише для навчання студентів непрофільних, не пов'язаних безпосередньо з інформаційними технологіями, напрямків підготовки студентів.

У табл. 5.3 показано розподіл ролей для першого варіанту пропонованої технології навчання.

Таблиця 5.3

**Розподіл ролей між учасниками освітнього процесу
пропонована технологія навчання з використанням методів
програмної інженерії (варіант 1)**

Роль	Учасник	Функція
Автор	Викладач	Формулювання завдання
Модератор	Викладач	Розподіл завдань між студентами, визначення ролі кожного зі студентів, оцінювання роботи студента
Кодувальник	Студент 1	Написання програмного коду
Інспектор	Студент 2	Перевірка коду програми, формулювання зауважень з поліпшення і доопрацювання
Тестер	Студент 3	Розроблення сценаріїв тестування, запуск програми і її тестування

У даному варіанті викладач виконує ролі автора і модератора, а ролі кодувальника, інспектора та тестера виконують студенти. Викладач формулює завдання на ту чи іншу тему, розподіляє ролі між студентами, контролює процес їх роботи і оцінює результат. Студент-кодувальник пише програмний код і передає його студенту-інспектору. Інспектор аналізує представлений код, формулює свої зауваження та (при необхідності) повертає його кодувальнику на доопрацювання. У разі відсутності зауважень програма передається студенту-тестеру. Тестер розробляє сценарій тесту і "проганяє" через нього програму з формулюванням виведення типу "пройдено або не пройдено" по кожному пункту сценарію. По закінченні роботи тестера викладач виносить оцінку кожному учаснику.

У тому випадку, якщо для групи студентів реалізується поглиблене вивчення програмування, можливий другий варіант розподілу ролей учасників, поданий в табл. 5.4. У даному варіанті роль автора виконують студенти.

**Розподіл ролей між учасниками освітнього процесу
пропонована технологія навчання з використанням методів
програмної інженерії (варіант 2)**

Роль	Учасник	Функція
Автор	Студент 1	Формулювання завдання
Модератор	Викладач	Визначення ролі кожного зі студентів, розподіл завдань між студентами, оцінювання роботи студента
Кодувальник	Студент 2	Написання програмного коду
Інспектор	Студент 3	Перевірка коду програми, формулювання зауважень з поліпшення і доопрацювання
Тестер	Студент 4	Розроблення сценаріїв тестування, запуск програми і її тестування

Застосування описаної технології в процесі навчання дисципліни "Технології створення програмних продуктів" дозволяє підвищити ефективність навчального процесу з точки зору придбання студентами компетенцій в області створення програмного забезпечення. Крім цього, її використання дозволяє підвищити мотивацію студентів до вивчення дисципліни, познайомити їх з прийомами колективної роботи, застосовуваними в професійному середовищі і забезпечити прозорість виставлення оцінок з дисципліни.

6. Самостійна робота

Самостійна робота студента (СРС) – це форма організації навчального процесу, за якої заплановані завдання виконуються студентом самостійно під методичним керівництвом викладача.

Мета СРС – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

Навчальний час, відведений для самостійної роботи студентів денної форми навчання, визначається навчальним планом і становить 58 % (84 години) від загального обсягу навчального часу на вивчення дисципліни (144 години).

У ході самостійної роботи студент має перетворитися на активного учасника навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними і практичними знаннями, вільно орієнтуватися в інформаційному просторі, нести індивідуальну відповідальність за якість власної професійної підготовки.

СРС включає:

- опрацювання лекційного матеріалу;
- опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами дисципліни;
- підготовку до лабораторних занять;
- поглиблене опрацювання окремих лекційних тем або питань;
- пошук (підбір) та огляд літературних джерел за заданою проблематикою дисципліни;
- аналітичний розгляд наукової публікації;
- контрольну перевірку студентами особистих знань за запитаннями для самодіагностики;
- підготовку до контрольних робіт та інших форм поточного контролю;
- систематизацію вивченого матеріалу з метою підготовки до заліку.

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною економічною літературою, нормативними актами з питань державного регулювання економіки, статистичними матеріалами. Основні види самостійної роботи, які запропоновані студентам для засвоєння теоретичних знань з навчальної дисципліни, наведені в табл. 6.1.

Таблиця 6.1

Завдання для самостійної роботи студентів та форми її контролю

Назва теми	Зміст самостійної роботи студентів	Кількість годин	Форми контролю СРС	Література
1	2	3	4	5
Змістовий модуль 1. Загальні питання та методи ТСПП				
<i>Тема 1.</i> Базові поняття створення програмних продуктів	Вивчення лекційного матеріалу; огляд теоретичного матеріалу та підготовка доповіді "SWEBOOK – основоположним документ, який відображає думку зарубіжних і вітчизняних фахівців у галузі програмної інженерії"	11	Презентація результатів	Основна: [2; 5; 7; 8; 10; 12]. Додаткова: [22 – 55]

Продовження табл. 6.1

1	2	3	4	5
<i>Тема 2.</i> Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення	Вивчення лекційного матеріалу; підготовка до лабораторного заняття, огляд теоретичного матеріалу та підготовка доповіді "Структура життєвого циклу відповідно до стандарту ISO / IEC 12207"	11	Презентація результатів, захист лабораторної роботи	Основна: [2; 4; 7; 9; 12]. Додаткова: [13 – 55]
<i>Тема 3.</i> Класифікація та сутність процесів розроблення програмного забезпечення	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Специфіка розробки програмного забезпечення"	11	Презентація результатів	Основна: [6; 8; 10; 12]. Додаткова: [43 – 50]
<i>Тема 4.</i> Типова схема аналізу вимог	Вивчення лекційного матеріалу, підготовка до лабораторного заняття, огляд теоретичного матеріалу та підготовка доповіді "Визначення вимог до програмного продукту"	11	Презентація результатів, захист лабораторної роботи	Основна: [1; 6; 8; 10; 12]. Додаткова: [34 – 53]
Змістовий модуль 2. Реалізація, тестування та супровід ПЗ				
<i>Тема 5.</i> Якість програмного забезпечення	Вивчення лекційного матеріалу, підготовка до лабораторного заняття, підготовка до контрольної роботи, огляд теоретичного матеріалу та підготовка доповіді "Описання якості програмних продуктів на основі критеріїв відбору необхідних властивостей"	11	Презентація результатів, захист лабораторної роботи, контрольна робота за темами 1 – 4	Основна: [1; 5; 7; 9; 11]. Додаткова: [13 – 55]
<i>Тема 6.</i> Моделі якості та надійності в програмній інженерії	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Управління якістю програмного продукту"	11	Презентація результатів	Основна: [6; 9; 10; 12]. Додаткова: [45 – 54]
<i>Тема 7.</i> Тестування програмного забезпечення	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Планування робіт на етапі тестування програмного продукту"	11	Презентація результатів	Основна: [1; 6; 8; 10; 12]. Додаткова: [34 – 54]

1	2	3	4	5
Тема 8. Супровід програмного забезпечення	Вивчення лекційного матеріалу, огляд теоретичного матеріалу та підготовка доповіді "Види діяльності на етапі супроводу та експлуатації програмного продукту"	7	Презентація результатів	Основна: [1; 5; 9; 12]. Додаткова: [25 – 53]

6.1. Контрольні запитання для самодіагностики

Змістовий модуль 1 Загальні питання та методи ТСПП

Тема 1. Базові поняття створення програмних продуктів

1. Поясніть мотивацію до вивчення дисципліни. Ви згодні з такою мотивацією?
2. Назвіть цілі та завдання дисципліни.
3. Назвіть базові процеси, без реалізації яких не може обійтися жодна технологія розроблення програмного продукту.
4. Поясніть мету та зміст розробки специфікації програмного продукту.
5. Поясніть мету та зміст проектування та реалізації програмного продукту.
6. Поясніть мету та зміст атестації програмного продукту.
7. Поясніть мету та зміст еволюції програмного продукту.
8. Поясніть шляхи удосконалення процесу створення програмного продукту.
9. В чому полягає специфіка та особливості створення програмного продукту?
10. Розкрийте зміст поняття "життєвий цикл".
11. В чому полягає структура життєвого циклу програмного продукту відповідно до стандарту ISO / IEC 12207?
12. Назвіть стадії життєвого циклу програмного продукту.
13. Назвіть причини обмеженого застосування автоматизованих засобів пов'язаних з розробкою програмних продуктів.

14. Назвіть фундаментальні базові процеси, без реалізації яких не може обійтися жодна технологія розробки програмного продукту.
15. Назвіть специфічні особливості притаманні процесу розроблення програмного продукту.
16. Що розуміється під моделлю життєвого циклу програмного продукту?
17. Назвіть найпоширеніші моделі життєвого циклу програмного продукту.
18. Назвіть переваги та недоліки каскадної моделі.
19. Назвіть переваги та недоліки спіральної моделі.
20. Охарактеризуйте уніфікований процес розроблення програмного продукту.
21. Назвіть ознаки зрілої професії. Які з них притаманні програмній інженерії.
22. Назвіть галузі знань SWEBOOK інженерії розроблення програмного продукту.
23. Наведіть базові поняття SWEBOOK.
24. Визначте цілі і завдання галузі інженерії управління проектом.
25. Визначте цілі і завдання галузі інженерії управління якістю.

Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення

1. Розкрийте зміст поняття "модель життєвого циклу програмного засобу".
2. Розкрийте зміст понять "каскадна модель (модель водоспаду) *Cascade Model*" життєвого циклу програмного продукту.
3. Розкрийте зміст поняття "спіральна модель *Spiral Model*" життєвого циклу програмного продукту.
4. Назвіть основні переваги застосування каскадної моделі життєвого циклу програмного продукту.
5. Назвіть основні недоліки застосування каскадної моделі життєвого циклу програмного продукту.
6. Назвіть основні переваги застосування спіральної моделі життєвого циклу програмного продукту.
7. Охарактеризуйте поняття моделі життєвого циклу і назвіть їх види.
8. Дайте характеристику каскадної моделі.

9. Визначте відмінну особливість спіральної моделі життєвого циклу.
10. Які загальні риси мають інкрементна і еволюційна моделі?
11. Дайте перелік процесів життєвого циклу стандарту і назвіть їх призначення.
12. Як побудувати нову модель життєвого циклу на основі стандарту?
13. Дайте класифікацію процесів життєвого циклу стандарту.
14. Назвіть процеси управління проектом.
15. Назвіть процеси управління якістю.
16. Проведіть порівняльне оцінювання моделі процесів життєвого циклу стандарту 12207 і областей-процесів ядра знань SWEBOOK.
17. Охарактеризуйте циклічний характер розробки програмного продукту.

Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення

1. Які поняття включає предмет технології розробки програмного забезпечення згідно RUP?
2. Розкрийте зміст поняття "персонал".
3. Розкрийте зміст поняття "проект".
4. Розкрийте зміст поняття "продукт".
5. Розкрийте зміст поняття "процес".
6. Розкрийте зміст поняття "утиліти".
7. Наведіть і поясніть основні завдання RUP.
8. Наведіть і поясніть відмінні риси RUP.
9. Поясніть зміст поняття "життєвий цикл RUP".
10. Дайте характеристику специфікації компонента.
11. Визначте мовні засоби опису компонентів.
12. Назвіть об'єкти мови JAVA.
13. Визначте методи інтеграції об'єктів мови JAVA.
14. Визначте основні характеристики об'єктів у системі CORBA.
15. Наведіть структуру опису специфікації інтерфейсу в мові IDL.
16. Розкажіть про особливості опису об'єктів в системі COM.
17. Проведіть порівняння засобів CORBA і COM.
18. Для яких цілей створено Rational Rose.
19. Назвіть інструменти Rational Rose якими Ви користувалися.
20. Дайте перелік діаграм мови моделювання UML.

21. Визначте процес розроблення ПС за допомогою UML.
22. Для яких цілей розроблено метод MSF?
23. Назвіть основні моделі MSF.
24. Як вирішуються питання управління проектом і ризиками в системі MSF.

Тема 4. Типова схема аналізу вимог

1. Розкрийте зміст поняття "програмні вимоги (*Software Requirements*)".
2. Розкрийте зміст поняття "вимоги до продукту та процесу (*Product and Process Requirements*)".
3. Розкрийте зміст понять "функціональні та нефункціональні вимоги (*Functional and Non-functional Requirements*)".
4. Розкрийте зміст поняття "бізнес-вимоги (*Business Requirements*)".
5. Розкрийте зміст поняття "користувальницькі вимоги (*User Requirements*)".
6. Розкрити зміст поняття "група нефункціональних вимог (*Non-Functional Requirements*)".
7. Розкрийте зміст поняття "бізнес-правила (*Business Rules*)".
8. Розкрийте зміст поняття "обмеження (*Constraints*)".
9. Розкрийте зміст поняття "системні вимоги (*System Requirements*)".
10. Розкрийте зміст поняття "незалежні властивості (*Emergent Properties*)".
11. Розкрийте зміст поняття "вимоги з кількісною оцінкою (*Quantifiable Requirements*)".
12. Розкрийте зміст поняття "системні вимоги та програмні вимоги (*System Requirements and Software Requirements*)".
13. Поясніть процес роботи з вимогами (*Requirements Process*).
14. Розкрийте зміст поняття "модель процесу визначення вимог".
15. Поясніть техніку витягу вимог (*Elicitation Techniques*).
16. Розкрийте зміст поняття "аналіз вимог (*Requirements Analysis*)".
17. Розкрийте зміст поняття "класифікація вимог (*Requirements Classification*)".
18. Розкрийте зміст поняття "архітектурне проектування та розподіл вимог (*Architectural Design and Requirements Allocation*)".
19. Розкрийте зміст поняття "специфікація вимог (*Requirements Specification*)".

20. Як називається етап ЖЦ розроблення ПЗ, на якому фіксується контракт між замовником і виконавцем розробки?
21. Назвіть дійових осіб процесу формування вимог.
22. Назвіть джерела відомостей про вимоги.
23. Яка послідовність кроків по використанню діючої системи в новій розробці?
24. Назвіть категорії класифікації вимог.
25. Назвіть цілі і складові концептуального моделювання проблеми.
26. Що визначає онтологія концептуального моделювання проблеми?
27. Поясніть суть відносин, за допомогою яких будуються поняття: узагальнення, декомпозиція, абстракція, асоціація.
28. Назвіть елементи об'єктно-орієнтованого моделювання програмних систем.
29. У чому полягає принцип приховування інформації?
30. Визначте концепцію моделі сценаріїв для збору вимог.
31. Дайте пояснення для нотації діаграми сценаріїв і базових відносин у них.
32. Назвіть основні типи об'єкти моделі.
33. Наведіть завдання трасування вимог.
34. Розкажіть про принципи взаємин між замовником і розробником вимог до системи.

Змістовий модуль 2

Реалізація, тестування та супровід ПЗ

Тема 5. Якість програмного забезпечення

1. Розкрийте зміст поняття "якість ПП".
2. Назвіть і поясніть основні критерії якості ПП.
3. Розкрийте зміст поняття "функціональна придатність".
4. Розкрийте зміст поняття "надійність".
6. Розкрийте зміст поняття "застосовність".
7. Розкрийте зміст поняття "ефективність".
8. Розкрийте зміст поняття "супровід".
9. Розкрийте зміст поняття "мобільність".

10. Назвіть систему стандартів, які регламентують питання управління якістю.
11. Поясніть мету стандарту ISO 9001.
12. Назвіть мінімальний набір вимог до управління якістю.
13. Розкрийте зміст поняття "вимоги до менеджменту компанії".
14. Розкрийте зміст поняття "вимоги до контролю продукції".
15. Розкрийте зміст поняття "вимоги до процесу розроблення".
16. Назвіть основні аспекти та рівні моделі якості ПЗ.
17. Визначте характеристики якості ПЗ і їх призначення.
18. Які методи використовуються при визначенні показників якості?
19. Визначте метрики програмного продукту та їх складові.
20. Які стандарти в області якості ПЗ існують?
21. Назвіть основні цілі та завдання системи управління якістю.
22. У чому сутність інженерії якості?
23. Назвіть зміст класифікації моделей надійності.
24. Визначте типи моделей надійності та їх базис.
25. Які дані необхідні для оцінювання надійності ПЗ?

Тема 6. Моделі якості та надійності в програмній інженерії

1. Розкрийте зміст та сутність процесу реалізації модулів.
2. Визначте поняття реалізація модулів.
3. Поясніть схему процесу реалізації модулів.
4. Наведіть типову схему процесу реалізації модулів.
5. Розкрийте зміст поняття "якість реалізації".
6. Розкрийте зміст понять "стандартні метрики коду".
7. Розкрийте зміст понять "індивідуальні метрики коду".
8. Розкрийте зміст понять "інспектування коду".
9. Розкрийте зміст поняття "індивідуальна програмна документація".
10. Як вирішуються завдання менеджменту програмного проекту?
11. Визначте процес планування менеджменту проекту.
12. Визначте поняття управління ризиком.
13. Що розуміється під процесом управління конфігурацією ПЗ?
14. Наведіть основні завдання управління конфігурацією.
15. Дайте загальну характеристику понять ідентифікації, урахування статусу.
16. Які дії виконуються в процесі управління версіями ПЗ?

Тема 7. Тестування програмного забезпечення

1. Що означає повністю протестувати програму?
2. Назвіть причини, за якими повне тестування не може бути виконано ніколи.
3. Поясніть вплив причини "занадто велика кількість всіх можливих комбінацій вхідних даних".
4. Поясніть вплив причини "занадто велика кількість усіх можливих послідовностей виконання коду програми".
5. Поясніть вплив причини "інтерфейс користувача, що включає всі можливі комбінації дій користувача і його переміщень по програмі, є складним для повного тестування".
6. Розкрийте зміст поняття "визначення вимог".
7. Розкрийте зміст поняття "аналіз".
8. Розкрийте зміст поняття "проекування".
9. Розкрийте зміст поняття "реалізація".
10. Розкрийте зміст поняття "тестування".
11. Розкрийте зміст поняття "дослідна експлуатація".
12. Розкрийте зміст поняття "остаточна приймання і сертифікація".
13. Розкрийте зміст поняття "супровід".
14. Наведіть і поясніть принцип прийняття рішень на підставі критерію Вальда.
15. Наведіть і поясніть принцип прийняття рішень на підставі критерію Севіджа.
16. Наведіть і поясніть принцип прийняття рішень на підставі критерію Гурвіца.
17. Поясніть мету тестування ПП.
18. Поясніть основні принципи тестування: динамічність, завершеність, вибір, очікувана поведінка.
19. Назвіть рівні тестування.
20. Поясніть типовий план модульного тестування.
21. Назвіть методи планування тестування.
22. Назвіть шляхи подальшого розвитку методів та засобів тестування.
23. У чому полягає багаторівнева модель якості програмного забезпечення?
24. Назвіть складові якості програмного забезпечення.
25. Вкажіть хто з членів групи тестування відповідає за розробку тестових специфікацій та аналіз результатів тестування?

26. Вкажіть, хто з членів групи тестування відповідає за організацію процесу тестування?
27. Який з критеріїв тестування вказується в описі вимог до програмного продукту?
28. У чому полягає особливість тестування на основі специфікацій та сценаріїв?
29. Охарактеризуйте регресійне тестування.
30. Охарактеризуйте альфа тестування.
31. Охарактеризуйте бета тестування.
32. Документування тестових процедур для ручних та автоматизованих тестів.
33. Назвіть основне призначення систем відстеження дефектів.
34. Наведіть структуру звіту про проблему.
35. Охарактеризуйте класи еквівалентності і граничні умови.
36. У чому полягає тестування функціональності?
37. У чому полягає тестування переходів між станами?
38. У чому полягає тестування випробування навантажень?

Тема 8. Супровід програмного забезпечення

1. Розкрийте сутність та методи супроводу програмних систем.
2. Поясніть організацію процесу супроводу.
3. Назвіть методи супроводу.
4. Назвіть види діяльності, які передбачає етап експлуатації і супроводження.
5. Визначення завдань супроводу згідно зі стандартом IEEE 1219-1992.
6. Розкрийте зміст поняття "контроль функціонування".
7. Розкрийте зміст поняття "внесення необхідних змін".
8. Розкрийте зміст поняття "модернізація функцій".
9. Розкрийте зміст поняття "розширення можливостей системи".
10. Поясніть принципи управління супроводом програмних систем.
11. Поясніть сутність та призначення метрик супроводу та їх використання.
12. Назвіть завдання технології супроводу ПП.
13. Назвіть основні технології супроводу ПП.

6.2. Перелік тем презентацій

Змістовий модуль 1 Загальні питання та методи ТСПП

Тема 1. Базові поняття створення програмних продуктів

1. Використання головних понять та категорій технології розроблення програмного продукту.
2. Мета та зміст атестації програмного продукту.
3. Шляхи удосконалення процесу створення програмного продукту.
4. Специфіка та особливості створення програмного продукту.
5. Структура життєвого циклу програмного продукту.
6. Стадії життєвого циклу програмного продукту.
7. Послідовність процедури прийняття рішень.
8. Класифікація задач прийняття рішень.
9. Найпоширеніші моделі життєвого циклу програмного продукту.
10. Переваги та недоліки каскадної моделі.
11. Переваги та недоліки спіральної моделі.
12. Уніфікований процес розроблення програмного продукту.
13. Галузі знань SWEBOOK інженерії розроблення програмного продукту.
14. Основні методи програмної інженерії.
15. Циклічний характер розроблення програмного продукту.

Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення

1. Модель життєвого циклу програмного засобу.
2. Каскадна модель життєвого циклу програмного продукту.
3. Основні переваги застосування каскадної моделі життєвого циклу програмного продукту.
4. Спіральна модель життєвого циклу програмного продукту.
5. Назвіть основні недоліки застосування каскадної моделі життєвого циклу програмного продукту.
6. Основні переваги застосування спіральної моделі життєвого циклу програмного продукту.

7. Загальні риси інкрементної та еволюційної моделі.
8. Призначення процесів життєвого циклу стандарту.
9. Модель життєвого циклу на основі стандарту.
10. Класифікація процесів життєвого циклу стандарту.
11. Процеси управління проектом.
12. Процеси управління якістю.
13. Порівняльне оцінювання моделі процесів життєвого циклу стандарту 12207 і областей-процесів ядра знань SWEBOOK.
14. Робочі процеси, які включає структура життєвого циклу програмного продукту на основі методології RUP.
15. Задачі моделювання предметної області.

Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення

1. Технологія розроблення програмного забезпечення згідно з *Rational Unified Process (RUP)*.
2. Основні завдання RUP.
3. Відмінні риси RUP.
4. Проектна група (*Development Team*).
5. Основні види діяльності керівника проекту.
6. Основні види діяльності менеджера з маркетингу.
7. Основні види діяльності розробника (*Developer*).
8. Основні види діяльності аналітика предметної області.
9. Основні види діяльності програміста (*Programmer*).
10. Основні види діяльності тестера (*Tester*).
11. Основні види діяльності аналітика предметної області.
12. Основні види діяльності представника групи технічної підтримки.
13. Основні види діяльності технічного редактора (*Writer*).
14. Призначення *Rational Rose*.
15. Інструменти *Rational Rose*.
16. Визначення процесу розроблення ПЗ за допомогою UML.
17. Призначення методу MSF.
18. Основні моделі MSF.
19. Управління проектом і ризиками в системі MSF.
20. Робочі процеси технології RUP.

Тема 4. Типова схема аналізу вимог

1. Процес роботи з вимогами (*Requirements Process*).
2. Техніка витягу вимог (*Elicitation Techniques*).
3. Аналіз вимог (*Requirements Analysis*).
4. Класифікація вимог (*Requirements Classification*).
5. Архітектурне проектування та розподіл вимог (*Architectural Design and Requirements Allocation*).
6. Специфікація вимог (*Requirements Specification*).
7. Функціональні та нефункціональні вимоги (*Functional and Non-functional Requirements*).
8. Категорії класифікації вимог.
9. Цілі і складові концептуального моделювання проблеми.
10. Онтологія концептуального моделювання проблеми.
11. Елементи об'єктно-орієнтованого моделювання.
12. Принцип приховування інформації.
13. Концепція моделі сценаріїв для збору вимог.
14. Основні типи об'єкти моделі.
15. Основні етапи процесу (*process*) розробки програмних систем.

Змістовий модуль 2

Реалізація, тестування та супровід ПП

Тема 5. Якість програмного забезпечення

1. Якість програмного продукту.
2. Основні критерії якості програмного продукту.
3. Система стандартів, які регламентують питання управління якістю.
4. Стандарт ISO 9001.
5. Мінімальний набір вимог до управління якістю.
6. Вимоги до менеджменту компанії.
7. Вимоги до контролю продукції.
8. Вимоги до процесу розроблення.
9. Основні аспекти та рівні моделі якості програмного продукту.
10. Характеристики якості програмного продукту.
11. Методи визначення показників якості.
12. Метрики програмного продукту та їх складові.

13. Стандарти в області якості програмного продукту.
14. Основні цілі та завдання системи управління якістю.
15. Рівні тестування, які можуть розглядатися як частина діяльності з конструювання.
16. Рівень тестування, який зазвичай базується на нефункціональних вимогах до програмного забезпечення.

Тема 6. Моделі якості та надійності в програмній інженерії

1. Зміст та сутність процесу реалізації модулів.
2. Схема процесу реалізації модулів.
3. Типова схема процесу реалізації модулів.
4. Якість реалізації.
5. Стандартні метрики коду.
6. Індивідуальні метрики коду.
7. Інспектування коду.
8. Індивідуальна програмна документація.
9. Завдання менеджменту програмного проекту.
10. Процес планування менеджменту проекту.
11. Управління ризиком.
12. Процес управління конфігурацією програмного продукту.
13. Основні завдання управління конфігурацією.
14. Рівні тестування програмного продукту як частина діяльності з конструювання.

Тема 7. Тестування програмного забезпечення

1. Причини, за якими повне тестування не може бути виконано ніколи.
2. Вплив причини "занадто велика кількість всіх можливих комбінацій вхідних даних".
3. Вплив причини "занадто велика кількість усіх можливих послідовностей виконання коду програми".
4. Вплив причини "інтерфейс користувача, що включає всі можливі комбінації дій користувача і його переміщень по програмі, є складним для повного тестування".
5. Мета тестування програмного продукту.

6. Основні принципи тестування: динамічність, завершеність, вибір, очікувана поведінка.

7. Рівні тестування.

8. Типовий план модульного тестування.

9. Методи планування тестування програмного продукту.

10. Основні принципи тестування.

11. Рівні тестування.

12. Типовий план модульного тестування.

13. Методи планування тестування документації.

14. Шляхи подальшого розвитку методів та засобів тестування.

15. Особливості багаторівневої моделі якості програмного забезпечення?

16. Складові якості програмного забезпечення.

17. Особливості етапу підготовки процесу верифікації ПЗ відповідно до стандарту ISO 12207.

18. Особливості процесу атестації програмного забезпечення відповідно до стандарту ISO 12207.

19. Особливості процесу забезпечення якості згідно стандарту ISO 9000.

20. Процес вдосконалення програмного забезпечення в стандарті ISO 12207.

21. Мета процесу забезпечення якості програмного забезпечення відповідно до стандарту ISO 12207.

Тема 8. Супровід програмного забезпечення

1. Сутність супроводу програмних систем.

2. Організація процесу супроводу.

3. Методи супроводу.

4. Види діяльності, які передбачає етап експлуатації і супроводження.

5. Завдання супроводу згідно зі стандартом IEEE 1219-1992.

6. Контроль функціонування.

7. Внесення необхідних змін.

8. Модернізація функцій.

9. Розширення можливостей системи.

10. Принципи управління супроводом програмних систем.

11. Сутність та призначення метрик супроводу та їх використання.
12. Основні завдання технології супроводу програмного продукту.
13. Основні технології супроводу програмного продукту.
14. Основні цілі розробки варіантів використання.

7. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі: індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, що винесені на поточний контроль тощо.

Індивідуально-консультативна робота з теоретичної частини дисципліни проводиться у вигляді:

- 1) індивідуальних консультацій (запитання – відповідь стосовно проблемних питань теоретичного матеріалу дисципліни);
- 2) групових консультацій (розгляд типових прикладів, практики впровадження та використання нових методів та методик у виробничу практику).

Індивідуально-консультативна робота з практичної частини дисципліни проводиться у вигляді:

- 1) індивідуальних консультацій (розгляд практичних завдань стосовно яких виникли запитання);
- 2) групових консультацій (розгляд практичних ситуацій, рольових ігор, які потребують колективного обговорення).

Індивідуально-консультативна робота для комплексної оцінки засвоєння програмного матеріалу проводиться у вигляді:

- 1) індивідуального захисту самостійних та індивідуальних завдань;
- 2) підготовки рефератів для виступу на науковому семінарі;
- 3) підготовки рефератів для виступу на науковій конференції.

8. Методи навчання

У процесі викладання навчальної дисципліни для активізації навчально-пізнавальної діяльності студентів передбачене застосування як активних, так і інтерактивних навчальних технологій, серед яких: лекції проблемного характеру, міні-лекції, робота в малих групах, комп'ютерна симуляція, презентації (табл. 8.1 і 8.2).

**Розподіл форм та методів активізації процесу навчання
за темами навчальної дисципліни**

Тема	Практичне застосування навчальних технологій
<i>Тема 1. Базові поняття створення програмних продуктів</i>	Дискусія з питання "Галузі знань SWEBOOK інженерії розробки програмного продукту"
<i>Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення</i>	Міні-лекція "Модель життєвого циклу на основі стандарту"
<i>Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення</i>	Лекція проблемного характеру з питання "Технологія розробки програмного забезпечення згідно <i>Rational Unified Process</i> "
<i>Тема 4. Типова схема аналізу вимог</i>	Лекція проблемного характеру з питання "Архітектурне проектування та розподіл вимог (<i>Architectural Design and Requirements Allocation</i>)"
<i>Тема 5. Якість програмного забезпечення</i>	Лекція проблемного характеру з питання "Характеристики якості програмного продукту"
<i>Тема 6. Моделі якості та надійності в програмній інженерії</i>	Лекція проблемного характеру з питання "Рівні тестування програмного продукту як частина діяльності з конструювання"
<i>Тема 7. Тестування програмного забезпечення</i>	Дискусія з питання "Шляхи подальшого розвитку методів та засобів тестування"
<i>Тема 8. Супровід програмного забезпечення</i>	Дискусія з питання "Завдання супроводу згідно зі стандартом IEEE 1219-1992"

Основні відмінності активних та інтерактивних методів навчання від традиційних визначаються не тільки методикою і технікою викладання, але й високою ефективністю навчального процесу, який виявляється у: високій мотивації студентів; закріпленні теоретичних знань на практиці; підвищенні самосвідомості студентів; формуванні здатності приймати самостійні рішення; формуванні здатності до ухвалення колективних рішень; формуванні здатності до соціальної інтеграції; набуття навичок вирішення конфліктів; розвитку здатності до знаходження компромісів.

Лекції проблемного характеру – один із найважливіших елементів проблемного навчання студентів. Вони передбачають поряд із розглядом основного лекційного матеріалу встановлення та розгляд кола проблемних питань дискусійного характеру, які недостатньо розроблені в науці й мають актуальне значення для теорії та практики. Лекції проблемного характеру

відрізняються поглибленою аргументацією матеріалу, що викладається. Вони сприяють формуванню у студентів самостійного творчого мислення, прищеплюють їм пізнавальні навички. Студенти стають учасниками наукового пошуку та вирішення проблемних ситуацій.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Вони проводяться, як правило, як частина заняття-дослідження. Міні-лекції відрізняються від повноформатних лекцій значно меншою тривалістю. Зазвичай міні-лекції тривають не більше 10 – 15 хвилин і використовуються для того, щоб стисло донести нову інформацію до всіх студентів. Міні-лекції часто застосовуються як частини цілісної теми, яку бажано викладати повноформатною лекцією, щоб не втомлювати аудиторію. Тоді інформація надається по черзі кількома окремими сегментами, між якими застосовуються інші форми й методи навчання.

Семінари-дискусії передбачають обмін думками і поглядами учасників з приводу даної теми, а також розвивають мислення, допомагають формувати погляди та переконання, виробляють вміння формулювати думки й висловлювати їх.

Робота в малих групах дає змогу структурувати практично-семінарські заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань, проектних робіт. Презентації можуть бути як індивідуальними, наприклад виступ одного студента, так і колективними, тобто виступи двох та більше студентів.

Комп'ютерна симуляція – це метод навчання, що спирається на використання спеціальних комп'ютерних програм, за допомогою яких можливе віртуальне моделювання бізнес-процесу. Студенти можуть змінювати параметри й дані, приймати рішення та аналізувати наслідки таких рішень. Метою використання даного методу є розвиток системного мислення студентів, їх здібностей до планування, формування вмінь розпізнавати й аналізувати проблеми, порівнювати й оцінювати альтернативи, приймати оптимальні рішення й діяти в умовах обмеженого часу.

Використання методик активізації процесу навчання

Тема навчальної дисципліни	Практичне застосування методик	Методики активізації процесу навчання
<i>Тема 1.</i> Базові поняття створення програмних продуктів	Дискусія з питання "Основні методи програмної інженерії"	Дискусія, презентації
<i>Тема 3.</i> Класифікація та сутність процесів розроблення програмного забезпечення	Лекція проблемного характеру з питання "Порівняльна оцінка моделі процесів життєвого циклу стандарту 12207 і областей-процесів ядра знань SWEBOOK"	Робота в малих групах, презентація
<i>Тема 5.</i> Якість програмного забезпечення	Дискусія з питання "Мінімальний набір вимог до управління якістю"	Комп'ютерна симуляція, презентації
<i>Тема 7.</i> Тестування програмного забезпечення	Дискусія з питання "Методи планування тестування програмного продукту"	Комп'ютерна симуляція, презентації

9. Методи контролю

Система оцінювання сформованих компетентностей (див. табл. 2.1) у студентів враховує види занять, які, згідно з програмою навчальної дисципліни, передбачають лекційні, лабораторні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою.

Відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця, контрольні заходи включають:

поточний контроль, що здійснюється протягом семестру під час проведення лекційних, лабораторних занять і оцінюється сумою набраних балів;

підсумковий/семестровий контроль, що проводиться у формі диференційованого заліку, відповідно до графіка навчального процесу.

Поточний контроль з даної навчальної дисципліни проводиться в таких формах:

- активна робота на лекційних заняттях;
- активна участь у виконанні лабораторних завдань;
- проведення поточного тестування;
- проведення контрольної роботи.

Підсумковий/семестровий контроль проводиться у формі диференційованого заліку.

Порядок проведення поточного оцінювання знань студентів

Оцінювання знань студента під час лабораторних занять проводиться за такими критеріями:

систематичність, активність та результативність роботи студента протягом семестру над вивченням програмного матеріалу дисципліни;

відвідування занять;

розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються;

вміння поєднувати теорію з практикою під час розгляду завдань обробки інформації;

якість, самостійність та своєчасність виконання завдань лабораторних занять;

ступінь засвоєння методів та моделей прийняття рішень та вміння використовувати сучасні засоби для розв'язання складних задач прийняття рішень;

ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються.

Поточний тестовий контроль (контрольна робота) проводиться два рази за семестр. Однією з основних і безперечних його переваг є мінімум часових витрат на отримання надійних підсумків контролю. У ході тестування використовуються електронні варіанти, які дозволяють отримати результати практично відразу після завершення тесту. Тестування виконує три основні взаємопов'язані функції: діагностичну, навчальну і виховну. Діагностична функція полягає у виявленні рівня знань, умінь, навичок студента. Це основна функція тестування. Навчальна функція тестування полягає в мотивуванні студента до активізації роботи із засвоєння навчального матеріалу. Для посилення навчальної функції тестування можуть бути використані додаткові заходи стимулювання студентів, такі, як роздача викладачем

приблизного переліку питань для самостійної підготовки, наявність у самому тесті підказок, спільний розбір результатів тесту. Виховна функція проявляється в періодичності тестового контролю. Це дисциплінує, організовує і спрямовує діяльність студентів, допомагає виявити й усунути прогалини в знаннях, формує прагнення розвинути свої здібності.

Тестовий контроль, як засіб перевірки знань студентів під час проведення поточного і підсумкового контролю має такі переваги [71; 72]:

тести є значно якіснішими і об'єктивним способом оцінювання завдяки стандартизованій процедурі проведення;

на всіх етапах тестування неможливо ввести суб'єктивну складову в оцінку, вона не залежить від настрою викладача, його стосунків з конкретним студентом, враженням від відповідей на попередні запитання;

тести мають велику ємність;

показники тестів орієнтовані на вимірювання ступеня, визначення рівня засвоєння ключових понять, тем і розділів навчальної програми, умінь, навичок, а не на констатацію наявності у студентів певної сукупності засвоєних знань;

стандартизована форма оцінювання досягнень дозволяє зіставити рівень досягнень студента з дисципліни в цілому та окремих його розділів із середнім рівнем досягнень у групі й рівнями досягнень кожного з них;

виконуючи підсумкову тестову роботу, кожний студент використовує знання з усіх тем програми в той час, як на усний іспит за звичай виноситься 2 – 4 теми, на письмовий – трохи більше;

тести є м'якішим інструментом;

тестування ставить всіх студентів у рівні умови, використовуючи єдину процедуру і єдині критерії оцінювання, що приводить до зниження передекзаменаційних нервових напружень;

тести мають ширшу шкалу оцінювання, її можна розширити як угору, так і донизу;

гуманізм тестування полягає в тому, що всім надаються рівні можливості, широта тесту дозволяє студенту показати свої досягнення на широкому полі матеріалу;

студент має деяке право на помилку, яке за традиційного способу оцінювання відсутнє.

Можуть бути передбачені такі різновиди тестів:

Перший різновид тестів превентивні тести. Такий різновид тестів містить анонс матеріалів, з якими має познайомитися студент в новому розділі або дисципліні. Крім того, превентивні тести ілюструють, які саме

знання й навички отримає студент в результаті його освоєння. Тут матеріали подаються в привабливій, емоційно яскравій формі, шляхом включення мультимедійних об'єктів (графіка, відео, аудіо) у текст запитання. Ці тести доступні для незареєстрованих студентів. Широко використовують мультимедійні форми подання матеріалу при повідомленні про помилки. Такі повідомлення провокуються відсутністю правильних відповідей з метою підказати, які методики має вивчити студент, щоб відповідати правильно.

Другий різновид тестів вхідний контроль. Такий різновид тестів передуює вивченню дисципліни, крім того він заснований на загальному банку завдань і він компілюється розробником, а не створюється. У разі неправильної відповіді тест повинен містити посилання до інших дисциплін, також його можна ініціювати повторно, але обов'язковий позитивний результат.

Третій різновид тестів самоконтроль. Такий різновид тестів здійснюється, у тому числі і повторно, і перевіряється самим студентом. При цьому позитивний результат не обов'язковий. Передбачається, що такий різновид тестів передуює тестам проміжного або підсумкового контролю, сприяючи адаптації студента до програми тестування й для попередньої самоперевірки. У разі неправильної відповіді повинен містити докладні коментарі.

Четвертий різновид тестів проміжний і підсумковий контроль. Проміжний контроль завершує частину дисципліни, а підсумковий – всю дисципліну. Він вимагає авторизації, найчастіше накладає обмеження на того, хто тестується за часом і місцем проведення тесту. Ураховується не тільки точність відповіді, але і його час. Такий тест передбачає індивідуальний добір списку питань. Результати подаються студенту в узагальненому вигляді (оцінка), але бажано структурувати оцінку за тематичними блоками. У той же час розробнику дисципліни й, особливо, викладачеві важливо бачити докладну оцінку з кожного питання окремо.

П'ятий різновид тестів контроль залишкових знань. Контроль залишкових знань провадиться за рамками дисципліни. Такий різновид тестів часто використовується при компіляції вхідного контролю інших дисциплін.

Програмний продукт, який забезпечить ефективну перевірку знань студентів шляхом електронного тестування, повинен відповідати низці вимог. Фундаментом архітектури такого програмного продукту має стати принцип подвійності його функціонування, тобто він призначається як викладачеві, так і студентові. Тест повинен включати запитання одиничного і множинного вибору щодо перевірки знань основних категорій навчальної дисципліни. Максимальна кількість балів за контрольну роботу становить 10.

Розподіл балів має такий вигляд: якщо студент дав правильну відповідь на 0 – 5 % запитань, то він отримує 1 бал;

6 – 15 % – 2 бали;

16 – 25 % – 3 бали;

26 – 35 % – 4 бали;

36 – 45 % – 5 балів;

46 – 55 % – 6 балів;

56 – 65 % – 7 балів;

66 – 75 % – 8 балів;

76 – 85 % – 9 балів;

86 – 95 % – 10 балів.

Приклади запитань одиничного та множинного вибору наведені на рис. 9.1, 9.2.

Файл Тест ?

Вопрос # 3 из 14:

Програмная ошибка - это:

Выберите несколько из 6 вариантов ответа:

- 1 расхождение между программой и ее спецификацией, когда спецификация существует и она верна
- 2 несоответствие разработанного программного продукта нормам системы стандартов менеджмента качества ISO
- 3 несоответствие разработанного программного продукта мировым стандартам
- 4 расхождение разработанного программного продукта с европейским стандартом
- 5 расхождение между программой и ее спецификацией
- 6 функциональность

Далее (проверить)

Рис. 9.1. Приклад запитання одиничного вибору

С точки зрения ISO качество ПО определяется как совокупная характеристика с учетом следующих составляющих:

Выберите несколько из 6 вариантов ответа:

<input checked="" type="checkbox"/> 1	надежность
<input checked="" type="checkbox"/> 2	сопровождаемость
<input checked="" type="checkbox"/> 3	удобство использования (практичность)
<input checked="" type="checkbox"/> 4	эффективность
<input checked="" type="checkbox"/> 5	мобильность (портативность)
<input checked="" type="checkbox"/> 6	функциональность

Рис. 9.2. Пример вопроса множественного выбора

Перелік питань тесту

1. До основних завдань бізнес-моделювання відносяться:

- а) опис бізнес-процесів, бізнес-функцій;
- б) опис бізнес-процесів і дійових осіб, в них беруть участь;
- в) опис бізнес-процесів, бізнес-функцій, бізнес-правил і всіх дійових осіб;
- г) опис дійових осіб і організаційної структури підприємства, що автоматизується.

2. До основних цілей розробки варіантів використання відносяться (вибрати аргументовано):

- а) визначення меж проектованої системи, вимог, до неї пред'являються;
- б) визначення основних дійових осіб для реалізації бізнес-функцій;
- в) розробка концептуальної моделі, визначення меж проектованої системи, вимог та документації.

3. Варіант використання це:

- а) тип сервісу, що надаються системою конкретному чинному особі;
- б) безліч сервісів, що надаються системою її дійовим особам;
- в) варіант взаємодії дійових осіб з системою.

4. Структура життєвого циклу програмного забезпечення на основі методології RUP включає робочі процеси:

- а) управлінські, основні;
- б) основні, допоміжні;
- в) управлінські, допоміжні.

5. До функціональних вимоги розроблення програмного забезпечення відносяться:

- а) вимоги розробника;
- б) бізнес-вимоги, призначені для користувача вимоги;
- в) опис цілей і завдань створюваної системи.

6. Вимоги до програмного забезпечення можуть розроблятися стосовно:

- а) продукту, його властивостям;
- б) процесу, окремим його стадіям;
- в) процесу і його результату.

7. Функціональні вимоги до програмного забезпечення визначають:

- а) що необхідно зробити;
- б) як необхідно зробити;
- в) за допомогою кого необхідно зробити.

8. Нефункціональні вимоги до програмного забезпечення визначають:

- а) що необхідно зробити;
- б) як необхідно зробити;
- в) за допомогою кого необхідно зробити.

9. Функціональні вимоги до програмного забезпечення включають:

- а) бізнес-вимоги;
- б) бізнес-правила;
- в) бізнес-вимоги і користувальницькі вимоги.

10. Процес ISO 12207 (15504) "Документування" відноситься до процесу:

- а) основного;
- б) допоміжного;
- в) організаційного.

11. Процес ISO 12207 (15504) "Забезпечення якості" відноситься до процесу:

- а) основного;
- б) допоміжного;
- в) організаційного.

12. Процес ISO 12207 (15504) "Управління проектами" відноситься до процесу:

- а) основного;
- б) управління конфігурацією;
- в) організаційного.

13. Виберіть види тестування за ознакою рівня (Test Levels):

- а) висхідне, спадне, модульне, інтеграційне;
- б) модульне, системне, інтеграційне;
- в) дослідницьке, приймальне, польове;
- г) системне, інтеграційне, модульне.

14. Зв'язок "розширення" (extend) на діаграмах варіантів використання показує:

- а) збільшення кількості функцій бізнес-процесу;
- б) механізм успадкування властивостей класами-нащадками від батьківського класу;
- в) частину альтернативного потоку;
- г) визначення точки розширення для варіанту використання та його деталізацію (зміна функціональності) на основі цієї точки.

15. Основний потік відрізняється від альтернативного таким чином:

- а) наявністю різних післямов і передумов;
- б) наявністю умови його реалізації на відміну від основного потоку;
- в) наявністю виключення або включення.

16. До типів робочих продуктів (артефактів) в технології RUP відносяться:

- а) документ, модель варіанта використання;
- б) варіант використання;
- в) план, документ, модель.

17. Дії в технології RUP розбиваються на:

- а) процедури;
- б) кроки;
- в) задачі.

18. В якості шаблонів в технології RUP можуть виступати:

- а) розроблені робочі продукти;
- б) наявні прототипи;
- в) існуючі продукти і прототипи проектування.

19. Ітерації розробки програмного забезпечення згідно технології RUP вимірюються:

- а) складністю системи;
- б) тривалістю;
- в) кількістю ітерацій, тривалістю;
- г) масштабом організації, тривалістю.

20. CASE-засоби розробки програмного забезпечення – це:

- а) інструментальні програмні засоби для проектування програмного забезпечення, формування вимог, кодування, тестування, документування та інших дій по створенню програмного забезпечення;
- б) інструментальні програмні засоби для аналізу вимог і проектування, текстові редактори, системи аналізу вихідних текстів;
- в) середовища візуального програмування і генерації вихідних кодів, компілятори, інтерпретатори;
- г) інструментальні програмні засоби для проектування програмного забезпечення, формування вимог, кодування.

21. Процес (process) розроблення програмного забезпечення – це:

- а) набір машинних програм, процедур і, можливо, пов'язаних з ними документації та даних;
- б) послідовність дій, спрямована на отримання результату для клієнта;
- в) набір взаємозв'язаних робіт, які перетворюють вихідні дані (вимоги) у вихідні результати (артефакти) для певних користувачів (клієнтів);
- г) набір зазвичай послідовних фаз проекту, кількість і склад яких визначається потребами управління проектом організацією або організаціями, які беруть участь у проекті.

22. Модель бізнес-процесів використовується для:

- а) відображення дійових осіб підприємства, їх автоматизуються функцій в прив'язці до підрозділам, в яких ці функції виконуються;

б) визначення бізнес-вимог до розроблюваної системи та виявлення всіх зв'язків між підрозділами, які беруть участь у вирішенні конкретного завдання;

в) формулювання вимог до функціонального поведінки проектованої системи, описи сервісів, які система надає акторові.

23. Передумова до варіанту використання – це:

а) набір умов, що виконуються після завершення основного потоку;

б) умови, які повинні бути виконані перед тим, як почне безпосередньо виконуватися варіант використання;

в) послідовність станів системи в ході реалізації основного потоку подій;

г) частина основного або альтернативного потоку.

24. Бізнес-варіант використання реалізує:

а) бізнес-процес;

б) бізнес-функцію;

в) бізнес-мету.

25. До завдань моделювання предметної області відноситься опис:

а) бізнес-процесів підприємства; дійових осіб; бізнес сутностей; business use case; станів бізнес-сутностей; бізнес-правил;

б) бізнес-процесів підприємства; дійових осіб; станів бізнес-сутностей;

в) бізнес-функцій, бізнес-правил, цілей і завдань розроблюваної системи, що автоматизує рішення конкретної задачі.

26. До якого класу тестів відноситься перевірка правильності роботи системи, навігація, оброблення даних тощо:

а) системне тестування;

б) функціональне тестування;

в) GUI-тестування;

г) UNIT-тестування;

д) стрес-тестування?

27. Мета програмної інженерії – це:

а) розроблення методів об'єктно-орієнтованого проектування та програмування;

б) скорочення вартості програмного забезпечення та підвищення ефективності його використання;

в) оптимізація життєвого циклу програмного забезпечення.

28. Перед виконанням тестового запуску програми потрібно задати:

а) сценарій тестування, параметри тестування, початкові умови, цілі, правила зупинки тестування;

б) тестовий набір, початкові умови, цілі, критерій адекватності тесту;
в) правила зупину тестування, цілі, критерій успішності тесту, сценарій тестування, параметри тестування, початкові умови.

29. *Виберіть список напрямів діяльності, які вказуються в заяві про політику та цілі стандарту ISO 9000:*

а) технологія, вдосконалення продукції, завоювання ринків, задоволеність персоналу, економія ресурсів;

б) нова техніка, нові технології, завоювання ринків, задоволеність персоналу, зниження відходів;

в) нова техніка, вдосконалення продукції, задоволеність споживачів, соціальні питання, постачальники;

г) нова техніка, нова продукція, задоволеність споживачів, цілі, пріоритети, задоволеність персоналу, економія ресурсів.

30. *Виберіть правильний список керівництва з якості, який вказується в ISO 9000:*

а) система менеджменту якості, менеджмент ресурсів;

б) технологія, вдосконалення продукції, завоювання ринків, економія ресурсів;

в) система менеджменту якості, менеджмент ресурсів, процеси життєвого циклу продукції, вимірювання, аналіз і поліпшення.

31. *Тестування програмного забезпечення – це:*

а) процес перевірки взаємодії між програмними компонентами, модулями, використовуваний для традиційних продуктів, ієрархічно структурованих систем;

б) демонстрація того, що програмне забезпечення задовольняє пропонувані вимогами і, відповідно, не знайдено дефектів;

в) спостереження за виконанням програми, запущеної з метою тестування із заданими параметрами, за заданим сценарієм або з іншими заданими початковими умовами або цілями тестування.

32. *Виберіть існуючі рівні тестування з наведених:*

а) модульне (*unit testing*), інтеграційне (*integration testing*), системне тестування (*system testing*);

б) функціональне (*functional testing*), нефункціональне (*non-functional testing*), змішане (*mixed testing*);

в) модульне (*unit testing*), системне тестування (*system testing*);

г) тестування рівня (*level testing*), відносне (*relational testing*), управлінське (*administrative testing*).

33. *Виберіть рівень тестування, який зазвичай базується на нефункціональних вимогах до ПЗ:*

- а) змішане тестування (*mixed testing*);
- б) системне тестування (*system testing*);
- в) модульне тестування (*unit testing*).

34. *До етапу підготовки процесу верифікації ПЗ згідно стандарту ISO 12207 відноситься:*

- а) визначення необхідності верифікації, визначення робіт і програмних продуктів, які потребують верифікації, розробка плану верифікації, виконання плану верифікації, усунення виявлених проблем;
- б) визначення необхідності верифікації, усунення виявлених проблем;
- в) визначення необхідності верифікації, визначення робіт і програмних продуктів, які потребують верифікації, усунення виявлених проблем.

35. *До етапу верифікації процесу верифікації ПЗ відповідно до стандарту ISO 12207 відноситься (вибрати найбільш повну відповідь):*

- а) верифікація процесу, вимог, програми;
- б) верифікація договору, збірки, документації;
- в) верифікація вимог.

36. *Процес атестації ПО згідно зі стандартом ISO 12207 – це:*

- а) визначення якості встановлених вимог, у відповідності до створеного програмного продукту його функціональному призначенню;
- б) визначення точності встановлених вимог, у відповідності до створеного програмного продукту його функціональному призначенню;
- в) визначення повноти встановлених вимог, у відповідності до створеного програмного продукту його функціональному призначенню.

37. *Зрілість технології в моделі CMM визначається:*

- а) продуктивністю і діапазоном результатів, які можна очікувати від організації;
- б) продуктивністю процесу і фактичними результатами, досягнутими організацією;
- в) ступенем визначеності, керованості, спостереження, контрольованості та ефективності процесу.

38. *Для кожного ключового процесу в моделі CMM визначаються (вибрати найбільш повну відповідь):*

- а) цілі, ключові практики, критерії ефективності процесу;

б) цілі, ключові практики, обсяг і кордон процесу, критерії ефективності процесів;

в) ключові практики, обсяг і кордон процесу.

39. Процес забезпечення якості в ISO 9000 складається з таких робіт:

а) забезпечення продукту, забезпечення процесу, забезпечення систем якості, атестація, аудит і рішення проблем;

б) атестація, спільні аналізи, аудит і рішення проблем;

в) верифікація, атестація, спільні аналізи, аудит і рішення проблем;

г) підготовка процесу, забезпечення продукту, забезпечення процесу, забезпечення систем якості.

40. Процес верифікації ПЗ відповідно до стандарту ISO 12207 – це:

а) визначення того, що ПЗ функціонує у повній відповідності з детальними вимогами;

б) визначення того, що ПЗ функціонує у повній відповідності з усіма вимогами;

в) визначення того, що ПЗ функціонує в повній відповідності з вимогами замовника і їм задовольняє.

41. Визначення завдань, плану, виконання плану атестації ПО відносяться до такого етапу проведення атестації:

а) підготовка процесу;

б) атестація.

42. Підготовка вимог до тестування, контрольних прикладів і технічних умов випробувань, забезпечення відповідності вимог, контрольних прикладів і технічних умов випробувань конкретним вимогам, проведення випробувань відносяться до такого етапу проведення атестації:

а) підготовка процесу;

б) атестація.

43. Процес удосконалення в стандарті ISO 12207 – це:

а) процеси встановлення, контролю і поліпшення будь-якого процесу життєвого циклу програмного забезпечення;

б) процеси встановлення, оцінки, вимірювання, контролю та поліпшення будь-якого процесу життєвого циклу програмного забезпечення;

в) процеси оцінки, вимірювання, контролю процесу життєвого циклу програмного забезпечення.

44. Процес вдосконалення в стандарті ISO 12207 включає в себе роботи:

- а) створення процесу, його оцінювання;
- б) створення та удосконалення процесу;
- в) створення процесу, його оцінювання, удосконалення.

45. Здатність до тестування (*Testability*) програмного забезпечення – це:

- а) можливість (ймовірність), статистичного вимірювання того, що при тестуванні проявиться збій програмної системи;
- б) ступінь легкості покриття тестами заданої програмної системи;
- в) ступінь легкості покриття тестами заданої програмної системи і можливість (ймовірність) статистичного вимірювання того, що при тестуванні проявиться збій програмної системи;
- г) здатність програми запускатися і бути спостережуваною для фіксації її поведінки.

46. Перед виконанням тестового запуску програми потрібно задати:

- а) сценарій тестування, параметри тестування, початкові умови, цілі, правила зупинки тестування;
- б) тестовий набір, початкові умови, цілі, критерій адекватності тесту;
- в) правила зупину тестування, цілі, критерій успішності тесту, сценарій тестування, параметри тестування, початкові умови.

47. Мета процесу забезпечення якості відповідно до стандарту ISO 12207 – це:

- а) адаптація процесу забезпечення якості до умов конкретного проекту;
- б) забезпечення продуктів і процесів вимогам та планам;
- в) верифікація, атестація, спільні аналізи, аудит і рішення проблем;
- г) координація процесу з процесами верифікації, атестації, спільного аналізу та аудиту.

48. Які рівні тестування з наведених можуть розглядатися як частина діяльності з конструювання:

- а) модульне (*unit testing*), інтеграційне (*integration testing*);
- б) інтеграційне (*integration testing*), системне тестування (*system testing*);
- в) модульне (*unit testing*), системне тестування (*system testing*);
- г) модульне (*unit testing*), інтеграційне (*integration testing*), системне тестування (*system testing*).

49. Виберіть до складу будь фази СММ входить управління процесами і виробництвом:

- а) фаза відбраковування;
- б) фаза управління якістю;
- в) фаза планування якості.

50. Для перевірки працездатності окремих елементів системи використовується:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- д) стрес-тестування;
- е) навантажувальне тестування.

51. Яке тестування необхідно виконати для перевірки працездатності програми в екстремальних умовах:

- а) системне тестування
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- д) стрес-тестування;
- е) навантажувальне тестування?

52. Тестування характеристик програмного забезпечення, пов'язаних зі швидкістю обробки даних, – це:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- д) стрес-тестування;
- е) навантажувальне тестування.

53. Які документи використовуються для розробки плану тестування (*test plan*):

- а) технічне завдання;
- б) тестові специфікації (*test case specification*);
- в) тестові процедури (*test procedure specification*);
- г) підсумковий звіт тестування (*test summary report*)?

54. Які з критеріїв тестування використовують інформацію про структуру програми:

- а) структурні критерії;
- б) функціональні критерії;
- в) критерії стохастичного тестування;
- г) мутаційні критерії?

55. На якому етапі тестування застосовуються структурні критерії:

- а) регресійне тестування;
- б) модульне тестування;
- в) інтеграційне тестування;
- г) системне тестування.

56. Які з критеріїв тестування забезпечують контроль ступеня виконання вимог замовника в програмному продукті:

- а) структурні критерії;
- б) функціональні критерії;
- в) критерії стохастичного тестування;
- г) мутаційні критерії?

57. Яка модель тестування використовується під час функціонального тестування:

- а) чорний ящик;
- б) білий ящик?

58. Яка модель тестування використовується під час модульного тестування:

- а) чорний ящик;
- б) білий ящик?

59. Яка модель тестування використовується під час інтеграційного тестування:

- а) чорний ящик;
- б) білий ящик?

60. Виявлення помилок, пов'язаних з роботою системи в цілому, виконується:

- а) системним тестуванням;
- б) модульним тестуванням;
- в) інтеграційним тестуванням;
- г) тестуванням інтерфейсів;

- д) стрес-тестуванням;
- е) навантажувальним тестуванням.

61. До якого класу тестування можна віднести тестування інтерфейсів (екранів, кнопок тощо):

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- д) стрес-тестування;
- е) навантажувальне тестування?

62. До якого класу тестів відноситься перевірка правильності роботи системи, навігація, обробка даних тощо:

- а) системне тестування;
- б) функціональне тестування;
- в) GUI-тестування;
- г) UNIT-тестування;
- д) стрес-тестування;
- е) навантажувальне тестування?

63. Тестується реєстраційна форма. Після введення всіх даних ви натискаєте кнопку підтвердження введення і бачите повідомлення про невірний введенні даних в полі Дата народження:

- а) це є помилкою роботи додатка;
- б) додаток працює правильно.

64. Тестується реєстраційна форма. Після введення чисел у полі Прізвище відображається модальне вікно з повідомленням про помилку:

- а) це є помилкою роботи додатка;
- б) додаток працює правильно.

65. Тестується головне меню додатка. Під час натискання пункту меню Довідники відкривається форма справ очників без заголовка:

- а) це є помилкою роботи додатка;
- б) додаток працює правильно.

Критерії оцінювання позааудиторної самостійної роботи студентів. Загальними критеріями, за якими здійснюється оцінювання позааудиторної самостійної роботи студентів, є: глибина і міцність знань, рівень мислення, вміння систематизувати знання за окремими темами, вміння

робити обґрунтовані висновки, володіння категорійним апаратом, навички і прийоми виконання практичних завдань, вміння знаходити необхідну інформацію, здійснювати її систематизацію та обробку, самореалізація на практичних та семінарських заняттях.

Критеріями оцінювання презентацій є:

здатність проводити критичне та незалежне оцінювання певних проблемних питань;

вміння пояснювати альтернативні погляди та наявність власної точки зору, позиції на певне проблемне питання;

застосування аналітичних підходів;

якість і чіткість викладення міркувань;

логіка, структуризація та обґрунтованість висновків щодо конкретної проблеми;

самостійність виконання роботи;

грамотність подачі матеріалу;

використання методів порівняння, узагальнення понять та явищ;

оформлення роботи.

Порядок підсумкового контролю з навчальної дисципліни. Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі проведення диференційованого заліку.

Студента слід **вважати атестованим**, якщо сума балів, одержаних за результатами підсумкової/семестрової перевірки успішності, дорівнює або перевищує 60.

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час поточного контролю за накопичувальною системою.

Якщо, сумарний результат у балах за семестр складає: *"60 і більше балів – зараховано"*, *"59 і менше балів – не зараховано"* та заноситься у залікову *"Відомість обліку успішності"* навчальної дисципліни.

10. Розподіл балів, які отримують студенти

Система оцінювання рівня сформованості професійних компетентностей студентів денної форми навчання наведена в табл. 10.1.

Система оцінювання рівня сформованості професійних компетентностей

Професійні компетентності	Навчальний тиждень	Год	Методи та форми навчання		Оцінка рівня сформованості компетентностей			
					Форми контролю	Максимальний бал		
1	2	3	4		5	6		
Змістовий модуль 1								
Загальні питання та методи ТСПП							50	
ПРЗГІС 1	Знати етапи життєвого циклу ПП	1	Ауд.	2	Лекція	Тема 1. Базові поняття створення програмних продуктів	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 1. Дослідження методики розробки специфікації вимог до програмного продукту	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 1	Знати етапи життєвого циклу ПП	2	Ауд.	4	Лабораторне заняття	Лабораторне заняття 1. Дослідження методики розробки специфікації вимог до програмного продукту	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 1	Аналізувати, створювати та обирати склад та типи вимог до ПП	3	Ауд.	2	Лекція	Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 1. Дослідження методики розробки специфікації вимог до програмного продукту	Активна участь у виконанні досліджень	2
							Захист лабораторної роботи №1	4
ПРЗГІС 1	Знати етапи життєвого циклу програмного продукту	4	Ауд.	4	Лабораторне заняття	Лабораторне заняття 2. Дослідження основних елементів мови програмування <i>Ruby</i>	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		

Продовження табл. 10.1

1		2	3		4		5	6
ПРЗГІС 1	Знати сучасні процеси проектування та розроблення програмних продуктів	5	Ауд.	2	Лекція	Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 2. Дослідження основних елементів мови програмування <i>Ruby</i>	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 1	Знати сучасні процеси проектування та розроблення програмних продуктів	6	Ауд.	4	Лабораторне заняття	Лабораторне заняття 2. Дослідження основних елементів мови програмування <i>Ruby</i>	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	Захист лабораторної роботи №2	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 2	Знати теоретичні основи управління якістю програмних продуктів	7	Ауд.	2	Лекція	Тема 4. Типова схема аналізу вимог	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 3. Знайомство з фреймворком <i>Rails</i> . Аналіз систем контролю версій, основи роботи з GIT	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 2	Знати теоретичні основи управління якістю програмних продуктів	8	Ауд.	4	Лабораторне заняття	Лабораторне заняття 3. Знайомство з фреймворком <i>Rails</i> . Аналіз систем контролю версій, основи роботи з GIT	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	Контрольна робота №1	10
			Змістовий модуль 2					
Реалізація, тестування та супровід ПЗ								50
ПРЗГІС 2	Знати теоретичні основи управління якістю програмних продуктів	9	Ауд.	2	Лекція	Тема 5. Якість програмного забезпечення	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		

Продовження табл. 10.1

1		2	3		4		5	6
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 3. Знайомство з фреймворком <i>Rails</i> . Аналіз систем контролю версій, основи роботи з GIT	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	Захист лабораторної роботи №3	4
ПРЗГІС 2	Знати принципи управління якістю програмних продуктів	10	Ауд.	4	Лабораторне заняття	Лабораторне заняття 4. Розробка додатку відповідно до індивідуального завдання, аналіз підходів до реалізації аутентифікації та авторизації	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 3	Знати та застосовувати методології та засоби програмної інженерії	11	Ауд.	2	Лекція	Тема 6. Моделі якості та надійності в програмній інженерії	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 4. Розробка додатку відповідно до індивідуального завдання, аналіз підходів до реалізації аутентифікації та авторизації	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
ПРЗГІС 3	Знати та застосовувати методології та засоби програмної інженерії	12	Ауд.	4	Лабораторне заняття	Лабораторне заняття 4. Розробка додатку відповідно до індивідуального завдання, аналіз підходів до реалізації аутентифікації та авторизації	Активна участь у виконанні досліджень	4
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	Захист лабораторної роботи №4	4
ПРЗГІС 3	Знати та застосовувати методи моделювання для опису об'єктів інформатизації	13	Ауд.	2	Лекція	Тема 7. Тестування програмного забезпечення	Робота на лекції	2
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 5. Дослідження основних можливостей фреймворку <i>Twitter Bootstrap</i> , основи роботи з мовою програмування <i>JavaScript</i> та фреймворком <i>JQuery</i>	Активна участь у виконанні досліджень	2
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою		

Закінчення табл. 10.1

1		2	3		4			5	6
ПРЗГІС 3	Знати та застосовувати методи та засоби програмної інженерії	14	Ауд.	4	Лабораторне заняття	Лабораторне заняття 5. Дослідження основних можливостей фреймворку Twitter Bootstrap, основи роботи з мовою програмування JavaScript та фреймворком JQuery	Активна участь у виконанні досліджень	4	
			СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою			
ПРЗГІС 3	Знати та застосовувати основні підходи до об'єктно-орієнтованого проектування інформаційних систем	15	Ауд.	2	Лекція	Тема 8. Супровід програмного забезпечення	Робота на лекції	2	
			СРС	5	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою			
			Ауд.	2	Лабораторне заняття	Лабораторне заняття 5. Дослідження основних можливостей фреймворку Twitter Bootstrap, основи роботи з мовою програмування JavaScript та фреймворком JQuery	Активна участь у виконанні досліджень	2	
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	Захист лабораторної роботи №5 Контрольна робота №2	4 10	
Усього годин			144		Загальна максимальна кількість балів із дисципліни			100	
з них									
аудиторні			60	42 %	поточний контроль			100	
самостійна робота			84	58 %	підсумковий контроль			100	

Розподіл балів у межах тем змістових модулів наведено в табл. 10.2.

Таблиця 10.2

Розподіл балів за темами

Поточне тестування та самостійна робота								Сума
Змістовий модуль 1				Змістовий модуль 2				
T1	T2	T3	T4	T5	T6	T7	T8	100
8	12	12	18	12	12	8	18	

Максимальну кількість балів, яку може накопичити студент протягом тижня за формами та методами навчання, наведено в табл. 10.3.

Розподіл балів за тижнями

Теми			Лекційні заняття	Лабораторні заняття	Захист лабораторної роботи	Контрольна робота	Усього
Змістовий модуль 1. Загальні питання та методи ТСПП	Тема 1	2 тижня	2	6	–	–	8
	Тема 2	2 тижня	2	6	4	–	12
	Тема 3	2 тижня	2	6	4	–	12
	Тема 4	2 тижня	2	6	–	10	18
Змістовий модуль 2. Реалізація, тестування та супровід ПЗ	Тема 5	2 тижня	2	6	4	–	12
	Тема 6	2 тижня	2	6	4	–	12
	Тема 7	2 тижня	2	6	–	–	8
	Тема 8	1 тиждень	2	2	4	10	18
Усього			16	44	20	20	100

Підсумкова оцінка з навчальної дисципліни визначається відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця (табл. 10.4).

Оцінки за цією шкалою заносяться до відомостей обліку успішності, індивідуального навчального плану студента та іншої академічної документації.

Таблиця 10.4

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D	задовільно	
60 – 63	E		
35 – 59	FX	незадовільно	не зараховано
1 – 34	F		

11. Рекомендована література

11.1. Основна

1. Бабенко Л. П. Основи програмної інженерії : навч. посіб. / Л. П. Бабенко, К. М. Лаврищева. – К. : Знання, 2001. – 270 с.
2. Буч Г. UML. Классика CS. 2-е изд. / Г. Буч, А. Якобсон, Дж. Рамбо ; под общ. ред. проф. С. Орлова. – СПб. : Питер, 2006. – 736 с.
3. Буч Г. Язык UML. Руководство пользователя. Второе издание / Г. Буч, А. Якобсон, Дж. Рамбо. – М. : ДМК, 2006. – 496 с.
4. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++, 2-е изд. / Г. Буч. – СПб. : Изд. Бином, 1999. – 560 с.
5. Кознов Д. В. Языки визуального моделирования: проектирование и визуализация программного обеспечения : учеб. пособ. / Д. В. Кознов. – СПб. : Изд. СПбГУ, 2004. – 143 с.
6. Липаев В. В. Программная инженерия. Основы / В. В. Липаев. – М. : Радио и связь, 2006. – 488 с.
7. Ларман Крэг. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development / Крэг Ларман. – М. : Вильямс, 2006. – 736 с.
8. Ларман К. Применение UML и шаблонов проектирования : учеб. пособ. / К. Ларман. – М. : Вильямс, 2001. – 145 с.
9. Орлик С. Введение в программную инженерию и управление жизненным циклом ПО / С. Орлик, Ю. Балуй. – М. : Синтег, 2005. – 654 с.
10. Шафер Д. Управление программными проектами: достижение оптимального качества при минимуме затрат / Д. Шафер, Р. Фатрел, Л. Шафер. – М. : Вильямс, 2003. – 136 с.
11. Шмуллер Дж. Освой самостоятельно UML 2 за 24 часа. Практическое руководство = Sams Teach Yourself UML in 24 Hours, Complete Starter Kit / Дж. Шмуллер. – М. : Вильямс, 2005. – 416 с.
12. Фаулер М. UML. Основы / М. Фаулер. – М. : Символ-Плюс, 2005. – 192 с.

11.2. Додаткова

13. Андон Ф. И. Методы инженерии распределенных компьютерных систем / Ф. И. Андон, Е. М. Лаврищева. – К. : Изд. "Наукова думка", 1997. – 228 с.

14. Андон Ф. И. Основы инженерии качества программных систем / Ф. И. Андон, Г. И. Коваль, Т. М. Коротун и др. – К. : Академперіодика, 2002. – 502 с.
15. Барлет Н. Программирование на JAVA / Н. Барлет, А. Лесли, С. Симкин. Путеводитель. – К., 1996. – 736 с.
16. Богсс У. UML и Rational Rose / У. Богсс, М. Богсс. – М. : Изд. "Лори", 2000. – 580 с.
17. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч. – М. : Вильямс ; СПб., 2008. – 546 с.
18. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем / А. М. Вендров. – М. : Финансы и статистика, 1998. – 176 с.
19. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма. – СПб. : Питер, 2010. – 608 с.
20. Гласс Г. Сопровождение программного обеспечения / Г. Гласс, Р. Нуазо. – М. : Мир, 1983. – 256 с.
21. Гриняк В. М. Использование методов программной инженерии в процессе обучения дисциплинам типа "Программирование" / В. М. Гриняк, Н. Л. Слугина // Современные проблемы науки и образования, 2013. – № 2.
22. Гультяев А. К. MS PROJECT 2002. Управление проектами. Русская версия. Практическое пособие / А. К. Гультяев. – СПб. : Корона, 2003. – 592 с.
23. Джалота П. Управление программными проектами на практике / П. Джалота. – М. : Лори, 2005. – 367 с.
24. Джордан Д. Обработка объектных баз данных в C++. Программирование по стандарту ODMG / Д. Джордан. – М. : Издательский дом "Вильямс", 2001. – 384 с.
25. Дунаев С. Б. Доступ к базам данных и техника работы в сети / С. Б. Дунаев. – М. : Диалог-Мифи, 1999. – 416 с.
26. Иан Соммервил. Инженерия программного обеспечения / Соммервил Иан. – М. ; СПб. ; К. , 2002.– 623 с.
27. Кватрани Т. Визуальное моделирование с помощью IBM Rational Software Architect и UML / Т. Кватрани. – М. : Изд. КУДИЦ-ПРЕСС, 2007. – 354 с.
28. Кендалл Скотт. Унифицированный процесс. Основные концепции / Скотт Кендалл. – М. ; СПб. ; К. , 2002. – 157 с.
29. Кратчен Ф. Введение в Rational Unified Process / Ф. Кратчен. – М. : ИД "Вильямс", 2002. – 240 с.

30. Лаврищева Е. М. Проблематика программной инженерии / Е. М. Лаврищева. – К. : Знання. – 1991. – 19 с.
31. Ларман К. Применение UML и шаблонов проектирования / К. Ларман. – М. : ИД "Вильямс", 2001. – 496 с.
32. Липаев В. В. Системное проектирование сложных программных средств для информационных систем / В. В. Липаев. – М. : СИНТЕГ, 1999. – 224 с.
33. Мацяшек Лешек А. Анализ и проектирование информационных систем с помощью UML 2.0 / Лешек А. Мацяшек. – М. : ООО "Вильямс", 2008. – 354 с.
34. Мейер Б. Объектно-ориентированное конструирование программных систем / Б. Мейер. – М. : Русская Редакция, 2005. – 612 с.
35. Монсон-Хейфел Р. Enterprise JavaBeans / Р. Монсон-Хейфел. – СПб. : Символ-Плюс, 2002. – 672 с.
36. Мороз Г. Б. Модели роста надежности программного обеспечения / Г. Б. Мороз, Е. М. Лаврищева. – К. : Препринт, 1992. – 23 с.
37. Роджерсон Д. Основы COM / Д. Роджерсон. – Microsoft Press, 2009. – 361 с.
38. Синтес А. Освой самостоятельно объектно-ориентированное программирование за 21 день / А. Синтес. – М. : Вильямс, 2002. – 352 с.
39. Скорін Ю. І. Робоча програма навчальної дисципліни "Методи та засоби комп'ютерних інформаційних технологій" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / Ю. І. Скорін, В. В. Федько, О. В. Щербаков. Навчальне видання. – Х. : Вид. ХНЕУ, 2011. – 48 с.
40. Скорін Ю. І. Інформаційне супроводження навчального процесу / Ю. І. Скорін, В. В. Стаднік, О. В. Щербаков // Збірник наукових статей "Управління розвитком". – 2010. – Вип. 7(88). – С. 273–274.
41. Слугина Н. Л. Повышение уровня подготовки кадров в области современных информационных технологий на базе центра компетенций / Н. Л. Слугина, В. М. Гриняк // Современные проблемы науки и образования. – 2012. – № 5. – С. 205.
42. Страуструп Б. Язык программирования С++ / Б. Страуструп. – М. : Радио и связь, 1991. – 352 с.
43. Тейер Т. Надежность программного обеспечения / Т. Тейер, Р. Липов, Е. Нельсон. – М. : Мир, 1981. – 325 с.
44. Шилдт Г. МFC: основы программирования / Г. Шилдт. – К. : Издательская группа BHV, 1997. – 560 с.

45. Шилдт Г. Теория и практика C++ / Г. Шилдт. – СПб. : BHV, 1999. – 416 с.
46. Шилдт Г. Программирование на C и C++ для Windows 95 / Г. Шилдт. – К. : Торгово-издательское бюро BHV, 1996. – 400 с.
47. Шлеер С. Объектно-ориентированный анализ: моделирование мира в состояниях / С. Шлеер, С. Меллор. – К. : Диалектика, 1993. – 240 с.
48. Шмуллер Дж. Освой самостоятельно UML за 24 часа / Дж. Шмуллер. – М. : Вильямс, 2005. – 135 с.
49. Чернецки К. Порождающее программирование. Методы, инструменты, применение / К. Чернецки, У. Айзенекер. – М. ; СПб. ; Х. ; Мн. : ИД "Питер", 2005. – 730 с.
50. Трофимов С. А. CASE-технологии: практическая работа в Rational Rose / С. А. Трофимов. – М. : Бином-Пресс, 2002. – 288 с.
51. Уилсон С. Принципы проектирования и разработки программного обеспечения / С. Уилсон, Б. Мейлс, Т. Ленгрейв. Учебный курс MCSD. – М. : Изд. торговый дом "Русская редакция", 2000. – 608 с.
52. Фраулер М. UML в кратком изложении. Применение стандартного языка объектного моделирования / М. Фраулер, К. Скотт. – М. : Мир, 1999. – 191 с.
53. Фаулер М. Рефакторинг: улучшение соответствующего кода / М. Фраулер. – СПб. : Символ-Плюс, 2003. – 432 с.
54. Холл М. Сервлеты и JavaServer Pages. Библиотека программиста / М. Холл. – СПб. : Питер, 2001. – 496 с.
55. Эммерих В. Конструирование распределенных объектов. Методы и средства программирования интероперабельных объектов в архитектурах OMG/CORBA, Microsoft COM и Java RMI / В. Эммерих. – М. : Мир, 2002. – 510 с.

11.3. Інформаційні ресурси

56. Вікіпедія. ISO 9000. Системи керування якістю – Основні положення і словник [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
57. Вікіпедія. ISO 9000-1. Стандарти з керування якістю та забезпечення якості. Ч. 1. Настанови щодо вибору та застосування [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

58. Вікіпедія. ISO 9000-2. Стандарти з керування якістю та забезпечення якості. Ч. 2. Настанови щодо застосування ISO 9001, ISO 9002, ISO 9003 [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

59. Вікіпедія. ISO 9000-3. Стандарти з керування якістю та забезпечення якості – Ч. 3. Настанови щодо застосування ISO 9001 під час розроблення, постачання та супроводження програмного забезпечення [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

60. Вікіпедія. ISO 9000-4. Стандарти з керування якістю та забезпечення якості – Ч. 4. Настанови щодо керування програмою надійності [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

61. Вікіпедія. ISO 9001-2001. Системи керування якістю. Вимоги [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

62. Вікіпедія. ISO 9003. Система якості. Модель забезпечення якості в процесі контролю готової продукції та її випробуваннях [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

63. Вікіпедія. ISO/IEC 9126-1:2001. Програмна інженерія. Якість продукту. Ч. 1. Модель якості [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

64. Вікіпедія. ISO/IEC TR 9126-2:2003. Програмна інженерія. Якість продукту. Ч. 2. Зовнішні метрики [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

65. Вікіпедія. ISO/IEC TR 9126-3:2003. Програмна інженерія. Якість продукту. Ч. 3. Внутрішні метрики [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

66. Вікіпедія. ISO/IEC TR 9126-4:2004. Програмна інженерія. Якість продукту. Ч. 4. Метрики якості при використанні [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

67. Вікіпедія. ISO/IEC 14598-2:2000. Програмна інженерія. Оцінювання програмного продукту. Ч. 2. Планування та керування [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

68. Вікіпедія. ISO/IEC 14598-3:2000. Програмна інженерія. Оцінювання програмного продукту. Ч. 3. Процес для розробників [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

69. Вікіпедія. ISO/IEC 14598-4:1999. Програмна інженерія. Оцінювання продукту. Ч. 4. Процес для замовників [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

70. Вікіпедія. ISO/IEC 14598-6:2001. Програмна інженерія. – Оцінка продукту – Ч. 6. Документація модулів оцінювання [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

71. Вікіпедія. ISO/IEC 19761:2003. Програмна інженерія. COSMIC-FFP. Метод вимірювання об'єму функціональних можливостей [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

72. Вікіпедія. ISO/IEC 25000:2005. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Настанова щодо оцінювання і вимоги до якості програмного продукту [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

73. Вікіпедія. ISO/IEC 25001:2007. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Планування та керування [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

74. Вікіпедія. ISO/IEC 25020:2007. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Настанова та еталонна модель вимірювання [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

75. Вікіпедія. ISO/IEC TR 25021:2007. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (ОВЯПП - SQuaRE). Елементи вимірювання якості [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

76. Вікіпедія. ISO/IEC 25030:2007. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Вимоги якості [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

77. Вікіпедія. ISO/IEC 25051:2006. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Вимоги до якості готових комерційних продуктів програмних засобів та інструкції по тестуванню [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

78. Вікіпедія. ISO/IEC 25062:2006. Програмна інженерія. Оцінювання і вимоги до якості програмного продукту (SQuaRE). Загальний промисловий формат, який використовується при тестуванні звітів [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

79. Вікіпедія. ДСТУ ISO/IEC TR 9126-2. Програмна інженерія. Якість продукту. Ч. 2 Зовнішні метрики [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

80. Вікіпедія. ДСТУ ISO/IEC TR 9126-3. Програмна інженерія. Якість продукту. Ч. 3 Внутрішні метрики [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.

81. Вікіпедія. ДСТУ ISO/IEC TR 9126-4. Програмна інженерія. Якість продукту. Ч. 4 Метрики якості при використанні [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
82. Вікіпедія. ДСТУ ISO/IEC 14598-1. Інформаційні технології. Оцінювання програмного продукту. Ч. 1 Загальний огляд [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
83. Вікіпедія. ДСТУ ISO/IEC 14598-2. Програмна інженерія. Оцінювання програмного продукту. Ч. 2. Планування та керування [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
84. Вікіпедія. ДСТУ ISO/IEC 14598-3. Програмна інженерія. Оцінювання програмного продукту. Ч. 3. Процес для розробників [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
85. Вікіпедія. ДСТУ ISO/IEC 14598-4. Програмна інженерія. Оцінювання продукту. Ч. 4. Процес для замовників [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
86. Вікіпедія. ДСТУ ISO/IEC 14598-6. Програмна інженерія. – Оцінка продукту – Ч. 6. Документація модулів оцінювання [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
87. Вікіпедія. ДСТУ ISO/IEC 15939. Інженерія систем і програмних засобів. Процес вимірювання [Електронний ресурс]. – Режим доступу : <https://ru.wikipedia.org/wiki/ISO>.
88. Забудский Е. И. Учебно-методический комплекс дисциплины "Object-Oriented Programming, C#" / Е. И. Забудский. – М. : Кафедра Архитектуры программных систем ГУ-ВШЭ [Электронный ресурс]. – Режим доступа : <http://zei.narod.ru>.
89. Кулямин В. В. Технологии программирования. Компонентный подход. Лекция 2. Жизненный цикл и процессы разработки ПО. МГУ. ВМК. Каф. Системного программирования [Электронный ресурс] / В. В. Кулямин. – Режим доступа : <http://www.ispras.ru/RedVerst/RedVerst/Lectures and training courses /Software Development Technologies/Lecture02.doc>.
90. Липаев В. В. Стандарты, регламентирующие жизненный цикл сложных программных комплексов [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.pcweek.ru/year1998/N24/CP1251/Reviews/chapt1.htm>.
91. Липаев В. В. Программная инженерия. Методологические основы. Предисловие к курсу [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://citforum.ru/SE/lipaev/preface.shtml>.

92. Липаев В. В. Программная инженерия в жизненном цикле программных средств [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://citforum.ru/SE/lipaev/>.

93. Липаев В. В. Экономика программной инженерии заказных программных продуктов. Учебное пособие [Электронный ресурс] / В. В. Липаев. – Режим доступа : <file:///C:/Users/Мой%-20компьютер/AppData/Local/Microsoft/Windows/INetCache/IE/O7QJSFEK/Economy%20of%20Software%20Engineering%20of%20Custom%20Software.pdf>.

94. Липаев В. В. Надежность и функциональная безопасность комплексов программ реального времени [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

95. Липаев В. В. Проектирование и производство сложных заказных программных продуктов [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

96. Липаев В. В. Экономика производства программных продуктов [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

97. Липаев В. В. Человеческие факторы в программной инженерии: рекомендации и требования к профессиональной квалификации специалистов [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

98. Липаев В. В. Тестирование компонентов и комплексов программ [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

99. Липаев В. В. Сопровождение и управление конфигурацией сложных программных средств [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

100. Липаев В. В. Документирование сложных программных средств [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

101. Липаев В. В. Сертификация программных средств [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

102. Липаев В. В. Функциональная безопасность программных средств [Электронный ресурс] / В. В. Липаев. – Режим доступа : <http://www.computer-museum.ru/books/lipaev/>.

11.4. Методичне забезпечення

103. Скорін Ю. І. Робоча програма навчальної дисципліни "Технології тестування програмних продуктів" для студентів за напрямом підготовки 6.050101 "Комп'ютерні науки" усіх форм навчання / Ю. І. Скорін. – Х. : Вид. ХНЕУ, 2012. – 48 с.

104. Золотарьова І. О. Робоча програма навчальної дисципліни "Технології створення програмних продуктів" з підготовки бакалаврів напряму 6.050101 "Комп'ютерні науки" (шифр за ОПП 3.1.08) / І. О. Золотарьова, С. В. Мінухін. – Х. : Вид. ХНЕУ, 2014. – 15 с.

105. Лавріщева К. М. Робоча програма навчальної дисципліни "Технологія програмування та створення програмних продуктів, методи та засоби комп'ютерні інформаційні технології" для студентів спеціальності 6.050103 "Програмна інженерія" / К. М. Лавріщева. – К. : Київський національний університет імені Тараса Шевченка, 2012. – 10 с.

106. Мохов В. А. Рабочая программа по дисциплине "Программная инженерия" для направления 230100 "Информатика и вычислительная техника" / В. А. Мохов. – Новочеркасск : ЮРГТУ(НПИ), 2011. – 28 с.

107. Балалаев М. А. Рабочая программа по дисциплине "Технология разработки программного обеспечения" для направления 230100.68 "Информатика и вычислительная техника" / М. А. Балалаев. – Дальневосточный государственный университет путей сообщения : Естественно-научный институт, 2011. – 17 с.

108. Самойленко В. П. Рабочая программа по дисциплине "Технология разработки программного обеспечения" для направления 654600 "Информатика и вычислительная техника" / В. П. Самойленко. – СПб. : Санкт-Петербургский государственный электротехнический университет "ЛЭТИ", 2001. – 19 с.

109. Воробьева М. С. Рабочая программа для студентов направления 010500.62 "Математическое обеспечение и администрирование информационных систем", профили подготовки "Технологии программирования", "Параллельное программирование" / М. С. Воробьева. – Тюмень : Тюменский государственный университет, 2011. – 19 с.

Додатки

Додаток А
Таблиця А.1

Структура складових професійних компетентностей з навчальної дисципліни "Технології створення програмних продуктів" за Національною рамкою кваліфікацій України

Складові компетентності, яка формується в рамках теми	Мінімальний досвід	Знання	Вміння	Комунікації	Автономність і відповідальність
1	2	3	4	5	6
Тема 1. Базові поняття створення програмних продуктів					
Визначати способи формалізації й реалізації процесу створення ПП	Сутність і специфіка поняття розроблення програмних продуктів, життєвого циклу ПП. Основні моделі життєвого циклу ПП	Знання специфіки створення ПП, основних форм її реалізації, основних моделей життєвого циклу ПП	Ідентифікувати ключові створення ПП	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПП	Відповідальність за точну ідентифікацію ключових проблем створення ПП
Тема 2. Схематичне уявлення та сутність моделей життєвого циклу програмного забезпечення					
Визначати технологію створення ПП	Основні етапи та стадії створення ПП	Знання технологій створення ПП	Проводити декомпозицію проблеми. Формувати пріоритети цілей та критеріїв	Презентувати результати визначення найбільш ефективної методики створення ПП	Відповідальність за точність і коректність створення ПП

1	2	3	4	5	6
Тема 3. Класифікація та сутність процесів розроблення програмного забезпечення					
Здійснювати обґрунтований вибір методів та моделей створення ПП	Характеристика формалізованих та неформалізованих методів створення ПП	Знання методичних основ створення, та супроводу ПП	Здійснювати вибір методичного інструментарію для вирішення конкретної проблеми. Розробляти моделі ЖЦ для розв'язання конкретної проблеми	Презентувати результати побудови моделей ЖЦ	Самостійно приймати ефективні рішення та відповідати за коректність і адекватність розроблених моделей ЖЦ
Тема 4. Типова схема аналізу вимог					
Формувати напрями підвищення ефективності створення, тестування та супроводу ПП	Сутність обґрунтування методів створення ПП. Чинники, що впливають на вибір методів створення ПП	Знання критеріїв оцінки та принципів обґрунтування методів створення ПП	Здійснювати обґрунтування обрання методів та засобів створення ПП	Ефективно формувати комунікаційну стратегію	Приймати ефективні рішення та відповідати за надійність і точність результатів
Тема 5. Якість програмного забезпечення					
Проводити оцінювання якості створення ПП	Визначення поняття "якість ПП" та його основних принципів	Знання основних етапів оцінювання якості створення ПП	Будувати моделі результатів діяльності щодо управління якістю ПП	Презентувати результати тестування ПП	Самостійно приймати ефективні рішення щодо сценаріїв тестування ПП. Відповідати за коректність та адекватність розроблених сценаріїв

1	2	3	4	5	6
Тема 6. Схема реалізації модулів, стандарти та якість реалізації					
Проводити аналіз впливу невизначеності на якість ПП	Визначення поняття "невизначеність" та "ризик". Причини виникнення програмної помилки	Знання формалізованих підходів до опису природи програмної помилки	Визначати та формалізувати природу програмної помилки. Визначати взаємозв'язок невизначеності та ризику	Презентувати результати формалізації невизначеності. Здатність до групової взаємодії в процесі з'ясування причин виникнення програмної помилки	Приймати рішення в умовах виникнення програмної помилки та відповідати за точність і коректність результатів
Тема 7. Тестування програмного забезпечення					
Визначати ефективні методи та засоби тестування ПП	Визначення поняття "Тестування ПП" та його основних принципів	Знання методичних основ тестування ПП	Обґрунтовувати обрання методів та засобів тестування ПП	Презентувати результати тестування ПП	Приймати ефективні управлінські рішення щодо розробки сценаріїв тестування ПП
Тема 8. Супровід програмного забезпечення					
Визначати ефективні методи та засоби супроводу ПП	Визначення поняття "Супровід ПП". Види систем супроводу ПП	Знання призначення, функціональних можливостей та принципів супроводу ПП	Будувати системи супроводу ПП	Презентувати результати побудови систем супроводу ПП	Приймати ефективні управлінські рішення за допомогою систем супроводу ПП

Зміст

Вступ.....	3
1. Опис навчальної дисципліни	4
2. Мета та завдання навчальної дисципліни	4
3. Програма навчальної дисципліни	8
4. Структура навчальної дисципліни.....	12
5. Теми лабораторних занять.....	13
6. Самостійна робота.....	18
6.1. Контрольні запитання для самодіагностики	21
6.2. Перелік тем презентацій.....	29
7. Індивідуально-консультативна робота	34
8. Методи навчання	34
9. Методи контролю	37
10. Розподіл балів, які отримують студенти	54
11. Рекомендована література.....	60
11.1. Основна	60
11.2. Додаткова	60
11.3. Інформаційні ресурси.....	63
11.4. Методичне забезпечення	68
Додатки.....	69

НАВЧАЛЬНЕ ВИДАННЯ

**Робоча програма
навчальної дисципліни
"ТЕХНОЛОГІЇ СТВОРЕННЯ
ПРОГРАМНИХ ПРОДУКТІВ"**

**для студентів напряму підготовки
6.050101 "Комп'ютерні науки"
денної форми навчання**

Самостійне електронне текстове мережеве видання

Укладач **Скорін** Юрій Іванович

Відповідальний за видання *О. Г. Руденко*

Редактор *В. О. Бутенко*

Коректор *В. О. Бутенко*

План 2016 р. Поз. № 281 ЕВ. Обсяг 73 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А
*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*