

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНБАСЬКА ДЕРЖАВНА МАШИНОБУДІВНА АКАДЕМІЯ
Кафедра «Автоматизація виробничих процесів»

“ЗАТВЕРДЖУЮ”
Ректор ДДМА
В.Д. Ковальов
“ 04 ” 2020 року



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
„ ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ”
(назва дисципліни)

Галузь знань: 12 «Інформаційні технології»

Спеціальність 123 «Комп’ютерна інженерія»

Освітній рівень – перший (бакалаврський)

ОПП «Комп’ютерні системи та мережі»

Факультет «Машинобудування»

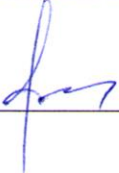
(назва інституту, факультету, відділення)

Робоча програма навчальної дисципліни «Інженерія програмного забезпечення» для студентів галузі знань 12 «Інформаційні технології» спеціальності 123 «Комп'ютерна інженерія».

Розробник: **Костіков О.А.**, канд. фіз.-мат. наук, доц.


Погоджено з групою забезпечення освітньої програми (для обов'язкових дисциплін).

Керівник групи забезпечення:


_____ О.В. Суботін, к.т.н., доцент

Розглянуто і затверджено на засіданні кафедри «Автоматизація виробничих процесів», протокол № 10 від 22.06.2020 року.

Завідувач кафедри АВП:


_____ Г.П. Клименко, д.т.н., професор

Розглянуто і затверджено на засіданні Вченої ради факультету машинобудування, протокол № 07 від 31.08.2020 року.

20/08
Голова Вченої ради факультету:


_____ В.Д. Кассов, д.т.н., професор

1.Опис навчальної дисципліни

Показники		Галузь знань, спеціальність, ОПП (ОНП), професійне (наукове) спрямування, рівень вищої освіти	Характеристика навчальної дисципліни	
			денна	прискорена
Кількість кредитів		Галузь знань: 12 «Інформаційні технології». Спеціальність: 123 «Комп'ютерна інженерія». ОПП «Комп'ютерні системи та мережі»	Обов'язкова дисципліна	
7,5	4			
Загальна кількість годин				
225	120			
Модулів – 2			Рік підготовки	
Змістових модулів –2			4	1
Індивідуальне науково-дослідне завдання : <u>Розробка проекту та реалізація програмної системи з використанням мови UML та Case засобу Visual Paradigm Community Edition</u>			Семестр	
Тижневих годин для <u>денної</u> форми навчання: аудиторних – 8; самостійної роботи студента – 7,5		Рівень вищої освіти: <u>перший (бакалаврський)</u>	Лекції	
			45	36
			Лабораторні	
				18
			Практичні	
			71	
			Самостійна робота	
			109	66
Вид контролю		екзамен	екзамен	

2. ЗАГАЛЬНІ ВІДОМОСТІ, МЕТА І ЗАВДАННЯ ДИСЦИПЛІНИ

Мета дисципліни - отримання загальних відомостей і орієнтація студентів в сутності такої області діяльності, як створення прикладного програмного забезпечення.

Завдання дисципліни - формування професійних компетенцій, що дозволяють вирішувати завдання професійної діяльності на основі знань, пов'язаних з методами виявлення вимог до прикладних систем, отримання функціональних вимог на основі вимог користувачів, визначення вимог до прикладної програмної системи, з моделюванням вимог і вибором адекватних методів проектування і стратегій тестування.

Програмні компетентності. Освітня компонента «Програмна інженерія» повинна сформулювати наступні загальні та фахові програмні компетентності, що передбачені освітньо-професійною програмою підготовки бакалаврів «Комп'ютерні системи та мережі»:

ЗК1. Здатність до абстрактного мислення, аналізу і синтезу.

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Навички міжособистісної взаємодії.

ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

ЗК8. Здатність працювати в команді.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж

ФК4. Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки.

ФК14. Здатність проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, включаючи створення, налаштування, експлуатацію, технічне обслуговування та утилізацію.

ФК15. Здатність аргументувати вибір методів розв'язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення

ФК18. Знання принципів, методів і алгоритмів комп'ютерної графіки, вміння застосовувати їх під час розробки графічних інтерфейсів взаємодії людини з комп'ютером.

Програмні результати навчання. Освітня компонента «Програмна інженерія» повинна сформулювати наступні програмні результати навчання, що передбачені освітньо-професійною програмою підготовки бакалаврів «Комп'ютерні системи та мережі»:

ПР1. Знати та розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж.

ПР3. Знати новітні технології в галузі комп'ютерної інженерії.

ПР11. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

ПР13. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПР15. Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державних та виробничих інтересів.

ПР17. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

Передумови для вивчення дисципліни: попереднє вивчення дисциплін «Комп'ютерні технології та програмування» та «Комп'ютерні мережі».

Мова викладання: українська.

Обсяг навчальної дисципліни та його розподіл за видами навчальних занять:

- загальний обсяг для денної форми навчання становить 165 годин/ 5,5 кредити, в тому числі: лекції - 30 годин, лабораторні роботи - 30 годин, самостійна робота студентів - 105 година;

- загальний обсяг для прискореної форми навчання становить 105 годин/ 3,5 кредити, в тому числі: лекції 36 годин, практичні заняття - 18 години, самостійна робота студентів - 66 годин.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1.

Основні заходи взаємодії у мережевому середовищі у контексті застосування ОС Linux

Тема 1. Вступ.

Визначення і основні поняття програмної інженерії. Роль програмної інженерії в практиці та дослідженнях. Важливість програмної інженерії для розробки програмного продукту.

Тема 2. Життєвий цикл програмного забезпечення

Визначення поняття життєвого циклу програмного забезпечення. Етапи життєвого циклу, їх цілі. Модель життєвого циклу. Традиційні моделі: водоспадна, модель на основі розробки прототипу, спіральна. Гнучкі моделі: екстремальне програмування(XP), SCRUM, інкрементальна модель RUP.

Тема 3. Командна робота над програмним проектом. Методи керування програмним проектом.

Основні учасники і рольові групи команди проекту. Організаційні структури управління проектом. Основні моделі управління командою проекту. Роль керівника в команді проекту. Мотивація програміста як учасника проекту. Менеджмент проекту. Основні поняття та задачі. Головні цілі менеджменту проекту. Модель процесу керування проектом. Інфраструктура програмного проекту. Методи керування і планування проектом. Метод критичного шляху - СРМ. Метод аналізу й оцінки проекту – PERT. Планування і контроль проекту. Оцінювання вартості проекту. Методи керування ризиками у проекті.

Тема 4. Об'єктна модель.

Об'єкти та виклик методів. Інтерфейси. Нотація класів в UML. Відношення між об'єктами. Порівняння структурного та об'єктно орієнтованого підходів до проектування програмного забезпечення. Визначальні методологічні ознаки якісного дизайну програмного забезпечення: простежуваність, тестовність, вимірюваність, безпека.

Тема 5. Конфігураційне керування.

Основні визначення. Процес керування конфігурацією по ISO/IEC12207. Системи управління версіями. Визначення "гілки" проекту. Управління збірками. Засоби версійного контролю. Одиниці конфігураційного управління. Поняття baseline.

Тема 6. Інженерія вимог.

Визначення вимоги та інженерії вимог. Стейкхолдери. Типи вимог. Трасування вимог. Зв'язок між вимогами. Функціональні і нефункціональні вимоги.

Тема 7. Архітектура програмного забезпечення.

Визначення архітектури програмного забезпечення. Декомпозиція систем. Концептуальна цілісність системи. Компоненти програмного забезпечення. UML нотація для компонентів програмного забезпечення. Популярні архітектурні стилі: архітектура бази даних Central Repository(Database), клієнт-серверна архітектура, REST, peer-to-peer, мікросервіси.

Тема 8. Варіанти використання(Use Cases).

Актори та елементи Use Case. Визначення Use Case із системних вимог. Діаграми Use Case. Відношення в діаграмах Use Case. Робота з елементами Use Case. Специфікація елементів Use Case. Матриця відповідності вимог(Traceability matrix).

Тема 9. Аналіз предметної області та моделювання.

Глосарій предметної області. Порівняння Use Case з моделлю предметної області. Побудова моделі предметної області із UseCases. Види класів аналізу: граничний, керуючий, сутність та їх визначення. Діаграма класів аналізу, її призначення і склад. Визначення атрибутів, асоціацій та відповідальності класів аналізу. Правила і рекомендації з розробки діаграм класів аналізу.

Тема 10. Проектування: розподіл відповідальностей.

Взаємодія об'єктів. Діаграма послідовностей. Характеристики професійного дизайну. Розподіл відповідальностей. Проектування на основі відповідальності. Бізнес правила. Діаграма класів.

Тема 11. Принципи об'єктно-орієнтованого проектування.

SOLID принципи: принцип єдиної відповідальності, принцип відкритості/закритості, принцип підстановки Барбара Лісков, принцип розділення інтерфейсу, принцип інверсії залежностей. Мета застосування принципів проектування. Приклади використання.

Тема 12. Визначення систем.

Предметна область, явища. Стани. Мікростани і макростани. Події. Контекстні діаграми. Системи та опис систем. Основні формалізми для специфікації: булева логіка, автомати з кінцевою кількістю станів.

Тема 13. Діаграми станів та об'єктна мова обмежень OCL.

Діаграми автоматів UML: нотація. Діяльності станів: діяльність при вході:(entry activity), діяльність при виході(exit activity), діяльність в стані(do activity). Об'єктна мова обмежень OCL: типи обмежень та операції. Навігація в OCL. Доступ до колекцій в OCL. Класифікація обмежень:інваріант класу inv:, передумова операції pre:, постумова операції post:, тіло запиту body:, початкове значення атрибуту або з'єднання init:, правило виводу derive:, додаткове обмеження def:.

Тема 14. Архітектурне проектування.

Діаграми пакетів. Діаграми компонентів. Порівняльний аналіз пакетів і компонентів. Порівняльний аналіз компонентів, класів та інтерфейсів.

Тема 15. Детальне проектування.

Діаграми класів. Атрибути, операції класів. Множинність. Відношення в діаграмах класів: асоціація, узагальнення, залежність, реалізація, агрегація, композиція. Діаграми діяльності. Призначення і склад діаграми діяльності. Правила та рекомендації побудови діаграми діяльності. Основні принципи детального проектування. Принципи пакування класів в архітектурні підсистеми. Документування процесу проектування. Діаграми розгортання.

Тема 16. Архітектурні патерни.

Патерни Спостерігач, Компонувальник, Команда, Фасад, Абстрактна Фабрика, Ланцюжок Відповідальностей, Посередник.

Тема 17. Проектування інтерфейсу користувача.

Суттєвостна ефективність. Узгодженість задач. Спостереження задач. Однаковість компонування. Візуальна зв'язність.

Тема 18. Проектування бази даних.

Зберігання бізнес-об'єктів. Модель реляційної бази даних. Відображення атрибутів об'єктів і класів в реляційну базу даних. Відображення відношень об'єктів в реляційну базу даних.

Тема 19. Тестування і налагодження програм.

Терміни і визначення. Тестування методами "білої шухляди" і "чорної шухляди". Порядок розробки тестів. Автоматизація тестування. Модульне тестування. Інтеграційне тестування. Системне тестування. Ефективність і оптимізація програм. Налagodження програм.

Тема 20. Метрики програмного забезпечення.

Необхідність вимірювання програмного забезпечення. Основи теорії вимірювань. Точки варіантів використання. Фактор технічної складності (ТСФ). Фактор складності навколишнього середовища(ЕСФ).

Тема 21. Цикломатична складність.

Формальне визначення. Обмеження складності під час розробки. Вимірювання «структурованості» програми. Значення для тестування програмного забезпечення.

Тема 22. Модель якості програмних систем.

Стандартні показники якості. Метрики якості. Стандартна оцінка показників якості. Керування якістю програмних систем. Моделі оцінки надійності програмних систем. Ґрунтовні поняття проблематики надійності. Класифікація моделей надійності. Марковські та пуассонівські моделі надійності.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

4.1 Розподіл обсягу дисципліни за семестрами.

Семестр	Всього		Розподіл за семестрами та видами занять, год					
	Годин	Кредитів	Лекц.	Практичн.	Лаб.роб.	Курсова робота	СРС	Семестрова атестація
Повний курс								
7	180	6	45	45			90	Екзамен
8	45	1,5				45		Диф.залік
Всього	225	7,5	45	45		45	90	
Прискорений курс								
4	90	3	39		13		38	Екзамен
4	30	1				30		Диф.залік
Всього	120	4	39		13	30	38	

4.1 Розподіл обсягу дисципліни за видами навчальних занять та темами

№ з/п	Назви змістових модулів і тем	Кількість годин				
		Усього	в т.ч.			
			Л	П (С)	Лаб	СРС
Змістовий модуль 1						
Вступ. Об'єктно-орієнтовна програмна інженерія.						
1.	Тема 1. Вступ.	8	2			6
2.	Тема 2. Життєвий цикл програмного забезпечення	8	2	2		2
3.	Тема 3. Командна робота над програмним проектом. Методи керування програмним проектом.	9	3	2		4
4.	Тема 4. Об'єктна модель.	8	2	2		4
5.	Тема 5. Конфігураційне керування.	8	2	2		4
6.	Тема 6. Інженерія вимог.	8	2	2		4
7.	Тема 7. Архітектура програмного забезпечення.	8	2	3		4
8.	Тема 8. Варіанти використання(Use Cases)..	8	2	2		4
9.	Тема 9. Аналіз предметної області та моделювання.	8	2	2		4

№ з/п	Назви змістових модулів і тем	Кількість годин				
		Усього	В т.ч.			
			Л	П (С)	Лаб	СРС
10.	Тема 10. Проектування: розподіл відповідальностей.	9	2	3		4
11.	Тема 11. Принципи об'єктно-орієнтованого проектування.	8	2	2		4
Змістовий модуль 2						
Моделювання та специфікація системи.						
Вимірювання та оцінка програмного забезпечення.						
12	Тема 12. Визначення систем.	8	2	2		4
13	Тема 13. Діаграми станів та об'єктна мова обмежень OCL.	8	2	2		4
14	Тема 14. Архітектурне проектування.	10	2	2		6
15	Тема 15. Детальне проектування.	10	2	3		4
16	Тема 16. Архітектурні патерни.	8	2	2		4
17	Тема 17. Проектування інтерфейсу користувача.	8	2	2		4
18	Тема 18. Проектування бази даних.	8	2	2		4
19	Тема 19. Тестування і налагодження програм.	8	2	2		4
20	Тема 20. Метрики програмного забезпечення.	8	2	2		4
21	Тема 21. Цикломатична складність.	8	2	2		4
22	Тема 22. Модель якості програмних систем.	8	2	2		4
Всього		180	45	45		90
Курсова робота						
Тема роботи: «Розробка проекту та реалізація програмної системи з використанням мови UML та Case засобу Visual Paradigm Community Edition»		45		26		19
Всього		45		26		19
Разом		225	45	71		109

Л – лекції, П (С) – практичні (семінарські) заняття, Лаб – лабораторні заняття, СРС – самостійна робота студентів.

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Тема	Назва практичної роботи	Кількість годин
Змістовий модуль 1			
Вступ. Об'єктно-орієнтовна програмна інженерія.			
1	Тема 2	Розробка технічного завдання на програмне забезпечення	2
2	Тема 3	Командна робота. Формулювання клієнтських вимог до програмного продукту. Створення глосарію термінів.	2
3	Тема 4	Побудова об'єктної моделі предметної області	2
4	Тема 5	Розробка структури програмного проекту. Інтеграція проекту з системою управління версіями коду Git	2
5	Тема 6	Командна робота. Визначення функціональних і нефункціональних вимог до проєктованої програмної системи.	2
6	Тема 7	Архітектура програмного забезпечення	3
7	Тема 8	Командна робота. Специфікація функціональних вимог.	2
8	Тема 9.	Командна робота. Аналіз предметної області	2
9	Тема 10	Командна робота. Побудова діаграми взаємодії за допомогою CASE засобу Visual Paradigm Community Edition.	3
10	Тема 11.	Принципи проєктування класів(SOLID)	2
Змістовий модуль 2			
Моделювання та специфікація системи.			
Вимірювання та оцінка програмного забезпечення.			
11	Тема 12	Побудова контекстної діаграми предметної області	2
12	Тема 13	Командна робота. Специфікація угод OCL.	2
13	Тема 14	Командна робота. Архітектура та дизайн системи.	2
14	Тема 15	Командна робота. Розробка діаграми класів та специфікацій інтерфейсів в Visual Paradigm Community Edition.	3
15	Тема 16	Командна робота. Використання патернів проєктування.	2
16	Тема 17	Командна робота. Проєктування інтерфейсу користувача	2

№ з/п	Тема	Назва практичної роботи	Кількість годин
17	Тема 18	Командна робота. Проектування бази даних.	2
18	Тема 19	Командна робота. Розробка тестів	2
19	Тема 20	Бальна оцінка варіантів використання	2
20	Тема 21	Тестування програми методом «білої скриньки». Обчислення цикломатичної складності ПЗ.	2
21	Тема 22	Якість програмного забезпечення.	2
Усього годин			45

6. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Ціль індивідуальних завдань - формування навиків та вмінь у використанні методики проектування та реалізації програмного забезпечення.

Метою виконання курсового проекту є закріплення та поглиблення знань, отриманих в процесі вивчення курсу «Інженерія програмного забезпечення», набуття практичних навичок та вмінь подальшого їх використання для проектування і розробки програмного забезпечення.

Основними завданнями курсового проектування є:

- узагальнення теоретичних знань, отриманих під час вивчення дисципліни «Інженерія програмного забезпечення», за допомогою поглибленого вивчення додаткової фахової літератури;
- набуття навичок практичного застосування теоретичних знань, проведення дослідження та аналізу існуючих програмних систем та розробка програмного забезпечення;
- набуття практичних вмінь постановки інженерних задач, проектування складних систем та їх реалізація, розробка супровідної технічної документації до розробленого проекту.

7. САМОСТІЙНА РОБОТА

№ з/п	Назва теми	Кількість годин
1	Характеристика областей знань з інженерії програмного забезпечення - SWEBOOK	4
2	Технологія екстремального програмування. SCRUM технологія. Організація колективної роботи над проектом при використанні технологій швидкої розробки.	4

№ з/п	Назва теми	Кількість годин
3	Класифікація вимог та форми їх подання. Методи визначення вимог у програмній інженерії	4
4	Рекомендації щодо оформлення специфікації вимог до програмного забезпечення	4
5	Виявлення вимог на основі етнографічного дослідження	4
6	Специфікація вимог в екстремальному програмуванні	5
7	Програмний інструментарій підтримки виконання проектів	5
8	Керування ризиками. Діяльності, пов'язані з оцінкою.	5
9	Керування ризиками. Діяльності, пов'язані контролем і управлінням.	5
10	GIT: Архітектура та команди.	5
11	Основи роботи з CASE засобом Visual Paradigm Community Edition	5
12	Проектування різних видів архітектур програмних систем.	5
13	Рефакторинг програмних систем	5
14	Проектування систем на основі раніше розроблених компонент	5
15	Інструменти тестування. Критерії тестування.	5
16	Керування якістю програмних систем. Моделі оцінки надійності програмних систем.	5
17	Основні завдання, що вирішуються на етапі впровадження. Процес усунення помилок на етапі впровадження.	5
18	Документування програмного забезпечення.	5
19	Організація та методи супроводу програмних засобів. Технічна підтримка користувачів на етапі супроводу.	5
Разом		90

8. МЕТОДИ НАВЧАННЯ

В навчальному процесі застосовуються: лекції з використанням мультимедіа матеріалів; практичні роботи в комп'ютерному класі з пошуком інформації в Інтернет; індивідуальні заняття та самостійна робота.

9. МЕТОДИ КОНТРОЛЮ

При вивченні дисципліни використовуються наступні методи контролю: проведення поточного контролю, письмового підсумкового контролю у вигляді екзамену, модульного контролю.

10. КОНТРОЛЬНІ РОБОТИ

Контрольні роботи з теоретичної частини розподілені таким чином:

№ роботи	№ теми	Тема контрольної роботи	Кількість варіантів
1	1-5	Об'єктно-орієнтовна програмна інженерія	20
2	6-14	Моделювання та специфікація системи. Вимірювання та оцінка програмного забезпечення.	20

11. КРИТЕРІЇ ОЦІНЮВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

8.1. Розподіл балів, які отримують студенти (кількісні критерії)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовий модуль 1			
Виконання практичних робіт	0...4	10	0...40
Модульний контроль	0...10	1	0...10
Змістовий модуль 2			
Виконання практичних робіт	0...4	10	0...40
Модульний контроль	0...10	1	0...10
Усього			55... 100

8.2. Критерії оцінки знань студентів протягом семестру

8.2.1 Критерії оцінки знань за практичні роботи

Вимоги	Кількість балів
Завдання відзначається повнотою виконання без допомоги викладача. Визначає рівень поінформованості, потрібний для прийняття рішень. Вибирає інформаційні джерела. Робить висновки і приймає рішення у ситуації невизначеності. Володіє уміннями творчо-пошукової діяльності.	4
Завдання - повні, з деякими огріхами, виконані без допомоги викладача. Планує інформаційний пошук: володіє способами систематизації інформації. Студент може зіставити, узагальнити, систематизувати	3

Вимоги	Кількість балів
інформацію під керівництвом викладача: вільно застосовує вивчений матеріал у стандартних ситуаціях. Робить висновки і приймає рішення у ситуації невизначеності. Володіє уміннями творчо-пошукової діяльності.	
Завдання відзначається неповнотою виконання за консультацією викладача. Застосовує запропонований викладачем спосіб отримання інформації, має фрагментарні навички в роботі з підручником, науковими джерелами: Вибирає відомі способи дій для виконання фахових методичних завдань.	1-2

8.2.2 Критерії оцінювання модульної контрольної роботи

9-10 балів виставляється, коли студент дає абсолютно правильні відповіді на теоретичні питання з викладенням оригінальних висновків, отриманих на основі програмного, додаткового матеріалу та нормативних документів. При виконанні практичного завдання студент застосовує системні знання навчального матеріалу, передбачені навчальною програмою.

8 балів виставляється студенту, який повністю розкрив теоретичні питання на основі програмного та додаткового матеріалу. При виконанні практичних завдань студент застосовує узагальнені знання навчального матеріалу, передбачені навчальною програмою.

7 балів виставляється студенту, який повністю розкрив теоретичні питання, а програмний матеріал викладено у відповідності до вимог. Практичні завдання виконані в цілому правильно, але мають місце окремі неточності.

6 балів виставляється, коли студент розкрив теоретичні питання, проте при викладенні програмного матеріалу допущені окремі помилки. При виконанні практичних завдань студент припускається помилок, за рахунок недостатнього розуміння програмного матеріалу.

5 балів виставляється, коли студент неповністю розкрив теоретичні питання, відповідь містить суттєві помилки. При виконанні практичних завдань студент припускається значних помилок, а виконання завдань викликає значні труднощі у студента.

3-4 бали виставляються студенту, який не розкрив теоретичні питання і не може виконати практичні завдання. Як правило такий студент виявляє здатність до викладення думки лише на елементарному рівні.

0-2 бали виставляється студенту, який не виконав навчальну програму або якийсь елемент її складової, має фрагментарні знання, які не дозволяють розкрити теоретичні питання і виконати практичні завдання. Такий студент не може викласти свою думку навіть на елементарному рівні.

8.3. Підсумкова оцінка за семестр.

Контроль знань здобувачів здійснюється за рейтинговою накопичувальною (100-бальною) системою, яка передбачає складання обов'язкових контрольних точок.

Підсумкова оцінка за дисципліну складається з оцінки, отриманої на протязі семестру, та оцінки, отриманої на екзамені.

Після завершення семестру проводиться семестровий екзамен, завданням якого є перевірка розуміння студентом програмного матеріалу в цілому, логіки та взаємозв'язків між окремими розділами, здатності творчого використання накопичених знань, вміння формулювати своє ставлення до певної проблеми навчальної дисципліни тощо.

Максимальна кількість балів (100) при оцінюванні знань з навчальної дисципліни, яка завершується екзаменом, формується з двох частин, з коефіцієнтом 0,5 кожна:

– за поточну успішність 100 балів (сума балів, зароблена у семестрі, але не менше 55);

– на екзамені 100 балів (мінімально необхідна кількість балів за екзамен 55).

Підсумкові оцінки за триместр в цілому переводяться за національною шкалою та шкалою ECTS відповідно до таблиці перекладу, яка визначається діючим в ДДМА положення про організацію навчального процесу в кредитно-модульній системі підготовки фахівців:

Рейтингова оцінка	У національній шкалі	У шкалі ECTS
90-100	Відмінно (зараховано)	A
81-89	Добре (зараховано)	B
75-80	Добре(зараховано)	C
65-74	Задовільно (зараховано)	D
55-64	Задовільно (зараховано)	E
30-54	Незадовільно (не зараховано)	FX
0-29	Незадовільно (не зараховано)	F

Для отримання позитивної оцінки з дисципліни студент повинен одержати не менше ніж 55 балів сумарної оцінки.

8. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ

1. Інженерія програмного забезпечення. Конспект лекцій (для студентів спеціальності 123 «Комп'ютерна інженерія»).

2. Методичні вказівки до виконання лабораторних робіт з дисципліни "Інженерія програмного забезпечення" (для студентів спеціальності 123 «Комп'ютерна інженерія»).

9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Література основна

1. Pressman, Roger, Maxim, Bruce. Software Engineering: A Practitioner's Approach. – NY: McGraw-Hill Education, 2019. – 705p.
2. Sommerville, Ian. Software Engineering, 10th Edition. - Pearson, 2016. – 811p.
3. Voorhees, David. Guide to Efficient Software Design An MVC Approach to Concepts, Structures, and Models. - Springer Nature Switzerland AG, 2020. – 519p.
4. Bruegge, Bernd, Dutoit, Allen. Object-oriented software engineering : using UML, Patterns, and Java. - Harlow, UK: Pearson Education Limited, 2014. – 723 p.
5. Gaopande, Laxmidhar. Software Engineering: A Practical Approach. – 2020.- 241p.
6. Marsic, Ivan. Software Engineering. - Rutgers University, New Brunswick, New Jersey, 2012. – 627p.
7. Бородкіна, Ірина, Бородкін, Георгій. Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів.- К.: ТОВ «Видавництво "Центр навчальної літератури"», 2018. – 204 с.
8. Лавріщева К.М. Програмна інженерія.–К.– 2008.–319 с.
9. Орлов, С. А. Программная инженерия. Учебник для вузов, 5-е изд.– СПб:Питер, 2016. – 640с.
10. Мацяшек, Л. А., Лионг, Б.Л. Практическая программная инженерия на основе учебного примера — М. : БИНОМ. Лаборатория знаний, 2015. — 959с.
11. Мацяшек, Лешек. Анализ и проектирование информационных систем с помощью UML 2.0,3-е изд. : Пер. с англ. — М. : ООО “И.Д. Вильямс”, 2008. — 816 с.

Література додаткова

1. Ларман К. Применение UML 2.0 и шаблонов проектирования : практическое руководство / Ларман К.: пер. с англ. – М. ООО «И.Д. Вильямс», 2009. – 736 с.
2. Фаулер М. UML. Основы, 3-е изд. – СПб.: Символ-Плюс, 2005. – 184с.
3. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2006. – 192 с.
4. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose: Учебное пособие. – М.: Интернет-Университет Информационных Технологий; БИНОМ, Лаборатория знаний, 2006. – 320 с.

Інформаційні ресурси в мережі Інтернет

1. Офіційний сайт спільноти користувачів CASE – засобу Visual Paradigm [Електрон-ний ресурс:]. Режим доступу // <https://www.visual-paradigm.com/>

2. Стандарт UML 2.0 [Електронний ресурс:]. Режим доступу // <http://www.omg.org/spec/UML/2.0/>

ДОДАТОК А

СТРУКТУРА КУРСОВОГО ПРОЕКТУ

Курсовий проект з дисципліни «Інженерія програмного забезпечення» виконується з метою формування у студентів здатності проектувати системи та їхні компоненти з урахуванням усіх аспектів їх життєвого циклу та поставленої задачі, освоєння методики об'єктно-орієнтованого моделювання програмних систем на мові UML (Unified Modeling Language), а також придбання навичок роботи з сучасним CASE-засобом Visual Paradigm.

Предметом проектування є програмна система, призначена для автоматизації процесів в певній предметній області

Зміст курсового проекту має розкривати суть етапів створення програми для заданої предметної області. Основна частина розрахунково-пояснювальної записки повинна розкривати тему в повному обсязі і складатися з наступних розділів:

- ✓ Постановка задачі
- ✓ Системні вимоги
- ✓ Специфікація функціональних вимог
- ✓ Аналіз предметної області
- ✓ Діаграми взаємодії
- ✓ Діаграми класів та специфікації інтерфейсів
- ✓ Архітектура та проект системи
- ✓ Алгоритми та структури даних
- ✓ Проектування та впровадження інтерфейсу користувача
- ✓ Кодування та тестування програмного забезпечення

Додаток Б

Питання для підготовки до контрольної роботи та екзамену з дисципліни «Інженерія програмного забезпечення»

1. Технологія розробки програмного забезпечення як галузь інженерної науки.
2. Історичний та сучасний погляди на розробку програмного забезпечення.
3. Програмна інженерія.
4. Аналіз і характеристика областей знань SWEBOOK.
5. Основи програмних вимог.
6. Проектування ПЗ.
7. Конструювання ПЗ.
8. Тестування ПЗ.
9. Супроводження ПЗ.
10. Управління конфігурацією ПЗ.
11. Управління інженерією ПЗ.
12. Процес інженерії ПЗ.
13. Методи і засоби інженерії ПЗ.
14. Якість ПЗ.
15. Вступ в життєвий цикл ПЗ.
16. Каскадна модель ЖЦ.
17. Інкрементна модель ЖЦ.
18. Спіральна модель.
19. Еволюційна модель ЖЦ.
20. Стандартизована модель системи.
21. Співставлення моделі ЖЦ стандарту ISO/IEC 12207 і областей знань SWEBOOK.
22. Характеристика процесів стандарту. Характеристика моделі процесів в ядрі SWEBOOK.
23. Визначення понять і видів вимог.
24. Види вимог.
25. Аналіз і збір вимог.
26. Інженерія вимог ПЗ.
27. Верифікація і формалізація вимог.
28. Об'єктно-орієнтована інженерія вимог.
29. Модель аналізу вимог. Визначення об'єктів.
30. Класифікація вимог.
31. Об'єктно-орієнтовані методи аналізу і побудови моделей предметної області.
32. Основні поняття аналізу предметної області.
33. Інформаційна модель і модель станів. Модель процесів.
34. Методи проектування архітектури ПЗ.
35. Стандартний підхід до проектування системи.
36. Загальносистемний підхід до проектування архітектури системи.

37. Технічне проектування.
38. Методи систематичного програмування.
39. Структурний підхід до проектування
40. Об'єктно-орієнтований метод проектування.
41. Метод моделювання UML.
42. Компонентний підхід до проектування.
43. Методологія компонентної розробки систем.
44. Аспектно-орієнтоване програмування.
45. Методи теоретичного програмування.
46. UML як інструмент моделювання.
47. UML як мова візуалізації.
48. UML як мова специфікації.
49. UML як мова конструювання.
50. UML як мова документування.
51. Сутність UML.
52. Відношення UML.
53. Діаграми UML.
54. Правила мови UML.
55. Загальні механізми мови UML.
56. Загальна характеристика CASE-засобу Visual Paradigm.
57. Особливості робочого інтерфейсу програми Visual Paradigm.
58. Вікно браузера проекту Visual Paradigm.
59. Діаграми варіантів використання
60. Особливості розробки діаграм варіантів використання в середовищі Visual Paradigm.
61. Діаграми класів
62. Особливості розробки діаграм класів у середовищі Visual Paradigm.
63. Діаграми кооперації
64. Особливості розробки діаграм кооперації в середовищі Visual Paradigm.
65. Діаграми послідовності
66. Особливості розробки діаграми послідовності в середовищі Visual Paradigm. Діаграми станів.
67. Особливості розробки діаграми станів середовищі Visual Paradigm.
68. Діаграми діяльності
69. Особливості розробки діаграми діяльності у середовищі Visual Paradigm.
70. Діаграми компонентів.
71. Діаграми розгортання
72. Особливості розробки діаграми компонентів у середовищі Visual Paradigm.
73. Особливості розробки діаграми розгортання у середовищі Visual Paradigm.
74. Інженерія повторного використання компонент.
75. Різновиди ПВК.
76. Специфікація повторного використання компонент.

77. Репозитарій компонентів.
78. Інженерія додатків і предметної області.
79. Опис інтерфейсу об'єктів-компонентів в розподіленому середовищі.
Інженерія оцінювання якості реалізації предметної області із компонентів
80. Елементи управління комунікаціями.
81. Планування комунікацій.
82. Аналіз вимог до комунікацій.
83. Засоби комунікації.
84. План управління комунікаціями.
85. Розповсюдження інформації.
86. Навики комунікації.
87. Системи збору і вибірки інформації.
88. Способи поширення інформації.
89. Аналіз накопичених знань.
90. Активи організаційного процесу.
91. Звітність про виконання.
92. Інструменти представлення інформації.
93. Збір і зведення інформації про виконання робіт.
94. Наради по оцінці поточного стану.
95. Звіти про виконання.
96. Управління учасниками проекту.
100. Методи комунікації.