МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДОНБАСЬКА ДЕРЖАВНА МАШИНОБУДІВНА АКАДЕМІЯ

Кафедра «Автоматизація виробничих процесів»

Затверджую:
Декан факультету
машинобудування
_____ Кассов В.Д.
«27» травня 2024р.

Гарант освітньої програми:
к.т.н., доцент
_____ Разживін О.В.
«08» травня 2024р.

Розглянуто і схвалено
на засіданні кафедри автоматизації
виробничих процесів
Протокол №_13 від 06.05.2024р.
Зав. кафедри
_____Марков О.Є.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇДИСЦИПЛІНИ

„ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ СКЛАДНИХ СИСТЕМ ”

(назва дисципліни)

Галузь знань 15 – «Автоматизація та приладобудування»

Спеціальність 151 – «Автоматизація та комп'ютерно-інтегровані технології»

Освітній рівень перший (бакалаврський)

ОПП «Автоматизація та комп'ютерно-інтегровані технології»

Факультет «Машинобудування»
(назва інституту, факультету, відділення)

КРАМАТОРСЬК-ТЕРНОПІЛЬ, 2024

Робоча навчальна програма дисципліни «Технологія програмування складних систем» для студентів першого (бакалаврського) рівня за ОПП 151 «Автоматизація та комп'ютерно-інтегровані технології» галузі знань 15 «Автоматизація та приладобудування» спеціальності 151 «Автоматизація та комп’ютерно-інтегровані технології». - 16 с.

Розробник Картамишев Д. О., к.т.н., асистент _____

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

Керівник групи забезпечення

_____ О.В. Разживін, к.т.н., доцент

Розглянуто і затверджено на засіданні кафедри «Автоматизація виробничих процесів», протокол № 13 від 06.05.2024 року.

Зав кафедри АВП:

_____ О.Є. Марков, д.т.н., професор

Розглянуто і затверджено на засіданні Вченої ради факультету машинобудування, протокол № 10-24/05 від 27.05.2024 року

Голова Вченої ради факультету

_____ В.Д. Кассов, д.т.н., професор

І. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

<table>
<tr><td colspan="2" rowspan="2">Показники</td><td rowspan="2">Галузь знань, спеціальність, ОПП (ОНП), професійне (наукове) спрямування, рівень вищої освіти</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td>Денна повна/прискор</td><td>Заочна повна/прискор</td><td>Денна повна/прискорена</td><td>Заочна повна/прискорена</td></tr>
<tr><td colspan="2">Кількість кредитів</td><td rowspan="4">Галузь знань:
«15 «Автоматизація та приладобудування».
Спеціальність:
151 «Автоматизація та комп’ютерно-інтегровані технології»</td><td colspan="2" rowspan="4">Обов’язкова дисципліна</td></tr>
<tr><td>7,5/3,0</td><td>7,5/3,0</td></tr>
<tr><td colspan="2">Загальна кількість годин</td></tr>
<tr><td>225/90</td><td>225/90</td></tr>
<tr><td colspan="2">Модулів – 2</td><td rowspan="3">ОПП
«Автоматизація та комп'ютерно-інтегровані технології»</td><td colspan="2">Рік підготовки</td></tr>
<tr><td colspan="2">Змістових модулів–2</td><td>4 / 2</td><td>5 / 3</td></tr>
<tr><td colspan="2">Індивідуальне завдання:
Розробка проекту та реалізація програмної системи з використанням мови UML та Case засобу Visual Paradigm Community Edition</td><td colspan="2">Семестр</td></tr>
<tr><td colspan="2" rowspan="2">Тижневих годин
для денної форми навчання:
аудиторних – 6;
самостійної роботи студента – 8
для денної прискореної форми навчання:
аудиторних – 4;
самостійної роботи студента – 3
для заочної повної форми навчання:
аудиторних – 4;
самостійної роботи студента – 12,9
для заочної прискореної форми навчання:
аудиторних – 4;
самостійної роботи студента – 4,9</td><td rowspan="2">Рівень вищої освіти:
перший (бакалаврський)</td><td>7,8 / 4</td><td>9 / 5</td></tr>
<tr><td colspan="2">Лекції
30 / 26 | 12 / 12
Лабораторні
0 / 0 | 0
Практичні
30 / 26 | 12 / 4
Самостійна робота
120 / 38 | 156 / 74
Вид контролю
Іспит
Курсовий проект
Практичні | Практичні
26 / 0 | 8/0
Самостійна робота | Самостійна робота
19 / 0 | 37 / 0
Вид контролю: діф. залік</td></tr>
</table>

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної повної форми навчання – 86/139

для денної прискореної форми навчання – 52/38

для заочної повної форми навчання – 32/193

для заочної прискореної форми навчання – 16/74

ІІ. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Для вирішення завдань комплексної автоматизації управління промисловим підприємством застосовуються складні програмні системи з ієрархічною архітектурою, в якій технологічний, виробничий та адміністративні рівні програмного забезпечення повинні охоплювати певні функції та мати відповідні засоби взаємодії з різноманітними програмними додатками.

Мета викладання дисципліни «Технологія програмування складних систем» є отримання загальних відомостей і орієнтація студентів в сутності такої області діяльності, як створення прикладного програмного забезпечення.

Дисципліна «Технологія програмування складних систем» відноситься до вибіркового циклу загальних дисциплін з напрямку 151 «Автоматизація та комп'ютерно-інтегровані технології».

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 системі, з моделюванням вимог і вибором адекватних методів проектування і стратегій тестування.

Завдання дисципліни полягає у формуванні здатностей студентів:

Знати:

– розуміти наукові положення, що лежать в основі функціонування комп’ютерних засобів, систем та мереж.

– новітні технології в галузі технології проектування складних систем.

– методи та техніки моделювання програмних систем на різних рівнях абстракції

– CASE-інструменти для ефективного моделювання, на прикладі використання середовища Visual Paradigm.

– основні функціональні можливості CASE-інструментів для автоматизації створення та супроводу програмних систем

– концепції та принципи об'єктно-орієнтованої парадигми розробки програмного забезпечення.

– роль і значення уніфікованої мови моделювання Unified Modeling Language (UML) у процесах проектування програмного забезпечення.

– принципи Rational Unified Process (RUP)

– принципи використання UML для документування архітектурних шаблонів

Вміти:

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

– ефективно працювати як індивідуально, так і у складі команди.

– поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням виробничих інтересів.

– оцінювати отримані результати та аргументовано захищати прийняті рішення.

– визначати та аналізувати складності програмних систем через візуальне моделювання

– застосовувати UML для моделювання взаємодій між компонентами програмного забезпечення

– аналізувати та синтезувати проектні рішення на основі вимог до програмного продукту

– використовувати візуальні засоби моделювання для оптимізації процесів розробки ПЗ

– визначати та аналізувати складності програмних систем через візуальне моделювання

Опанувати навиками:

– використання UML для створення детальних моделей та діаграм, що допоможе в аналізі та візуалізації складних взаємозв'язків у програмному проекті;

– інтеграції теоретичних знань та практичних навиків при використанні CASE-інструментів, зокрема Visual Paradigm для ефективної автоматизації процесів розробки;

– комунікації з різними зацікавленими сторонами проекту за допомогою стандартизованих UML-діаграм, що сприяє чіткому розумінню вимог та взаємодій між компонентами системи;

– проектування та реалізації тестових сценаріїв на основі UML-моделей, що включає розробку випробувальних процедур і валідацію поведінкових моделей;

– аналізу програмних систем на основі створених моделей, використовуючи методики системного аналізу для виявлення потенційних проблем та оптимізації проектних рішень.

Передумови для вивчення дисципліни:

Комп’ютерні технології та програмування, Технічні засоби автоматизації, Проектування систем управління на базі ПЛК, Проектування систем автоматизації

Мова викладання – українська.

Обсяг навчальної дисципліни та його розподіл за видами навчальних занять:

- загальний обсяг для денної форми навчання становить 225 годин (7,5 кредити), в тому числі: лекції – 30 годин, практичні роботи – 30 годин, практичні з курсової роботи – 26, самостійна робота студентів – 139 годин.

- загальний обсяг для денної (прискореної) форми навчання становить 90 годин (3,0 кредити), в тому числі: лекції – 26 годин, практичні роботи – 26 годин, самостійна робота студентів – 38 годин.

- загальний обсяг для заочної форми навчання становить 225 годин (7,5 кредити), в тому числі: лекції – 12 годин, практичні роботи – 12 години, практичні з курсової роботи – 8, самостійна робота студентів – 193 годин

- загальний обсяг для заочної прискореної форми навчання становить 90 годин (3,0 кредити), в тому числі: лекції – 12 годин, практичні роботи – 4 години, самостійна робота студентів – 74 годин.

ІІІ ПРОГРАМНІ РЕЗУЛЬТАТИ НАВЧАННЯ

Освітня компонента «Технологія програмування складних систем» повинна сформувати наступні **програмні результати** навчання, що передбачені освітньо-професійною програмою підготовки бакалаврів «Автоматизація та комп'ютерно-інтегровані технології»:

ПРН03. Вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп’ютерні програми з використанням мов високого рівня та технологій об’єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси.

ПРН09. Вміти проектувати багаторівневі системи керування і збору даних для формування бази параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу, використовуючи новітні комп’ютерно-інтегровані технології

ПРНД1. Оцінювати ризики та здійснювати запобіжні дії їх уникнення, вести професійну діяльність з урахуванням доброчесності та авторського права.

ПРНД2. Усвідомлювати необхідність навчання та саморозвитку продовж усього життя з метою поглиблення знань .

ПРНД3. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

У результаті вивчення навчальної дисципліни «Технологія програмування складних систем» студент повинен продемонструвати достатній рівень сформованості певних результатів навчання через здобуття наступних **програмних компетентностей**:

ІК. Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми, що характеризуються комплексністю та невизначеністю умов, під час професійної діяльності у галузі автоматизації, або у процесі навчання, що передбачає застосування теорій та методів галузі.

ЗК01. Здатність застосовувати знання у практичних ситуаціях.

ЗК05. Здатність до пошуку, опрацювання та аналізу інформації з різних джерел.

ЗКД1. Здатність діяти свідомо та соціально-відповідально за результати прийняття стратегічних рішень.

ЗКД2. Здатність до навчання та саморозвитку.

СК16. Здатність використовувати для вирішення професійних завдань новітні технології у галузі автоматизації та комп’ютерно-інтегрованих технологій, зокрема, проектування багаторівневих систем керування, збору даних та їх архівування для формування бази даних параметрів процесу та їх візуалізації за допомогою засобів людино-машинного інтерфейсу.

СК19. Здатність вільно користуватись сучасними комп’ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп’ютерно-інтегровані середовища для вирішення задач автоматизації.

СКД1 Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

IV ПРОГРАМА ТА СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Денна повна форма навчання

Вид навчальних занять / контролю	Розподіл між учбовими тижнями (1 семестр)														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Повний курс															
Лекції	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Практ. заняття	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Сам. робота	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Консультації						К									К
Контр. роботи							КР1								КР2
Змістовні модулі	ЗМ1						ЗМ2								
Контроль по модулю	ПР1	ПР2		ПР3		ПР4		ПР5		ПР6		ПР7		ПР8	

Вид навчальних занять/контролю	Розподіл між учбовими тижнями (2 семестр)																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Лекції																		
Курсова робота	2	2	2	2	2	2	2	2	2	2	2	2	2					
Сам. робота	1	1	1	1	1	1	1	2	2	2	2	2	2					
Консультації																		
Контр. роботи																		
Змістовні модулі																		
Контроль по модулю																		

Денна (прискорена) форма навчання

Вид навчальних занять / контролю	Розподіл між учбовими тижнями												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Лекції	2	2	2	2	2	2	2	2	2	2	2	2	2
Практ. заняття	2	2	2	2	2	2	2	2	2	2	2	2	2
Сам. робота	3	3	3	3	3	3	3	3	3	3	3	3	3
Консультації				К					К				
Контр. роботи							КР1						КР2
Змістовні модулі	ЗМ1						ЗМ2						
Контроль по модулю	ПР1	ПР2		ПР3		ПР4		ПР5		ПР6		ПР7	

Заочна форма навчання

Вид навчальних занять / контролю	Розподіл між учбовими тижнями														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Повний курс															
Лекції	4	4							4						
Практич. заняття	4	4								4					
Курсова робота		2	2										2	2	
Сам. робота	12	12	13	13	13	13	13	13	13	13	13	13	13	13	13
Консультації			К		К						К				К
Контр. роботи						КР1									КР2
Змістовні модулі	ЗМ1						ЗМ2								
Контроль по модулю	ПР 1,2	ПР 3,4							ПР5, 6	ПР7, 8					

Вид навчальних занять / контролю	Розподіл між учбовими тижнями																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Прискорений курс																		
Лекції	4	4							4									
Практич. заняття		2								2								
Сам. робота	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5			
Консультації															К			
Контр. роботи								КР1							КР2			
Змістовні модулі	ЗМ1								ЗМ2									
Контроль по модулю	ПР 1,2	ПР 3,4							ПР 5,6									

Лекції

№ з/п	Найменування змістовних модулів і тем	Кількість годин (денна/ денна прискорена) / (заочна / заочна прискорена)					
		Разом	в т.ч.				
			Л	П	Лаб	СРС	Література
1	2	3	4	5	6	7	8
Змістовий модуль 1. Об'єктно-орієнтовна програмна інженерія							
1	Лекція 1. Вступ. Визначення і основні поняття програмної інженерії. Роль програмної інженерії в практиці та дослідженнях. Важливість програмної інженерії для розробки програмного продукту.	(12/7) / (12,5 / 4,5)	(2/2) / (0,5/0,5)	(2/2) / (0/0)		(8/3) / (12/4)	[1], [2]
2	Лекція 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.	(12/7) / (12,5 /5,5)	(2/2) / (0,5/0,5)	(2/2) / (0/0)		(8/3) / (12/5)	[1], [2], [4].
3	Лекція 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 - CPM. Планування і контроль проекту. Оцінювання вартості проекту. Методи керування ризиками у проекті.	(12/7) / (13,5 /5,5)	(2/2) / (0,5/0,5)	(2/2) / (0/0)		(8/3) / (13/5)	[2] [4]

1	2	3	4	5	6	7	8
4	Лекція 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.	(12/7) / (14,5 /6,5)	(2/2) / (0,5/0, 5)	(2/2) / (1/1)		(8/3) / (13/5)	[1], [2]
5	Лекція 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.	(12/7) / (15 /6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[1], [2]
6	Лекція 6. Інженерія вимог. Визначення вимоги та інженерії вимог. Стейкхолдери. Типи вимог. Трасування вимог. Зв’язок між вимогами. Функціональні і нефункціональні вимоги.	(12/7) / (14 /6)	(2/2) / (1/1)	(2/2) / (0/0)		(8/3) / (13/5)	[3], [6]
7	Лекція 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.	(12/7) / (15 /6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[5]
8	Лекція 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).	(12/7) / (15 /7)	(2/2) / (1/1)	(2/2) / (1/1)		(8/3) / (13/5)	[6]

1	2	3	4	5	6	7	8
Змістовий модуль 2 Моделювання та специфікація системи. Вимірювання та оцінка програмного забезпечення.							
9	Лекція 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.	(12/7) / (15/6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[3], [5], [6]
10	Лекція 10. Проєктування: розподіл відповілальностей. Взаємодія об'єктів. Діаграма послідовностей. Характеристики професійного дизайну. Розподіл відповідальностей. Проектування на основі відповідальності. Бізнес правила. Діаграма класів.	(12/7) / (15/6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[5], [6]
11	Лекція 11. Принципи об'єктно-орієнтованого проєктування. SOLID принципи: принцип єдиної відповідальності, принцип відкритості/закритості, принцип підстановки Барбари Лісков, принцип розділення інтерфейсу, принцип інверсії залежностей. Мета застосування принципів проектування. Приклади використання.	(12/7) / (15/6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[7]
12	Лекція 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.	(12/7) / (15/6)	(2/2) / (1/1)	(2/2) / (1/0)		(8/3) / (13/5)	[1], [2], [7]

1	2	3	4	5	6	7	8
13	Лекція 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:, додаткове обмеження def:.	(12/7) / (15/7)	(2/2) / (1/1)	(2/2) / (1/1)		(8/3) / (13/5)	[2], [4]
14	Лекція 14. Архітектурне проектування. Діаграми пакетів. Діаграми компонентів. Порівняльний аналіз пакетів і компонентів. Порівняльний аналіз компонентів, класів та інтерфейсів.	(12/0) / (14,5/5,5)	(2/0) / (0,5/0 0,5)	(2/0) / (1/0)		(8/0) / (13/5)	[7], [8]
15	Лекція 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.	(12/0) / (15,5/6,5)	(2/0) / (0,5/0 0,5)	(2/0) / (2/1)		(8/0) / (13/5)	[8]
Курсова робота				(26 / 0) / (8 / 0)		(19 / 0) / (37 / 0)	
Разом годин		(225/90) / (225 / 90)	(30/ 26) / (12 / 12)	(56/ 26) / (20 / 4)		(139/ 38) / (193 / 74)	

Теми практичних занять

№ з/п	Тема	Назва практичної роботи
Змістовий модуль 1 Вступ. Об'єктно-орієнтовна програмна інженерія.		
1	Тема 4	Вивчення архітектури, візуальних інтерфейсів та інструментальних засобів CASE-системи Visual Paradigm Community Edition
2	Тема 7	Аналіз вимог та розробка UML – діаграм концептуального рівня проектування програмної системи
3	Тема 9.	Розробка UML – діаграм логічного рівня проектування програмної системи: моделювання статичних аспектів
4	Тема 10	Розробка UML – діаграм логічного рівня проектування програмної системи: моделювання динамічних аспектів
Змістовий модуль 2 Моделювання та специфікація системи. Вимірювання та оцінка програмного забезпечення.		
5	Тема 12	Розробка UML – діаграм фізичного рівня проектування програмної системи
6	Тема 14	Розробка діаграми класів та специфікацій інтерфейсів в Visual Paradigm Community Edition.
7	Тема 15	Архітектура та дизайн системи.
8	Тема 16	Використання патернів проектування.

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 «Технологія програмування складних систем», 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 «Технологія програмування складних систем», 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.

5. МЕТОДИ НАВЧАННЯ

За джерелами знань використовуються такі методи навчання: словесні – розповідь, пояснення, лекція, інструктаж; наочні – демонстрація,

ілюстрація практичні роботи в комп'ютерному класі з пошуком інформації в Інтернет.

За характером логіки пізнання використовуються такі методи: аналітичний, синтетичний, аналітико-синтетичний, індуктивний, дедуктивний.

За рівнем самостійної розумової діяльності використовуються методи: проблемний, частково-пошуковий, дослідницький.

При викладанні дисципліни передбачається використовування мультимедійних засобів, плакатів і натурних зразків. Розглядаються характерні приклади реальних процесів.

Для покращення засвоєння матеріалу студентами їм рекомендується поглиблене самостійне вивчення окремих питань. Успіх вивчення дисципліни залежить від систематичної самостійної роботи студента з матеріалами лекцій і рекомендованою літературою.

6. МЕТОДИ КОНТРОЛЮ

Передбачається використовування модульно-рейтингової системи оцінювання знань. Основною формою контролю знань студентів в кредитно модульній системі є складання студентами всіх запланованих модулів. Формою контролю є накопичувальна система. Складання модуля передбачає виконання студентом комплексу заходів, запланованих кафедрою і передбачених семестровим графіком навчального процесу та контролю знань студентів, затверджених деканом факультету.

Підсумкова оцінка за кожний модуль виставляється за 100-бальною шкалою. Переведення набраних студентом балів за 100-бальною шкалою в оцінки за національною (5-бальною) шкалою та шкалою ECTS здійснюється в відповідності до таблиці:

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

Контроль знань студентів передбачає проведення вхідного, поточного і підсумкового контролю.

Вхідний контроль включає контроль залишкових знань з окремих навчальних дисциплін, які передують вивченню даної дисципліни.

Поточний контроль знань студентів включає наступні види:

- вибірковий усний опит перед початком кожної практичної роботи по темі заняття із виставленням оцінок (балів);
- захист кожної практичної роботи з виставленням оцінок (балів);
- захист індивідуальних завдань з самостійної роботи;
- письмові контрольні роботи з окремих модулів дисципліни.

Підсумковий контроль знань включає наступні види:

- модульний контроль за результатами захисту практичних робіт, програмованого контролю знань і контрольних робіт;
- екзамен (письмовий) після завершення вивчення дисципліни.

7. 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

8. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ

Методичні вказівки

1. Технологія програмування складних систем. Конспект лекцій (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»).
2. Методичні вказівки до виконання практичних робіт з дисципліни ” Технологія програмування складних систем ” (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»).

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, Бородкин, 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. Кучерук Г. І., Чумаченко В. М., Яковлев В. С. Візуальне моделювання з UML. – Київ: Наукова думка, 2008. – 256 с.

10. Сахацький О. Л. Об’єктно-орієнтоване моделювання програмних систем. – Львів: Новий Світ-2000, 2007. – 318 с.

Література додаткова

1. Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. 2nd ed. – Boston: Addison-Wesley Professional, 2005. – 496 p.

2. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. – Boston: Addison-Wesley Professional, 2003. – 208 p.

3. Larman C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3rd ed. – Upper Saddle River: Prentice Hall, 2004. – 736 p.

4. Ambler S. W., Lines M. Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. – New York: John Wiley & Sons, 2002. – 400 p.

Робоча програма складена
к.т.н, асист. кафедри АВП,
Картамишев Дмитро Олександрович