

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

«ЗАТВЕРДЖУЮ»
Голова вченої ради
ФІПС
Чепинога А.В. /
Протокол № 1
«30» серпня 2021 р.



РОБОЧА ПРОГРАМА

навчальної дисципліни

«Моделювання програмних систем та їх реалізація»

Шифр за ОНП – ОД₇

Рівень вищої освіти	Освітньо-науковий (доктор філософії)
Спеціальність -	121 - Інженерія програмного забезпечення
Освітня програма -	«Інженерія програмного забезпечення»

2021-2022 навчальний рік

Робоча програма навчальної дисципліни *«Моделивання програмних систем та їх реалізація»* підготовки здобувачів освітньо-наукового рівня «Доктор філософії» за спеціальністю 121 – Інженерія програмного забезпечення, освітня програма «Інженерія програмного забезпечення» - 11 сторінок.

Розробник:

Салапатов В.І., доцент кафедри програмного забезпечення автоматизованих систем, к.т.н., доцент

Робоча програма навчальної дисципліни затверджена на засіданні кафедри програмне забезпечення автоматизованих систем

Протокол № 2 від « 26 » серпня 2021 р.

Завідувач кафедри  /Первунінський С.М./

Обговорено та рекомендовано до затвердження методичною комісією факультету інформаційних технологій і систем

«30» серпня 2021 р., протокол № 1

Голова методичної комісії факультету інформаційних технологій і систем



підпис

/Карпетян А.Р./

ПІБ

1 ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Прізвище, ім'я, по батькові	Салапатов Володимир Іванович
Науковий ступінь	кандидат технічних наук
Вчене звання	доцент
Посада	доцент кафедри програмного забезпечення автоматизованих систем
Місце роботи	Черкаський державний технологічний університет
Адреса кафедри	бульв. Шевченка, 460, м. Черкаси, ЧДТУ, 1 корпус, каб. 605
Контактний телефон	+38(093)-419-82-76
Профайл викладача	https://pzas.chdtu.edu.ua/staff/salapatov-volodymyr-ivanovych/
e-mail:	V_salapatov@ukr.et
Профайл дисципліни	
Розклад консультацій	https://pzas.chdtu.edu.ua/debts/

2. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Галузь знань, спеціальність, освітня програма, освітній рівень	Загальні характеристики		Навчальне навантаження з дисципліни	
			денна форма	заочна форма
<u>Галузь знань</u> 12 «Інформаційні технології»	Обов'язкова		Курс підготовки:	
			1	
<u>Спеціальність</u> 121 – Інженерія програмного забезпечення	Загальна кількість кредитів ЄКТС	4	Семестр підготовки:	
	Загальна кількість годин	120	1	
<u>Освітня програма</u> «Інженерія програмного забезпечення»	Кількість аудиторних годин	48	Лекції	
			32	
	Кількість годин самостійної роботи	72	Практичні, семінарські	
			16	
<u>Освітньо-науковий рівень</u> Доктор філософії	Мова навчання - українська		Лабораторні	
			-	
			Самостійна робота	
			72	
			Форма підсумкового контролю	
			іспит	

3 МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета викладання дисципліни	Здобуття аспірантами знань щодо методів моделювання та розробки безпомилкових і надійних програм. Аспіранти мають ознайомитись і усвідомити сучасні методи побудови моделей програм. Вони також мають засвоїти сучасні засоби опису моделей і створення за їхньою допомогою власно моделей. Розглядаються, також, методи перетворення моделей програм у власно програми та методи синхронізації паралельних програм.
Завдання вивчення дисципліни	Підготовка аспірантів до опанування сучасних технологій моделювання програмних систем, вивчення та опис інформаційних зв'язків між окремими частинами програмних систем. Набуття компетенцій, знань, умінь та навичок для подальших досліджень у галузі побудови надійних програм, зокрема паралельних програм.

4 РЕЗУЛЬТАТИ НАВЧАННЯ

№ з/п	Результати навчання
1	РН1. Мати передові концептуальні та методологічні знання з інженерії програмного забезпечення та дотичних до неї міждисциплінарних напрямів.
2	РН3. Пропонувати нові ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.
3	Засвоїти існуючі методи моделювання програмних систем, починаючи з концептуальних моделей і переходу до адекватних математичних моделей. Усвідомити сучасні методи створення моделей програм на основі недетермінованих скінчених автоматів.
4	Ознайомитись і засвоїти засоби синхронізації для паралельних програм та їх моделювання. Вміти описувати і створювати моделі програм. Усвідомити особливості опису потоків у паралельних програмах.
5	Вміти та застосовувати на практиці сучасні технології моделювання програмних систем. Вміти коректно визначати дані та їх типи і забезпечити сумісність даних між собою..
6	Засвоїти засоби перетворення моделей програм у власно програми на будь-якій цільовій процедурній мові програмування. Вміти провести остаточну перевірку і валідацію програм.

5 ПРЕРЕКВІЗИТИ

Дисципліна «Моделювання програмних систем та їх реалізація» базується на знаннях і навичках, отриманих студентами при вивченні шкільних дисциплін «Вступ до програмної інженерії», «Архітектура мікропроцесорів», «Основи програмування та алгоритмічні мови».

6 ПОСТРЕКВІЗИТИ

На базі дисципліни «Моделювання програмних систем та їх реалізація» закладаються основи для подальшого дослідження розробки програмних систем, зокрема паралельних систем.

7 ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Передові концептуальні та методологічні знання в області розробки програм. Методи побудови моделей програм
Зміст дисципліни. Огляд існуючих технологій розробки програм. Проблема коректності програм. Визначення моделі. Побудова блок-схем програм. Причини виникнення помилок. Методологія верифікації програм. Побудова моделей програм.
Тема 2. Методи і моделі розроблення, впровадження, супроводу програмного забезпечення на всіх етапах життєвого циклу. UML-технологія
Подання UML. Статичне подання моделі. Подання взаємодії. Діаграма послідовності, діаграма кооперації, діаграма діяльності. Пакет. Взаємодія між пакетами. Залежності експорту та імпорту. Метамоделі.
Тема 3. Ефективні методи і моделі розроблення, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу. Технологія Model Checking
Технологія опису і побудови програм за технологією Model Checking . Порядок розробки програм згідно технології Model Checking . CTL-логіка як засіб визначення гілок програми. Подання моделі програми за допомогою структури Крипке. Оператори темпоральної логіки. Приклади застосування темпоральної логіки. Недетермінований скінченний автомат як результат опису моделі програми. Дії у станах моделі програми. Розгалуження у станах моделі. Порядок опису та створення автоматної моделі програми. Верифікація моделі програми за допомогою програм-верифікаторів.
Тема 4. Переваги та недоліки технології Model Checking. Опис моделі за допомогою предикатів
Предикат як засіб для опису алгоритму автоматних моделей. Опис умови застосування предикату. Опис дій згідно предикату. Специфікації та обмеження при описі моделі. Програма створення автоматної моделі. Верифікатори моделей. Поняття монітору для доступу спільних ресурсів. Опис моніторів. Опис паралельних гілок програми.
Тема 5. Подання автоматної моделі програми. Правила обробки автоматної моделі програми
Застосування бази даних для подання автоматної моделі програми. Типова структура бази даних для подання моделі програми. Порядок обробки дерева автоматної моделі по її гілках. Умови завершення обробки гілок. Умова завершення обробки моделі.
Тема 6. Засоби обробки автоматної моделі
Лінійне просування при обробці моделі. Застосування стеку при лінійному просуванні при обробці моделі програми. Формування внутрішнього подання моделі програми у вигляді бази даних. Логіка обробки моделі. Стековий алгоритм.
Тема 7. Перехід від моделі до програми
Реалізація дій у кожному стані моделі. Визначення кінця стану, реалізація переходів

у наступні стани. Перевірка сумісності типів даних. Приведення типів даних до припустимих для конкретної процедурної мови програмування. Остаточна валідація програми.

8 ТЕМАТИЧНИЙ ПЛАН НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ теми	Назва модулів і тем	Форми організації навчання, кількість годин				Література, інформаційні ресурси
		Денна форма				
		Лекції	Практичні роботи	Лабораторні роботи	Самостійна робота	
1	Зміст дисципліни. Огляд існуючих технологій розробки програм. Проблема коректності програм.	2			4	1, 2, 3, 4
2	Побудова блок-схем програм. Причини виникнення помилок. Методологія верифікації програм. Побудова моделей програм.	2				1, 2, 3, 4
3	Подання UML. Статичне подання моделі. Подання взаємодії.	2	2		8	1, 2, 3, 4
4	Подання UML. Статичне подання моделі. Подання взаємодії.	2	4			1, 2, 3, 4
5	Пакет. Взаємодія між пакетами. Залежності експорту та імпорту. Мета модель.	2	2		8	1, 2, 3, 4
6	Технологія опису і побудови програм за технологією Model Checking . Порядок розробки програм згідно технології Model Checking .	2	2		8	1, 2, 3, 4
7	CTL-логіка як засіб визначення гілок програми. Подання моделі програми за допомогою структури Крипке. Оператори темпоральної логіки. Приклади застосування темпоральної логіки.	2	2			1, 3, 4, 5
8	Недетермінований скінченний автомат як результат опису моделі програми. Дії у станах моделі програми. Розгалуження у станах моделі програми. Верифікація моделі програми за допомогою програм-верифікаторів.	2	2		4	1, 2, 3, 4
9	Предикат як засіб для опису алгоритму автоматних моделей. Опис умови застосування предикату. Опис дій згідно предикату. Специфікації та обмеження при описі моделі. Програма створення автоматної моделі. Верифікатори моделей.	2	2		8	1, 2, 3, 4

10	Поняття монітору для доступу спільних ресурсів. Опис моніторів. Опис паралельних гілок програми.	2	2		8	1, 2, 3, 4
11.	Застосування бази даних для подання автоматної моделі програми. Типова структура бази даних для подання моделі програми.	2	2		8	1, 2
12.	Порядок обробки дерева автоматної моделі по її гілках. Умови завершення обробки гілок. Умова завершення обробки моделі.	2	2			1, 2
13.	Лінійне просування при обробці моделі. Застосування стеку при лінійному просуванні при обробці моделі програми.	2	2		8	1, 2
14.	Формування внутрішнього подання моделі програми у вигляді бази даних. Логіка обробки моделі. Стековий алгоритм.	2				1, 2, 3, 4
15.	Реалізація дій у кожному стані моделі. Визначення кінця стану, реалізація переходів у наступні стани.	2	2			1, 2, 3, 4
16.	Перевірка сумісності типів даних. Приведення типів даних до припустимих для конкретної процедурної мови програмування. Остаточна валідація програми.	2	2		8	1, 2, 3, 4
Разом		32	16		72	

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

9 ПРАКТИЧНІ РОБОТИ

№ з/п	Назва теми	Кількість годин
1.	Створення моделі програми перетворення константи типу INTEGER із зовнішнього подання у внутрішній формат. Програмне моделювання.	4
2.	Створення моделі програми перетворення константи типу FLOAT POINT із зовнішнього подання у внутрішній формат. Програмне моделювання.	4
3.	Створення моделі програми перетворення константи типу FLOAT POINT із внутрішнього подання у зовнішній формат для виведення. Програмне моделювання.	4
4.	Створення моделі програми зі створення монітора для доступу до спільних ресурсів. Програмне моделювання.	4
5.	Створення моделі програми зі створення програми з паралельними потоками гілок програм. Програмне моделювання.	
Разом		16

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Методичні вказівки з курсу «Моделювання програмних систем та їх реалізація» для бакалаврів спеціальності: 121 – Інженерія програмного забезпечення. Укл.:Салапатов В.І. – Черкаси. – ЧДТУ. – 2021.- 55 с.

10 САМОСТІЙНА РОБОТА

10.1 Рекомендації до самостійної роботи здобувачів вищої освіти денної форми навчання

№ з/п	Назва теми	Кількість годин
1	Структурне програмування, функціональне програмування, об'єктно-орієнтоване програмування.	12
2	UML-технологія.	12
3	CTL- та LTL-логіка. Приклади застосування.	12
4	Поняття предикату, застосування предикатів для опису програм.	12
5	Подання автоматної моделі.	12
6	Обробка автоматної моделі.	12
Разом		72

МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Методичні вказівки до виконання практичних і лабораторних робіт з курсу “ Моделювання програмних систем та їх реалізація ” для бакалаврів спеціальності: 121 – Інженерія програмного забезпечення. Укл.:Салапатов В.І. – Черкаси. – ЧДТУ. – 2022.- 55 с.

11 СИСТЕМА ОЦІНЮВАННЯ НАВЧАЛЬНИХ ДОСЯГНЕНЬ

11.1 МЕТОДИ КОНТРОЛЮ

Поточний контроль знань, вмінь та навичок аспірантів передбачає застосування таких видів:

- контрольні завдання;
- практичні завдання;
- обговорення проблеми, дискусія;
- аналіз конкретних ситуацій (поданих у вигляді усного, текстового або графічного матеріалу);
- командна робота;
- презентації результатів роботи;
- інші.

Об'єктами поточного контролю знань аспірантів є:

- систематичність, активність, своєчасність та результативність роботи над вивченням програмного матеріалу дисципліни, у т.ч. виконання домашніх завдань та розв'язання задач;
- виконання завдань для самостійного опрацювання.

Поточний контроль знань здійснюється шляхом двох модулів у формі контрольної модульної роботи. Оцінювання знань аспірантів здійснюється за 100-бальною шкалою:

– максимальна кількість балів при оцінюванні знань аспірантів з дисципліни, яка завершується екзаменом, становить за поточну успішність 60 балів, на іспиті– 40 балів;

– при оформленні документів за залікову сесію використовується таблиця відповідності оцінювання знань аспірантів за різними системами:

СУМА БАЛІВ	ОЦІНКА ECTS	Оцінка за національною шкалою	
		екзамен	іспит
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D		
60 – 63	E	задовільно	не зараховано
35 – 59	FX	незадовільно	
1 – 34	F		

Об'єктом *підсумкового контролю знань* аспірантів є *іспит* у формі тесту. Для відповіді на тестові питання аспірант повинен розв'язати контрольні завдання та відповісти на теоретичні питання.

На *іспит* виносяться базові питання та завдання, що потребують творчого підходу та вміння опрацювати лекційний матеріал.

Результати іспиту аспірантів денної форми навчання оцінюються в діапазоні від 0 до 40 балів.

11.2 ПИТАННЯ ДО ІСПИТУ

1. Визначення моделі. Призначення моделі у програмуванні.
2. Поняття верифікації програм, методи верифікації.
3. Подання UML. Статичне подання моделі. Подання взаємодії.
4. Діаграми послідовності, діаграма кооперації, діаграма діяльності. Їх призначення.
5. Пакети, взаємодія пакетів.
6. Технологія **Model Checking**, недетермінований скінченний автомат.
7. Порядок розробки програм за технологією **Model Checking**.
8. Темпоральна логіка. Оператори темпоральної логіки.
9. Структура Крипке як форма подання автоматної моделі.
10. Недетермінований скінченний автомат як результат опису моделі програми.
11. Дії у станах моделі програми. Функція виходу і функція переходу.
12. Розгалуження у станах моделі.
13. Порядок опису та створення автоматної моделі програми.
14. Предикат як засіб для опису алгоритму автоматних моделей.
15. Опис умови застосування предикату. Опис дій згідно предикату.
16. Специфікації та обмеження при описі моделі. Програма створення автоматної моделі. Верифікатори моделей.
17. Поняття монітору для доступу спільних ресурсів.

18. Опис моніторів.
19. Опис паралельних гілок програми.
20. Подання автоматної моделі програми. Типова структура бази даних для подання моделі програми.
21. Порядок обробки дерева автоматної моделі по її гілках.
22. Умови завершення обробки гілок. Умова завершення обробки моделі.
23. Лінійне просування при обробці моделі.
24. Застосування стеку при лінійному просуванні при обробці моделі програми.
25. Формування внутрішнього подання моделі програми у вигляді бази даних.
26. Логіка обробки моделі.
27. Стековий алгоритм. Сутність алгоритму.
28. Реалізація дій у кожному стані моделі.
29. Визначення кінця стану, реалізація переходів у наступні стани.
30. Перевірка сумісності типів даних.
31. Приведення типів даних до припустимих для конкретної процедурної мови програмування.
32. Остаточна валідація програми.

11.3 КРИТЕРІЇ ОЦІНЮВАННЯ

Для аспірантів денної форми навчання	
Вид навчальної роботи	Кількість балів максимум
<u>Постійна частина</u>	
Виконання практичної роботи №1	12
Виконання практичної роботи №2	12
Виконання практичної роботи №3	12
Виконання практичної роботи №4	12
Виконання практичної роботи №5	12
<i>Всього</i>	60
<u>Додаткова частина</u>	
Підготовка та захист реферату за індивідуальною темою	20
Участь в науковій конференції за темою дисципліни	30
Оформлення презентації за індивідуальною темою	10
<u>Штрафна частина</u>	
Пропуск одного заняття без поважної причини	-10
Здача звіту з практичних занять пізніше узгодженого терміну	-5
ІСПИТ	40
ПІДСУМКОВА СЕМЕСТРОВА ОЦІНКА	100

12 РЕКОМЕНДОВАНА ЛІТЕРАТУРА

12.1 Основна

1. Н. Вирт. Алгоритмы + структуры данных = программы. — М., "Мир", 1985. — 360 с.
2. Джон Хопкрофт, Раджив Модвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений, 2-е изд.; Пер. С англ. — М. Издательский дом «Вильямс», 2002. — 528 с.
3. Карпов Ю.Г. MODEL CHECKING. Верификация параллельных и распределенных программных систем. /Ю.Г. Карпов — СПб.; БХВ-Петербург, 2010.- 560 с..
4. Ч. Хоар. Взаимодействующие последовательные процессы. Перевод с англ. — М.: Мир. 1989.- 264 с.
5. А.Є. Конверський. Логіка (традиційна та сучасна): Підручник для студентів вищих навчальних закладів. — К. центр учбової літератури. — 2008. — 536 с.
6. Джеймс Рамбо, Айвар Якобсон, Грэд Буч. UML: специальный справочник. — СПб.: Питер, 2002 — 656 с.

12.2 Додаткова

1. Ошибка в ПО Airbus A350 вынуждает перезагружать системы самолетов каждые 149 часов [Электронный ресурс] : [Веб-сайт]. — Стаття. — режим доступу: <https://internetua.com/oshibka-v-po-airbus-a350-vynujdaet-perezagrujat-sistemy-samoletov-kajdye-149-csasov>.
2. V. Salapatov. Technology for modelling software systems based non-deterministic finite automats. SECURITY \$ FUTURE INTERNETIONAL SCIENTIFIC JOURNAL. Sofia. Bulgaria. 2019. P. 113-114. Abstracts of reports.