

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ВАЛІДАЦІЯ ТА ВЕРИФІКАЦІЯ
ПРОГРАМНИХ СИСТЕМ

для студентів

галузь знань **12 «Інформаційні технології»**
спеціальність **122 «Комп'ютерні науки»**
освітній рівень **магістр**
освітня програма **«Інформатика»**
вид дисципліни **обов'язкова**

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	2
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: д.ф.-м.н., проф. **Анатолій ДОРОШЕНКО**

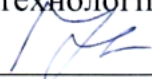
Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.

на 20__/20__ н.р. _____ (_____) «__» 20__ р.

КИЇВ – 2021

Розробник: Анатолій ДОРОШЕНКО, д.ф.-м.н., професор, професор кафедри теорії та технології програмування

ЗАТВЕРДЖЕНО

Завідувач кафедри
теорії та технології програмування
 Микола НІКІТЧЕНКО

Протокол № 10 від «27» квітня 2021 року

Схвалено гарантом освітньо-наукової програми «Інформатика»

 Степан ШКІЛЬНЯК

«6» травня 2021 року

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «6» травня 2021 року № 10

Голова науково-методичної комісії  Людмила ОМЕЛЬЧУК

«6» травня 2021 року

1. Мета дисципліни – засвоєння техніки перевірки правильності програм шляхом побудови моделей програм з подальшим доведенням властивості побудованих моделей, що характеризують її правильність, а також техніки тестування програмних систем в основних його видах.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* базові поняття математичної логіки, теорії алгоритмів, теорії програмування; мати уяву про складність алгоритмів.

2. *Вміти:* будувати логічні специфікації програм з урахуванням встановлених вимог, виконувати верифікаційні процедури з перевірки правильності та валідності програмного забезпечення, а також проводити основні види тестування програм.

3. *Володіти засобами:* розробки тестів у середовищі Visual Studio.

3. Анотація навчальної дисципліни

Навчальна дисципліна "Валідація та верифікація програмних систем" є складовою освітньо-наукової програми підготовки спеціалістів освітньо-кваліфікаційного рівня "магістр" галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-наукової програми – „Інформатика”.

Цей курс потрібен для подальшого вивчення дисциплін програмістського напрямку магістерської програми з інформатики, а також низки спецкурсів відповідного напрямку.

Дана дисципліна є обов’язковою навчальною дисципліною за *програмою “Інформатика”*. Викладається у 2 семестрі 1 курсу магістерської програми в обсязі – 120 год. (4 кредити ECTS), зокрема: лекції – 26 год., лабораторні – 12 год., консультації – 2 год., самостійна робота – 80 год. У курсі передбачено 2 частини та 2 контрольні роботи.

Завершується дисципліна – іспитом у 2 семестрі.

В результаті вивчення навчальної дисципліни студент повинен:

знати: місце і значення верифікації і валідації програмних систем в життєвому циклі програмних систем; основні етапи розробки програмних систем, методи інженерії вимог до програмного забезпечення; класифікацію відмов і помилок у програмних системах, основні поняття та методи тестування програмних систем, методи доведення правильності програм, логічні засоби специфікації програм та методи перевірки на моделях, процеси, артефакти та стандарти верифікації програмного забезпечення.

вміти: будувати логічні специфікації програм з урахуванням встановлених вимог; виконувати верифікаційні процедури з перевірки правильності та валідності програмного забезпечення, а також проводити основні види тестування програм.

Дисципліна "Валідація та верифікація програмних систем" є необхідною для засвоєння курсів магістерської програми: "Програмна інженерія", "Інформаційне моделювання систем", "Інструментальні засоби та технології програмування", "Програмні логіки та їх застосування".

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок, формування компетентностей на рівні новітніх досягнень у інформатиці, відповідно до кваліфікації «Магістр з комп’ютерних наук».

Зокрема, розвивати:

– СК2. Здатність ідентифікувати моделі складних систем і процесів, розробляти та застосовувати методи і засоби моделювання та прогнозування систем і процесів в умовах невизначеності.

– СК16. Здатність і готовність до проектування інформаційної системи визначеного прикладного застосування шляхом аналізу та синтезу складу та структури системи або окремих їх складових, розробка функціональних і нефункціональних вимог до системи, що проектується.

5. Результати навчання за дисципліною:

Результат навчання (РН) (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність*)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати необхідні методи верифікації і валідації програмних систем в життєвому циклі програмної системи	Лекція, лабораторне заняття	Контрольна робота, іспит	14%
РН1.2	Знати логічні засоби специфікації програм та методи перевірки на моделях	Лекція, лабораторне заняття	Контрольна робота, захист лабораторних робіт, іспит	10%
РН1.3	Знати процеси, артефакти та стандарти верифікації програмного забезпечення	Лекція, лабораторне заняття	Контрольна робота, захист лабораторних робіт, іспит	16%
РН2.1	Вміти застосовувати методи інженерії вимог до програмного забезпечення	Лекція, лабораторне заняття	Захист лабораторних робіт, іспит	20%
РН2.2	Вміти правильно класифікувати відмови і помилки у програмних системах	Лекція, лабораторне заняття	Захист лабораторних робіт іспит	20%
РН2.3	Вміти застосовувати методи доведення правильності програм	Лекція, лабораторне заняття, самостійна робота	Захист лабораторних робіт іспит	10%
РН2.4	Вміти планувати та використовувати методи тестування програмних систем	Лабораторна робота, самостійна робота	Захист лабораторних робіт іспит	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 2.2	РН 2.3	РН 2.4
	Програмні результати навчання (з опису освітньої програми)						
ПРН1. Ідентифікувати проблемні ситуації, виконувати їх дослідження на основі системного підходу, здійснювати обґрунтований вибір методів та моделей для формування ефективних управлінських рішень, застосовувати моделі і методи прийняття рішень у прогнозуванні розвитку підприємства та в предметній області комп'ютерних наук.	+				+		+
ПРН6. Використовувати розподілені високопродуктивні обчислювальні технології для забезпечення ефективного вибору та використання консолідованих ресурсів і послуг				+			+
ПРН15. Володіти методами розробки та впровадження заходів, спрямованих на підвищення ефективності інформаційних систем.	+	+	+			+	

7. Схема формування оцінки

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1 (тест): РН 1.1., РН 1.2 – 9 балів/4,5 балів.
2. Контрольна робота 2 (тест): РН1.3 – 9 балів/4,5 балів.
3. Захист лабораторної роботи 1 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.
4. Захист лабораторної роботи 2 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.
5. Захист лабораторної роботи 3 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.
6. Захист лабораторної роботи 4 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.
7. Захист лабораторної роботи 5 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.
8. Захист лабораторної роботи 6 (проект): РН 1.2, РН 1.3, РН 2.1, РН 2.2, РН 2,3, РН 2.4 – 7 балів/3,5 балів.

- підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН2.2, РН2.3, РН2.4;
- форма проведення: письмова робота.
- види завдань:

Структура екзаменаційної роботи та критерії оцінювання:

1. Теоретичне запитання з етапів життєвого циклу (РН1.1).
2. Теоретичне запитання з логічних засобів специфікації програм (РН1.2).
3. Теоретичне запитання з методів перевірки на моделях (РН1.2).
4. Теоретичне запитання про процеси, артефакти та стандарти верифікації програмного забезпечення (РН1.3).
5. Письмове завдання з методів інженерії вимог (РН2.1).
6. Письмове завдання з класифікації відмов і помилок (РН2.2).
7. Письмове завдання з методів доведення правильності програм (РН2.3).
8. Письмове завдання з методів планування та проведення тестування програмних систем (РН2.4).

Критерії оцінювання екзаменаційної роботи

Завдання	Види завдань	Максимальний відсоток за завдання (бал)	Всього відсотків (балів)
Завдання 1, 2, 3, 4,	Письмове запитання з курсу	по 10% (4 балів)	40% (16 балів)
Завдання 5, 6,7,8	Письмове завдання	По 15% (6 балів)	60% (24 балів)
Всього			100% (40 балів)

Запитання для підготовки до іспиту

1. Програмний проект та його тріада
2. Проблема розробки програмних проектів.
3. Основні категорії розробки ПЗ.
4. Методології розробки програмного забезпечення MSF
5. Методології розробки програмного забезпечення RUP
6. Методології розробки програмного забезпечення XP
7. Поняття життєвого циклу ПЗ. Послідовні та ітераційні методології розробки
8. Методи збирання та аналізу вимог до ПЗ
9. Розробка вимог до ПЗ
10. Специфікація вимог до ПЗ
11. Сценарний підхід до розробки вимог
12. Завдання верифікації та валідації ПЗ
13. Класифікація відмов та помилок у ПЗ
14. Причини появи помилок на етапах ЖЦ
15. Основні поняття та методи тестування програм
16. Процеси тестування програмних систем
17. Засоби автоматизації процесу тестування в середовищі MS Visual Studio
18. Засоби формальної специфікації програм
19. Моделювання станів програми за допомогою моделей Кріпке
20. Методи створення моделі першого порядку з тексту програми
21. Методи доведення правильності програм
22. Логічні засоби специфікацій програм
23. Специфікація поведінки програм засобами темпоральної логіки
24. Перевірка формул на моделі
25. Метод двійкових діаграм рішень для подання булевих функцій
26. Подання моделей Кріпке засобами двійкових діаграм
27. Символьна верифікація моделей
28. Процеси верифікації ПЗ
29. Верифікація різних артефактів життєвого циклу ПЗ
30. Інспекційні методи верифікації ПЗ
31. Методи статичного аналізу ПЗ
32. Динамічні методи верифікації ПЗ
33. Формальні методи верифікації
34. Синтетичні методи верифікації ПЗ

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів.

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 30 балів. Студент допускається до іспиту за умови виконання 70% передбачених планом лабораторних робіт.

7.2 Організація оцінювання

Терміни проведення форм оцінювання:

1. Виконання лабораторних робіт 1-3: до 8 тижня семестру.
2. Виконання лабораторних робіт 4-6 (тест): до 18 тижня семестру.
3. Контрольна робота 1: до 6 тижня семестру.
4. Контрольна робота 2: до 19 тижня семестру.

Студент має право на одне перескладання контрольної роботи з можливістю отримання не більше 80% попередньо визначених балів за роботу. Термін перескладання визначається викладачем.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою 10% балів за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура навчальної дисципліни

Тематичний план лекцій і лабораторних занять

№ п/п	Назва теми	Кількість годин		
		Лекції	Лабораторні	Самостійна робота
Частина 1. Життєвий цикл розробки програмного забезпечення (ПЗ)				
1	Тема 1. Проблема розробки програмного програмного забезпечення. Програмний проект та його тріада. Проблема розробка програмних проектів. Основні категорії розробки ПЗ. <i>Самостійна робота:</i> Основні питання управління програмними проектами.	2		6
2	Тема 2. Життєвий цикл ПЗ та методології розробки його життєвого циклу. Поняття життєвого циклу ПЗ. Послідовні та ітераційні методології розробки ПЗ. <i>Самостійна робота:</i> Вивчення особливостей та порівняльна характеристика різних методологій розробки програм (MSF,RUP,XP). Виконання лабораторної роботи	2	2	4
3	Тема 3. Інженерія вимог до ПЗ . Структура специфікації вимог згідно міжнародних стандартів щодо ПЗ. <i>Самостійна робота:</i> Вивчення різних підходів до моделювання вимог (моделі станів, зміни станів та поведінки). Виконання лабораторної роботи	2		6
	<i>Контрольна робота 1</i>		1	
Всього за частиною 1		6	3	16
Частина 2. Методи перевірки правильності та тестування програм				
4	Тема 4. Поняття верифікації і валідації програмного забезпечення. Місце верифікації серед процесів розробки програмного забезпечення. Завдання верифікації та валідації ПЗ. <i>Самостійна робота:</i> Вивчення проблеми забезпечення якості ПЗ та основних положень міжнародних стандартів щодо якості програмного забезпечення.	2		6
5	Тема 5. Класифікація відмов та помилок у ПЗ. Класифікація помилок ПЗ: помилки, дефекти та відмови. <i>Самостійна робота:</i> Категорії дефектів розробки програм. Причини появи помилок на етапах ЖЦ. Виконання лабораторної роботи.	2	1	6
6	Тема 6. Основні поняття та методи тестування програм. <i>Самостійна робота:</i> Вивчення статичних та динамічних методів тестування програм. Виконання лабораторної роботи.	2	1	6
7	Тема 7. Формальне моделювання програмних систем. Засоби специфікації програм. Формальне моделювання програмних систем <i>Самостійна робота:</i> Методи створення моделі першого порядку з тексту програми. Виконання лабораторної роботи.	2		6
8	Тема 8. Методи доведення правильності програм. Метод індуктивних тверджень Флойда-Наура та метод структурної індукції Хоара. <i>Самостійна робота:</i> Метод Дейкстри формального доведення правильності програм. Виконання лабораторної роботи.	2	2	7

9	Тема 9. Логічні засоби специфікацій програм . Темпоральні логіки, логіка дерев обчислень STL*. <i>Самостійна робота:</i> Повнота операторів логіка дерев обчислень STL*. Виконання лабораторної роботи.	2	2	6
10	Тема 10. Перевірка формул на моделі. Верифікація моделей для логіки дерев обчислень STL. <i>Самостійна робота:</i> Алгоритми верифікації моделей для логіка дерев обчислень STL. Виконання лабораторної роботи.	2		6
11	Тема 11. Двійкові діаграми рішень. Ефективне подання булевих функцій та метод двійкових діаграм рішень. <i>Самостійна робота:</i> Подання моделей Крипке засобами двійкових діаграм. Виконання лабораторної роботи	2	1	7
12	Тема 12. Процеси верифікації ПЗ. Процеси, артефакти та стандарти верифікації ПЗ. <i>Самостійна робота:</i> Завдання верифікації в ЖЦ ПЗ. Верифікація і валідація ПЗ. Виконання лабораторної роботи.	2	1	7
13	Тема 13. Інспекційні методи верифікації ПЗ. Класифікація методів верифікації. Експертизи ПЗ. <i>Самостійна робота:</i> Практика застосування експертиз ПЗ. Виконання лабораторної роботи.	2		7
	<i>Контрольна робота 2</i>		1	
Всього за частиною 2		20	9	64
ВСЬОГО		26	12	80
Консультація		2		

Загальний обсяг 120 год., в тому числі:

Лекцій – **26 год.**

Лабораторні заняття - **12 год.**

Консультації – **2 год.**

Самостійна робота - **80 год.**

Перелік лабораторних робіт

1. Модульні тести (unit tests)
2. Веб-тести (web tests)
3. Тести навантаження (load tests)
4. Впорядковані тести (ordered tests)
5. Тести, залежні від даних (data-driven tests)
6. Розробка на основі тестів (test-driven development)
7. Тестування на основі перевірки станів (state verification) та поведінки (behavior verification)
8. Тестування з використанням підміни об'єктів (mock objects)
9. Mock object frameworks (на прикладі Rhino Mocks)

9. Рекомендовані джерела:

Основні

1. Л.П. Бабенко, К.М. Лавріщева. Основи програмної інженерії. К.: "Знання", 2001. – 269 с.
2. Э.М. Кларк, О. Грамберг, Д. Пелед, Верификация моделей программ: Model Checking. Пер. с англ. под ред. Р. Смелянского. – М.: 2002, 416 с.
3. С.В. Сеницын, Н.Ю. Налютин, Верификация программного обеспечения, Интернет-Университет Информационных Технологий, www.INTUIT.ru
4. И. Соммервилл, Инженерия программного обеспечения. – ИД «Вильямс», 2002. – 624 с.

Додаткові:

5. М. Кантор, Управление программными проектами. – ИД «Вильямс», 2002.- 176 с.
6. С.А. Орлов, Технологии разработки программного обеспечения.- СПб: «Питер», 2002.- 464.
7. Clarke E.M., Grumberg O., Peled D.: Model Checking. MIT Press (1999).
8. Microsoft Visual Studio 2005-2008, www.microsoft.com.