

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи


Олена КАШПУР

«12» листопада 2021 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
МЕТОДИ СПЕЦИФІКАЦІЇ ТА ВЕРИФІКАЦІЇ
ПРОГРАМ

для студентів

галузь знань 12 «Інформаційні технології»

(шифр і назва)

спеціальність 122 «Комп'ютерні науки»

(шифр і назва спеціальності)

освітній рівень бакалавр

(молодший бакалавр, бакалавр, магістр)

освітня програма «Інформатика»

(назва освітньої програми)

вид дисципліни вибіркова

вибірковий блок «Теорія та технологія програмування»

Форма навчання

денна

Навчальний рік

2022/2023

Семестр

6

Кількість кредитів ECTS

3

Мова викладання, навчання

та оцінювання

українська

Форма заключного контролю


залік


Викладачі: к.ф.-м.н., асистент Криволап А.В.

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)


КИЇВ – 2021

Розробники: Криволап Андрій Володимирович, к.ф.-м.н., асистент кафедри «Теорії та технології програмування», 

Омельчук Людмила Леонідівна, к.ф.-м.н., доцент кафедри «Теорії та технології програмування» 

ЗАТВЕРДЖЕНО


Зав. кафедри «Теорії та технології програмування»


(підпис)

Микола НІКІТЧЕНКО
(прізвище та ініціали)

Протокол № 6 від «11» лютого 2021 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»


(підпис)

Людмила ОМЕЛЬЧУК «11» лютого 2021 рік

(прізвище та ініціали)

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

Протокол від «11» лютого 2021 року № 7

Голова науково-методичної комісії 
(підпис)

Людмила ОМЕЛЬЧУК

(прізвище та ініціали)

1. Мета дисципліни – засвоєння базових знань з основ тестування, формальної специфікації та верифікації програмного забезпечення. Оволодіння базовими навичками проектування тестів для програмних систем, роботи з найбільш уживаними засобами тестування, засобами для формальної специфікації та верифікації.

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

1. *Знати:* основи реляційних баз даних та мови запитів SQL, технології розробки інформаційних програмних систем, принципи роботи технологій доступу до даних на прикладі ADO.Net, основи HTML, CSS, JavaScript, базові елементи програмної інженерії, принципи роботи технологій створення веб-застосунків на прикладі ASP.Net.

2. *Вміти:* працювати з технологією ADO.Net на автономному рівні, працювати з технологією ADO.Net Entity Framework, працювати з технологією ASP.Net.

3. *Володіти елементарними навичками:* програмування мовами C#.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Методи специфікації та верифікації програм” є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти *галузі знань* 12 „Інформаційні технології” зі *спеціальності* 122 „Комп’ютерні науки”, *вибірковий блок* „Теорія та технологія програмування” *освітньо-професійної програми* „Інформатика”.

Дана дисципліна є навчальною дисципліною за вибором *програми* “Інформатика” за *блоком дисциплін* “Теорія та технологія програмування”.

Викладається у 6 семестрі 3 курсу в **обсязі – 90 год. (3 кредити ECTS)** зокрема: *лекції – 42 год, консультації – 2 год., самостійна робота – 46 год.* У курсі передбачено 3 частини та 2 *контрольні роботи*. Завершується дисципліна – **заліком в 6 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

знати місце тестування в життєвому циклі ПС, основні види тестування, основні поняття верифікації ПС та формальних методів, особливості верифікації реактивних систем.

вміти застосовувати на практиці інструментальні програмні засоби автономного та інтеграційного тестування, задавати формальну специфікацію для програмних систем та використовувати її для доведення коректності, спілкуватися з колегами з питань якості програмного забезпечення.

Для допуску до дисципліни „Методи специфікації та верифікації програм” *спеціалізації* „Теорія та технологія програмування” освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Інструментальні середовища та технології програмування” програми «Інформатика». Дисципліна „Методи специфікації та верифікації програм” є базовою для засвоєння дисципліни «Інформаційні технології та правовий захист» *спеціалізації* „Теорія та технологія програмування” програми «Інформатика».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно до освітньої кваліфікації “Бакалавр з комп’ютерних наук”. Зокрема, розвивати:

- здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами, технічним завданням та стандартами;
- Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.2., РН 2.2 — 32 бали/19,2 балів.
2. Контрольна робота 2: РН 1.3. — 20 балів/12 балів.
3. Самостійна лабораторна робота 1 (проект): РН1.1, РН 2.1, РН3.1 – 10 балів/6 балів.
4. Самостійна лабораторна робота 2 (проект): РН1.1, РН 2.1, РН3.1 – 10 балів/6 балів.
5. Самостійна лабораторна робота 3 (проект): РН2.1, РН 4.1, РН4.2 – 10 балів/6 балів;
6. Поточне оцінювання: РН3.1, РН 4.1, РН4.2 – 8 балів/4,8 балів;

- підсумкове оцінювання у формі заліку. Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 9 тижня семестру.
2. Контрольна робота 2: до 12 тижня семестру.
3. Лабораторна робота 1 (проект): до 5 тижня семестру.
4. Лабораторна робота 2 (проект): до 6 тижня семестру.
5. Лабораторна робота 3 (проект): до 7 тижня семестру.
6. Поточне оцінювання: впродовж семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. 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 одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій

№ лекції	Назва лекції	Кількість годин		
		Семін. зан.	Практ. зан.	Сам. р-та
Частина 1. Тестування. Автоматичне та інтеграційне тестування				
1.	Тема 1. ЖЦ програмних систем. Місце тестування в ЖЦ ПС. <i>Самостійна робота: опрацювання лекційного матеріалу.</i>	2		2
2.	Тема 2. Основні поняття тестування. Види тестування. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	2		2
3.	Тема 3. Автономне тестування. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	2		2
4.-5.	Тема 4. Бібліотека jUnit. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	4		4
6.-7.	Тема 5. Бібліотека TestNG <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	4		4
8.	Тема 6. Інтеграційне тестування. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	2		4
9	Тема 7. Бібліотека EasyMock. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	2		4
10.-11.	Тема 8. Бібліотека Mockito. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	4		4
Всього по частині 1		22		26
Частина 2. Верифікація програмних систем				
12.	Тема 9. Основні поняття верифікації. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	2		2
13.-14.	Тема 10. Поняття формального методу, формальної специфікації. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	4		4
15.-17.	Тема 11. Верифікація ПС використовуючи RAISE. Основні можливості RSL. <i>Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.</i>	5		4
	<i>Контрольна робота 1</i>	1		
Всього по частині 2		12		10
Частина 3. Верифікація реактивних систем				

18.	Тема 12. Верифікація реактивних систем на прикладі IoT пристроїв. Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.	2		2
19.	Тема 13. Архітектура контролера на прикладі Arduino. Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.	2		4
20.- 21.	Тема 14. Поняття основного циклу виконання, сигналів та переривань. Самостійна робота: опрацювання лекційного матеріалу та виконання лабораторної роботи.	3		4
	Контрольна робота 2	1		
Всього по частині 3		8		10
Консультації			2	
ВСЬОГО		42	2	46

Загальний обсяг 90 год., в тому числі:

Лекцій – 42 год.

Консультації – 2 год.

Самостійна робота - 46 год.

Теми, винесені на самостійне вивчення:

Критерії якості ПЗ. Тестування прийняття. Автоматизація тестування. Мова формальних специфікацій Z.

Виконання лабораторних робіт 1-3.

Умови самостійних лабораторних робіт:

Лабораторна робота 1: Використання бібліотеки junit для автономного тестування.

Лабораторна робота 2: Використання бібліотеки TestNG для автономного тестування.

Лабораторна робота 3: Використання бібліотеки Mockito для інтеграційного тестування.

9. Рекомендовані джерела:

Основні

1. Лаврищева Е.М. Современные методы программирования: возможности и инструменты // Проблемы програмування. – 2006.– № 2-3. – С.60-74.
2. Буй Д.Б., Нікітченко М.С., Омельчук Л.Л., Редько В.Н., Шишацька О.В. Класифікація мов та методів формальних специфікацій програмних систем // Theoretical and applied aspects of program systems development (TAAPSD'2008). – Abstracts. – Berdyansk, Ukraine, 2007. P. 27-33.
3. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.
4. Бабенко Л.П., Лаврищева К.М. Основи програмної інженерії: Навч. посіб.–К.: Т-во "Знання", 2001.– 269 с.
5. Оушеров Р. Искусство автономного тестирования с примерами на С#.-М.:ДМК Пресс, 2014.- 360с.
6. Месарош Дж. Шаблоны тестирования xUnit. Рефакторинг кода тестов.-М.: Изд. дом "Вильямс", 2009.-832с.
7. Hanne Riis Nielson, Flemming Nielson Semantics with Applications. An Appetizer, Springer-Verlag.– 2007.– 274p.

Додаткові

8. Mike Spivey. The Z Notation: A Reference Manual, 2nd edition. Prentice Hall International Series in Computer Science, 1992.
9. <https://junit.org/junit4/>
10. <https://testng.org/doc/documentation-main.html>
11. <http://static.javadoc.io/org.mockito/mockito-core/2.23.0/org/mockito/Mockito.html>
12. <https://www.arduino.cc/reference/en/>