

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2018 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ТЕСТУВАННЯ ТА ВЕРИФІКАЦІЯ ПРОГРАМ

для студентів

галузь знань	12 «Інформаційні технології» <i>(шифр і назва)</i>
спеціальність	122 «Комп'ютерні науки» <i>(шифр і назва спеціальності)</i>
освітній рівень	бакалавр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Інформатика» <i>(назва освітньої програми)</i>
спеціалізація	«Теорія та технологія програмування» <i>(назва спеціалізації)</i>
вид дисципліни	вибіркова

Форма навчання	денна
Навчальний рік	2018/2019
Семестр	5
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 – 2018

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

ЗАТВЕРДЖЕНО

В.о. Зав. кафедри «Теорії та технології програмування»

_____ (Панченко Т.В.)
(підпис) (прізвище та ініціали)

Протокол № ____ від «____» _____ 20__ р.

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

Протокол від «____» _____ 20__ року № ____

Голова науково-методичної комісії _____ (Хусаїнов Д.Я.)
(підпис) (прізвище та ініціали)

«____» _____ 20__ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 2018 року № ____

Голова вченої ради факультету _____ А.В. Анісімов

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 вживаними 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 “Тестування та верифікація 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”, *спеціалізації* „Теорія та технологія програмування” *освітньо-професійної програми* „Інформатика”.

Дана дисципліна є навчальною дисципліною за вибором за *програмою* “Інформатика” за *спеціалізацією* “Теорія та технологія програмування”.

Викладається у 5 семестрі 3 курсу в **обсязі – 90 год.**

(3 *кредити ECTS*) зокрема: *лекції – 40 год, консультації – 2 год., самостійна робота – 48 год.* У курсі передбачено 3 частини та 3 контрольні роботи. Завершується дисципліна – заліком в 5 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 „Тестування та верифікація 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 „Тестування та верифікація програм” є базовою для засвоєння дисципліни «Коректність програм та логіки програмування» *спеціалізації* „Теорія та технологія програмування” програми «Інформатика».

4. Завдання (навчальні цілі):

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.2., РН 2.2 — 35 балів/21 балів.
2. Контрольна робота 2: РН 1.3. — 20 балів/12 балів.
3. Лабораторна робота 1 (проект): РН1.1, РН 2.1, РН3.1 – 10 балів/6 балів.
4. Лабораторна робота 2 (проект): РН1.1, РН 2.1, РН3.1 – 10 балів/6 балів.
5. Лабораторна робота 3 (проект): РН1.1, РН 2.1, РН3.1 – 10 балів/6 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.

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 випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

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

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
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.	2		2
2.	Тема 2. Основні поняття тестування. Види тестування.	2		2
3.	Тема 3. Автономне тестування.	2		2
4.	Тема 4. Бібліотека <i>jUnit</i> .	4		4
5.	Тема 5. Бібліотека <i>TestNG</i>	4		4
6.	Тема 6. Інтеграційне тестування.	2		4
7.	Тема 7. Бібліотека <i>EasyMock</i> .	2		4
8.	Тема 8. Бібліотека <i>Mockito</i> .	4		4
Всього по частині 1		22		26
Частина 2. Верифікація програмних систем				
9.	Тема 9. Основні поняття верифікації.	2		4
10.	Тема 10. Поняття формального методу, формальної специфікації.	2		4
11.	Тема 11. Верифікація ПС використовуючи <i>RAISE</i> . Основні можливості <i>RSL</i> .	6		4
	Контрольна робота 1			
Всього по частині 2		10		12
Частина 3. Верифікація реактивних систем				
12.	Тема 12. Верифікація реактивних систем на прикладі IoT пристроїв.	2		2
13.	Тема 13. Архітектура контролера на прикладі <i>Arduino</i> .	2		4
14.	Тема 14. Поняття основного циклу виконання, сигналів та переривань.	4		4
	Контрольна робота 2			
Всього по частині 3		8		10
Консультації			2	
ВСЬОГО		40	2	48

Загальний обсяг 90 год., в тому числі:

Лекцій – 40 год.

Консультації – 2 год.

Самостійна робота - 48 год.

Теми, винесені на самостійне вивчення:

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

Виконання лабораторних робіт 1-3.

Умови 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:

Основна

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. – Verdyansk, 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.

Додаткова:

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