

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра прикладної математики

Затверджено

На засіданні
кафедри прикладної математики
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 31 серпня 2023 р.)

Завідувач кафедри



Юрій ЯЩУК

Силабус з навчальної дисципліни
“Тестування програмного забезпечення”,
що викладається в межах першого (бакалаврського) рівня
вищої освіти для здобувачів
з спеціальності 113 – прикладна математика

Львів 2023 р.

Назва дисципліни	Тестування програмного забезпечення
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра оприкладної математики
Галузь знань, шифр та назва спеціальності	11 – математика та статистика 113 – прикладна математика
Викладачі дисципліни	Марчук Юрій Богданович, асистент кафедри прикладної математики, Переймибіда Андрій Андрійович, кандидат фізико-математичних наук, доцент кафедри прикладної математики (лабораторні заняття)
Контактна інформація викладачів	Yurii.Marchuk@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/marchuk-yu-biu-andrii.pereimybida@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/pereymybida-andriy-andriyovych Головний корпус ЛНУ ім. І. Франка, каб. 378. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю).
Сторінка курсу	https://ami.lnu.edu.ua/course/testuvannia-prohramnoho-zabezpechennia-pm
Інформація про дисципліну	Дисципліна “Тестування програмного забезпечення ” є дисципліною на вибір зі спеціальності 113 – прикладна математика, яка викладається в 7-му семестрі в обсязі 6-и кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Курс “Тестування програмного забезпечення” охоплює такі розділи: Модульне тестування, Ручне тестування, Автоматизоване тестування та Безперервна Інтеграція. При розгляді цих тем основна увага зосереджується на постійному контролю якості програмного забезпечення на кожному етапі розробки. Викладення матеріалу здійснюється за допомогою сучасних термінів та понять з галузі інформаційних технологій.
Мета та цілі дисципліни	Метою курсу є ґрунтовне ознайомлення студентів із підходами забезпечення якості програмного забезпечення. Завданням вивчення навчальної дисципліни є формування у студентів теоретичних знань в області контролю якості; здобуття студентами практичних навичок для перевірки якості програмного забезпечення на кожному етапі розробки.
Література для вивчення дисципліни	Основна література 1. Gerard Meszaros. xUnit test patterns. – Addison-Wesley: 2009. – 944 p. 2. Roy Osherove. The art of unit testing. – Manning Publications: 2013. – 292 p. 3. Srinivasan Desikan. Software Testing: Principles and Practices. //Srinivasan Desikan, Gopaldaswamy Ramesh – Pearson Education Canada: 2009. – 480 p. 4. Gayathri Mohan. Full Stack Testing. – O’Reilly Media, Inc: 2022 – 405 p. 5. Manfred Baumgartner. Test Automation Fundamentals. //Manfred Baumgartner Thomas Steirer, Marc-Florian Wendland, Stefan Gwihs, Julian Hartner, Richard Seidl – Rocky Nook: 2022 – 330 p. 6. Daniel Irvine. Build your own test framework: A practical guide to writing

	<p>better automated tests. – APress: 2023 – 280 p.</p> <p>7. Anton Angelov. Design patterns for high-quality automated tests. – Kindle Edition : 2020 – 314 p.</p> <p>8. Eran Kinsbruner. A frontend web developer’s guide to testing. – Packt Publishing: 2022 – 304 p.</p> <p>9. Alan J. Richardson. Automating and testing REST API. – Compendium Developments: 2009 – 272 p.</p> <p>10. Paul M. Duvall. Continuous integration: improving software quality and reducing risk. //Paul M. Duvall, Steve Matyas, Andrew Glover – Pearson Education: 2009 – 336 p.</p> <p>11. Nikhil Pathania. Learning Continuous Integration with Jenkins - Third Edition. – Packt Publishing: 2017 – 362 p.</p> <p>12. Shantanu Das. Continuous Integration and Continuous Deployment with Azure. – EC-Council: 2020 – 3h 55m</p> <p>13. Ian Molyneaux. The art of application performance testing: from strategy to tools. – O’Reilly Media Inc: 2014 – 278 p</p> <p>14. Antonio Gomes Rodrigues. Master Apache - From Load Testing to DevOps. //Antonio Gomes Rodrigues, Bruno Demion, Philippe Mouawad – Packt Publishing: 2019 – 468 p.</p> <p>15. Andrew Stellman. Learning Agile: Understanding Scrum, XP, Lean, and Kanban – Ascent Audio: 2021 – 15h 04m</p> <p>16. Lisa Crispin. Agile Testing Essentials. //Lisa Crispin, Janet Gregory – Pearson: 2017 – 04h 23m</p>
Обсяг курсу	Загальний обсяг: 180 год., аудиторних занять: 64 год., з них 32 години лекцій та 32 години лабораторних робіт. Самостійної роботи: 116 год.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p>Знати:</p> <ul style="list-style-type: none"> - підходи та шаблони розробки модульних тестів, - техніки розробки тестових випадків - підходи та шаблони розробки фреймворків автоматизованого тестування. <p>Вміти:</p> <ul style="list-style-type: none"> - розробляти модульні тести, - розробити тестову документацію: тест план, тестові випадки, звіт про тестування, - розробляти фреймворки для автоматизованого тестування backend- та web-додатків, - виконати тестування швидкодії web-додатку, - налаштувати сервер безперервної інтеграції для виконання тестів.
Ключові слова	Тестування, тестова документація, автоматизоване тестування, безперервна інтеракція, framework.
Формат курсу	Очний Проведення лекцій, лабораторних робіт і консультацій.
Теми	Подано нижче у таблиці Схема курсу “Тестування програмного забезпечення”
Підсумковий контроль, форма	Залік
Пререквізити	Для вивчення курсу студенти потребують базових знань з <ul style="list-style-type: none"> - Основ програмування

	<ul style="list-style-type: none"> - Практичні навички програмування - Основ баз даних - Основ серверних сервісів(backend services) - Основ Web-додатків 																																			
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентації, лекції, модульний контроль. Індивідуальні завдання																																			
Необхідне обладнання	Комп'ютер із програмним забезпеченням: мова програмування на вибір студента, інтегроване середовище розробки(IDE – Integrated Development Environment).																																			
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою.																																			
	<table border="1"> <thead> <tr> <th colspan="2" rowspan="2">Оцінка за шкалою ECTS</th> <th rowspan="2">Оцінка в балах</th> <th colspan="2">Оцінка за національною шкалою</th> <th rowspan="2">залік</th> </tr> <tr> <th colspan="2">Екзамен, диференційований залік</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Відмінно</td> <td>100 - 90</td> <td>Відмінно</td> <td>5</td> <td rowspan="4">зараховано</td> </tr> <tr> <td>B</td> <td>Дуже добре</td> <td>81- 89</td> <td rowspan="2">Добре</td> <td rowspan="2">4</td> </tr> <tr> <td>C</td> <td>Добре</td> <td>71 -80</td> </tr> <tr> <td>D</td> <td>Задовільно</td> <td>61 - 70</td> </tr> <tr> <td>E</td> <td>Достатньо</td> <td>51- 60</td> <td>Задовільно</td> <td>3</td> <td rowspan="2">не зараховано</td> </tr> <tr> <td>FX (F)</td> <td>Незадовільно</td> <td>0 - 50</td> <td>Незадовільно</td> <td>2</td> </tr> </tbody> </table>	Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою		залік	Екзамен, диференційований залік		A	Відмінно	100 - 90	Відмінно	5	зараховано	B	Дуже добре	81- 89	Добре	4	C	Добре	71 -80	D	Задовільно	61 - 70	E	Достатньо	51- 60	Задовільно	3	не зараховано	FX (F)	Незадовільно	0 - 50	Незадовільно
Оцінка за шкалою ECTS					Оцінка в балах	Оцінка за національною шкалою		залік																												
		Екзамен, диференційований залік																																		
A	Відмінно	100 - 90	Відмінно	5	зараховано																															
B	Дуже добре	81- 89	Добре	4																																
C	Добре	71 -80																																		
D	Задовільно	61 - 70																																		
E	Достатньо	51- 60	Задовільно	3	не зараховано																															
FX (F)	Незадовільно	0 - 50	Незадовільно	2																																
	<p>Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> • індивідуальні завдання: 80% семестрової оцінки; максимальна кількість балів 80 (3 завдання: 25, 25 та 30 балів); 																																			
	<table border="1"> <thead> <tr> <th>№ 1 (25 балів)</th> <th>№ 2 (25 балів)</th> <th>№ 3 (30 балів)</th> <th>Критерії оцінювання</th> </tr> </thead> <tbody> <tr> <td>25 балів</td> <td>25 балів</td> <td>30 балів</td> <td>студент повністю виконав умови завдання, алгоритм реалізовано правильно, відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз та порівняння отриманих результатів, пропонує інші підходи до вирішення поставленого завдання;</td> </tr> <tr> <td>17-24 балів</td> <td>17-24 балів</td> <td>20-29 балів</td> <td>студент повністю виконав умови завдання, на деякі запитання, алгоритм реалізовано правильно, пов'язані з тематикою завдання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями;</td> </tr> <tr> <td>10-16 балів</td> <td>10-16 балів</td> <td>13-19 балів</td> <td>студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями;</td> </tr> <tr> <td>6-9 балів</td> <td>6-9 балів</td> <td>8-12 балів</td> <td>студент виконав завдання частково, алгоритм реалізовано з помилками, які</td> </tr> </tbody> </table>	№ 1 (25 балів)	№ 2 (25 балів)	№ 3 (30 балів)	Критерії оцінювання	25 балів	25 балів	30 балів	студент повністю виконав умови завдання, алгоритм реалізовано правильно, відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз та порівняння отриманих результатів, пропонує інші підходи до вирішення поставленого завдання;	17-24 балів	17-24 балів	20-29 балів	студент повністю виконав умови завдання, на деякі запитання, алгоритм реалізовано правильно, пов'язані з тематикою завдання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями;	10-16 балів	10-16 балів	13-19 балів	студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями;	6-9 балів	6-9 балів	8-12 балів	студент виконав завдання частково, алгоритм реалізовано з помилками, які															
№ 1 (25 балів)	№ 2 (25 балів)	№ 3 (30 балів)	Критерії оцінювання																																	
25 балів	25 балів	30 балів	студент повністю виконав умови завдання, алгоритм реалізовано правильно, відповідає на всі запитання, пов'язані з тематикою завдання, проводить чіткий аналіз та порівняння отриманих результатів, пропонує інші підходи до вирішення поставленого завдання;																																	
17-24 балів	17-24 балів	20-29 балів	студент повністю виконав умови завдання, на деякі запитання, алгоритм реалізовано правильно, пов'язані з тематикою завдання, відповідає з незначними неточностями, проводить аналіз отриманих результатів з незначними неточностями;																																	
10-16 балів	10-16 балів	13-19 балів	студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач, на деякі запитання, пов'язані з тематикою завдання, відповідає з неточностями, проводить аналіз отриманих результатів з неточностями;																																	
6-9 балів	6-9 балів	8-12 балів	студент виконав завдання частково, алгоритм реалізовано з помилками, які																																	

			частково може виправити, якщо на них вкаже викладач, на запитання відповідає з помилками, проводить аналіз отриманих результатів з помилками;
4-5 бали	4-5 бали	4-7 бали	студент виконав завдання частково, алгоритм реалізовано з помилками, які самостійно не може виправити, переважно не відповідає на запитання;
1-3 бал			студент виконав завдання частково або з грубими помилками, які самостійно не може виправити, демонструє незнання матеріалу;
0 балів			студент не виконав завдання.
<p>• контрольні заміри (модулі): 20% семестрової оцінки; максимальна кількість балів 20 (4 тестові модулі в середовищі MS Teams по 5 балів);</p> <p>Критерії оцінювання тестових завдань (модулі): 1 бал: відповідь на завдання правильна; 0 балів: відповідь на завдання неправильна.</p> <p>Загалом протягом семестру 100 балів. Академічна доброчесність: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання домашніх та індивідуальних завдань, передбачених курсом. Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих. Політика виставлення балів. Враховуються бали набрані при поточному тестуванні та самостійній роботі. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін. Жодні форми порушення академічної доброчесності не толеруються.</p>			
Питання до контрольних замірів	<ol style="list-style-type: none"> 1. Розробка модульних тестів. 2. Автоматизація тестування 3. Тестова документація. 4. Безперервна інтеграція. 		
Опитування	Оцінювання якості курсу буде доступне у системі "Dekanat" після завершенню курсу. Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.		

Схема курсу “Тестування програмного забезпечення ”

№ п/п	Тема, план, короткі тези	Форма діяльності (заняття)	Література, Ресурси в інтернеті	Завдання, год	Термін виконання
1	Тема 1. Вступ (<i>Означення тестування. Прикладні задачі</i>)	Лекція (2 год)		Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 1. Формулювання процесу роботи над індивідуальними завданнями, процедури задачі	Лабораторна (2 год)		Налаштування локального середовища (5 год.)	2 тижні
2	Тема 2. Основи модульного тестування	Лекція (2 год)	[1, 2, 3]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 2. Розробка модульних тестів	Лабораторна (2 год)	[1, 2, 3]	Розробка модульних тестів, перевірка покриття (5год.)	1 тиждень
3	Тема 3. Шаблони розробки модульних тестів.	Лекція (4 год)	[1, 2, 3]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 3. Розробка модульних тестів	Лабораторна (2 год)	[1, 2, 3]	Налаштування мутаційного тестування, використання mock-та stubs (5 год.)	2 тижні
4	Тема 4. Презентації та оцінювання індивідуального завдання №1	Лабораторна (2 год)		Захист індивідуального завдання	
5	Тема 5. Документування процесу тестування. Підходи розробки тестових випадків	Лекція (2 год)	[4, 5]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 5. Розробка тестових випадків	Лабораторна (2 год)	[4, 5]	Розробка тестових випадків для функціонального тестування та нефункціонального тестування (10 год.)	2 тижні
6	Тема 6. Основи автоматизації тестування. Розробка фреймворку	Лекція (4 год)	[6, 7, 8]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 6. Розробка фреймворку для автоматизованого тестування	Лабораторна (2 год)	[6, 7, 8]	Розробка основи фреймворку на прикладі мов JavaScript та Python(5 год.)	2 тижні
7	Тема 7. Підходи автоматизації тестування backend-додатків	Лекція (2 год)	[6, 7, 8, 10]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 7. Розробка фреймворку для	Лабораторна (2 год)	[6, 7, 8, 10]	Розробка фреймворку для	2 тижні

	тестування бекенд сервісу(REST API та gRPC)			тестування бекенд сервісу(5 год.)	
8	Тема 8. Презентації та оцінювання індивідуального завдання №2	Лабораторна (2 год)		Захист індивідуального завдання	
9	Тема 9. Підходи автоматизації тестування Web-додатків. Веб-елементи та їх локатори.	Лекція (2 год)	[6, 7, 9]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 9. Розробка фреймворку для тестування web-додатків	Лабораторна (2 год)	[6, 7, 9]	Розробка фреймворку для тестування Web UI додатків(5 год.)	2 тижні
10	Тема 10. Підходи автоматизації тестування Web-додатків. Шаблони розробки.	Лекція (2 год)	[6, 7, 9]	Опрацювання лекційного матеріалу (3год.)	1 тиждень
	Тема 10. Розробка фреймворку для тестування web-додатків	Лабораторна (2 год)	[6, 7, 9]	Розробка фреймворку для тестування Web UI додатків з п'ятьма рівнями класів(5 год.)	2 тижні
11	Тема 11. Основи безперервної інтеграції	Лекція (4 год)	[11, 12]	Опрацювання лекційного матеріалу (6год.)	1 тиждень
	Тема 11. Сервери безперервної інтеграції	Лабораторна (2 год)	[11, 12]	Основи налаштувань серверів безперервної інтеграції на прикладі Jenkins та Azure DevOps (5 год.)	2 тижні
12	Тема 12. Презентації та оцінювання індивідуального завдання №3	Лабораторна (2 год)		Захист індивідуального завдання	
13	Тема 13. Основи тестування швидкодії.	Лекція (2 год)	[14, 15]	Опрацювання лекційного матеріалу (4год.)	1 тиждень
	Тема 13. Розробка скриптів тестування швидкодії	Лабораторна (2 год)	[14, 15]	Розробка скриптів тестування швидкодії засобами JMeter(10 год.)	2 тижні
14	Тема 14. Моделі розробки програмного забезпечення.	Лекція (2 год)	[16]	Опрацювання лекційного матеріалу (4год.)	1 тиждень
	Тема 14. Контроль якості в гнучких моделях розробки програмного забезпечення.	Лекція (2 год)	[16]	Опрацювання лекційного матеріалу (4год.)	1 тиждень
15	Тема 15 Підходи розробки тест плану	Лекція (2 год)	[4, 5, 6, 11, 14]	Опрацювання лекційного матеріалу (4год.)	1 тиждень
	Тема 15. Розробка тест плану.	Лабораторна (2 год)	[4, 5, 6, 11,	Розробка загального тест плану, тест	2 тижні

			14]	плану функціонального тестування, тест плану тестування швидкодії (10 год.)	
16	Тема 16. Презентації та оцінювання індивідуального завдання №4	Лабораторна (4 год)		Захист індивідуального завдання	