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

Затверджено
на засіданні кафедри інформаційних систем
факультету прикладної математики та інформатики
Львівського національного університету імені Івана Франка
(протокол № 2 від 20.09 2022 р.)

Завідувач кафедри Шинкаренко Г.А.

Силабус з навчальної дисципліни

"Надійність програмних систем",

що викладається в межах ОПП «Інформатика» другого (магістерського) рівня вищої освіти для здобувачів з спеціальності 122 – Комп'ютерні науки

Львів 2022 р.

Назва дисципліни	**Надійність програмних систем**
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 інформаційних систем
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 122 – комп'ютерні науки
Викладачі дисципліни	Жолткевич Григорій Миколайович, професор кафедри інформаційних систем, доктор техн. наук. (лекції та лабораторні заняття)
Контактна інформація викладачів	grygoriy.zholtkevych@lnu.edu.ua; Головний корпус ЛНУ ім. І. Франка, каб. 260. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять (а також за розкладом консультацій кафедри).
Сторінка курсу	https://ami.lnu.edu.ua/course/nadiynist-prohramnykh-system-122-cs
Інформація про дисципліну	Дисципліна "**Надійність програмних систем**" є вибірковою дисципліною для спеціальності 122 – комп'ютерні науки для освітньої програми Інформатика, яка викладається в 1-му семестрі в обсязі 6-ти кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Мета та цілі дисципліни	***Метою*** *навчальної дисципліни є формування у студентів системи теоретичних знань і практичних навичок аналізу коректності програмного забезпечення з використанням програмних інструментів класу «помічник у доведенні (proof assistant)».* ***Завданням*** *вивчення навчальної дисципліни є сформувати у студентів теоретичні знання та практичні навички в області статичного аналізу програмного забезпечення; надати студентам уявлення про завдання та цілі формальної верифікації програмного забезпечення; надати студентам теоретичні засади та практичні навички використання The Coq Proof Assistant для статичного аналізу програм.*
Коротка анотація дисципліни	Навчальну дисципліну розроблено так, щоб забезпечити розуміння студентами концепції надійності технічних систем, її особливості для програмних систем, а також ознайомити студентів з сучасною технологією забезпечення розробки надійного програмного забезпечення – сертифікованим програмуванням, спираючись на можливості The Coq Proof Assistant та сформувати загальні навички її використання. Курс представляє собою приклади застосування технології сертифікованого програмування для низки простих але типових задач програмування з

	поясненням загальних концепцій на цих прикладах. Кожна лекція передбачає обговорення на лабораторному занятті з широким залученням студентів до такого обговорення.
Література для вивчення дисципліни	Основна література: 1. Chlipala, Adam. Certified Programming with Dependent Types, MIT Press 2. Bertot, Yves and Casteran, Pierre. Interactive Theorem Proving and Program Development. Coq' Art: The Calculus of Inductive Constructions, Springer 3. Paulin-Mohring, Christine. Introduction to the Coq proof-assistant for practical software verification, Lecture Notes in Computer Science, vol 7682. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35746-6_3 4. The Coq Proof Assistant. Reference Manual. https://coq.inria.fr/distrib/current/refman/ 5. The Coq Proof Assistant. Standard Library https://coq.inria.fr/distrib/current/stdlib/ Додаткова література: 1. Software Foundations Volume 1. Logical Foundation https://softwarefoundations.cis.upenn.edu/current/lf-current/index.html 2. Software Foundations Volume 2. Programming Language Foundations https://softwarefoundations.cis.upenn.edu/current/plf-current/index.html 3. Software Foundations Volume 3. Verified Functional Algorithms https://softwarefoundations.cis.upenn.edu/current/vfa-current/index.html 4. Software Foundations Volume 5. Verifiable C https://softwarefoundations.cis.upenn.edu/current/vc-current/index.html
Обсяг курсу	Загальний обсяг: 180 годин. Аудиторних занять: 64 год., з них 32 години лекцій та 32 години лабораторних занять. Самостійної роботи: 116 годин.

Очікувані результати навчання	**У результаті вивчення навчальної дисципліни студент повинен** ***знати:*** • концепцію надійності програмних систем, її особливості у контексті загального розуміння надійності технічних систем; • концепцію технології сертифікованого програмування і її теоретичні засади; • архітектуру системи The Coq Proof Assistant; • основні тактики системи The Coq Proof Assistant. ***вміти:*** • моделювати семантичні сутності за допомогою індуктивних визначень; • використовувати функціональну мову системи The Coq Proof Assistant; • використовувати основні тактики системи The Coq Proof Assistant для доведення тверджень про програми; • використовувати модулі системи The Coq Proof Assistant для запобігання конфліктам іменування; • використовувати інструмент класів для побудови фреймворків верифікації.
	Курс забезпечує набуття таких компетентностей: ІК, ЗК 1, ЗК 3-7, СК 1-6, СКФ 10 та програмних результатів навчання: ПРН 1-2, ПРН 6-7, ПРН 13, ПРН 16, ПРН 20.
Ключові слова	Технічна система, програмна система, надійність, статичний аналіз, формальна верифікація, теорія типів, тактика, доведення
Формат курсу	Очний, дистанційний. Проведення лекцій, лабораторних робіт і консультацій.
Теми	**Тема 1.** *Концепція надійності програмних систем* **Тема 2.** *Задача розробки простого програмованого стекового обчислювача* **Тема 3.** *Задача розробки програмованого стекового обчислювача з типізацією значень* **Тема 4.** *Загальне уявлення про The Coq Proof Assistant* **Тема 5.** *Індуктивні визначення* **Тема 6.** *Функціональне програмування з The Coq Proof Assistant* **Тема 7.** *Автоматизація доведення тверджень про програми* **Тема 8.** *Верифікація алгоритмів сортування* **Тема 9.** *Верифікація деяких базових структур даних*
Підсумковий контроль, форма	Залік у кінці першого семестру.

Пререквізити	Для вивчення дисципліни студенти потребують базові знання, що формуються при підготовці рівня бакалавр з математичної логіки, дискретної математики, основ функціонального програмування, теорії алгоритмів
Навчальні методи та техніки, які будуть використовувати ся під час викладання курсу	Презентації, лекції, модульний контроль Індивідуальні завдання Індивідуальні мініпроєкти
Необхідне обладнання	Комп'ютер, Internet.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням: • модульний контроль, тестування, усне опитування: 40% семестрової оцінки; максимальна кількість балів 40; • індивідуальні завдання: 60% семестрової оцінки; максимальна кількість балів 60. Підсумкова максимальна кількість балів 100.
	Академічна доброчесність: Очікується, що роботи студентів будуть їх самостійними 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. **Відвідування 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 всіх лабораторних робіт 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, отримані при поточному опитуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнень 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.
Питання до екзамену	Екзамен не передбачений
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенні курсу.

Схема курсу

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
1	**Тема 1. Концепція надійності програмних систем** (надійність технічних систем, технічні системи з інтенсивним використанням програмного забезпечення, відомі катастрофічні інциденти, викликані помилками програмного забезпечення, два підходи до розуміння надійності програмних систем, концепція сертифікованого програмування)	лекція, самостійна робота		2	1 тиждень
1	**Тема 2. Задача розробки простого програмованого стекового обчислювача** (встановлення The Coq Proof Assistant, інтерактивного середовища coqide, основи роботи з цими інструментами)	лабораторна робота	[4-5]		
2	**Тема 2. Задача розробки простого програмованого стекового обчислювача** (моделювання мови виразів, моделювання стекового обчислювача, розробка компілятора, доведення коректності компілятора)	лекція, самостійна робота	[1-5]	2	1 тиждень
2	**Тема 2. Задача розробки простого програмованого стекового обчислювача** (аналіз мініпроєкту)	лабораторна робота	[4-5]	2	1 тиждень
3	**Тема 3. Задача розробки програмованого стекового обчислювача з типізацією значень** (моделювання мови типізованих виразів, концепція родин типів, моделювання стекового обчислювача, що підтримує типізацію значень)	лекція, самостійна робота	[1-5]	2	1 тиждень
3	**Тема 3. Задача розробки програмованого стекового обчислювача з типізацією значень** (розробка компілятора, доведення коректності компілятора)	лекція, самостійна робота	[1-5]	2	1 тиждень
4	**Тема 3. Задача розробки програмованого стекового обчислювача з типізацією значень** (аналіз мініпроєкту)	лабораторна робота	[4-5]	4	2 тижні
4	**Тема 3. Задача розробки програмованого стекового обчислювача з типізацією значень** (аналіз мініпроєкту)	лабораторна робота	[4-5]		
5	**Тема 4. Загальне уявлення про The Coq Proof Assistant** (архітектурні компоненти системи, мови системи, концепція «терм-тип-сорт»)	лекція, самостійна робота	[1-4]		
5	**Тема 5. Індуктивні визначення** (перелічувані типи, рекурсивні типи, принципи індукції)	лекція, самостійна робота	[1-4]		

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
6	**Тема 5. Індуктивні визначення** (практикум з визначення індуктивних типів)	лабораторна робота	[4-5]	2	1 тиждень
6	**Тема 6. Функціональне програмування з The Coq Proof Assistant** (основні конструкції мови Galinna)	лекція, самостійна робота	[1-4]		
7	**Тема 6. Функціональне програмування з The Coq Proof Assistant** (залежні типи, специфіка застосування основних конструкцій мови Galinna у випадку залежних типів)	лекція, самостійна робота	[1-4]		
7	**Тема 6. Функціональне програмування з The Coq Proof Assistant** (залежні типи, специфіка застосування основних конструкцій мови Galinna у випадку залежних типів)	лабораторна робота	[4-5]	2	1 тиждень
8	**Тема 6. Функціональне програмування з The Coq Proof Assistant** (залежні типи, специфіка застосування основних конструкцій мови Galinna у випадку залежних типів)	лабораторна робота	[4-5]	2	1 тиждень
8	**Тема 7. Автоматизація доведення тверджень про програми** (основні тактики, їх застосування, розробка тактик)	лекція, самостійна робота	[1-4]		
9	**Тема 7. Автоматизація доведення тверджень про програми** (простори імен, модульні типи, класи)	лекція, самостійна робота	[1-4]		
9	**Тема 7. Автоматизація доведення тверджень про програми** (практикум із застосування і розробки тактик)	лабораторна робота	[4-5]	2	1 тиждень
10	**Тема 7. Автоматизація доведення тверджень про програми** (практикум з управління просторами імен, модульних типів, класів)	лабораторна робота	[4-5]	2	1 тиждень
10	**Тема 8. Верифікація алгоритмів сортування** (моделювання постановки задачі сортування засобами The Coq Proof Assistant)	лекція, самостійна робота	[1-4]	24	3 тижні
11	**Тема 8. Верифікація алгоритмів сортування** (верифікація алгоритму сортування злиттям)	лекція, самостійна робота	[1-4]		
11	**Тема 9. Верифікація деяких базових структур даних** (побудова теорій стеків та черг)	лекція, самостійна робота	[1-4]		
12	**Тема 9. Верифікація деяких базових структур даних** (побудова теорії словників)	лекція, самостійна робота	[1-4]		
12	**Тема 9. Верифікація деяких базових структур даних** (побудова теорії бінарних дерев)	лекція, самостійна робота	[1-4]		
13	**Тема 9. Верифікація деяких базових структур даних** (побудова теорії B-дерев)	лекція, самостійна робота	[1-4]	2	1 тиждень

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
13	**Тема 9. Верифікація деяких базових структур даних** (аналіз алгоритмів швидкого сортування та сортування Хоара)	лабораторна робота	[4-5]	2	1 тиждень
14	**Тема 9. Верифікація деяких базових структур даних** (аналіз теорій стеків та черг)	лабораторна робота	[4-5]	2	1 тиждень
14	**Тема 9. Верифікація деяких базових структур даних** (аналіз теорії словників)	лабораторна робота	[4-5]	2	1 тиждень
15	**Тема 9. Верифікація деяких базових структур даних** (аналіз теорії бінарних дерев)	лабораторна робота	[4-5]	2	1 тиждень
15	**Тема 9. Верифікація деяких базових структур даних** (аналіз теорії бінарних дерев)	лабораторна робота	[4-5]	2	1 тиждень
16	**Тема 9. Верифікація деяких базових структур даних** (аналіз теорії B-дерев)	лабораторна робота	[4-5]	2	1 тиждень
16	**Тема 9. Верифікація деяких базових структур даних** (підсумковий контроль)	лабораторна робота	[4-5]	2	