

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

Затверджено

На засіданні
кафедри обчислювальної математики
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 29 серпня 2023 р.)



Завідувач кафедри

 Роман ХАПКО

Силабус з навчальної дисципліни
“Розробка програмного забезпечення з відкритим кодом”,
що викладається в межах ОПП Прикладна математика
другого (магістерського) рівня вищої освіти для здобувачів з
спеціальності 113 – прикладна математика

Львів 2023 р.

Назва дисципліни	Розробка програмного забезпечення з відкритим кодом
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 обчислювальної математики
Галузь знань, шифр та назва спеціальності	11 – математика і статистика 113 – прикладна математика
Викладачі дисципліни	Вавричук Василь Григорович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики
Контактна інформація викладачів	vasyi.vavrychuk@lnu.edu.ua, https://ami.lnu.edu.ua/employee/vavrychuk Головний корпус ЛНУ ім. І. Франка, каб. 360. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять, за розкладом консультацій кафедри, а також в середовищі Microsoft Teams.
Сторінка курсу	https://ami.lnu.edu.ua/course/rozrobka-prohramnoho-zabezpechennia-z-vidkrytym-kodom
Інформація про дисципліну	Дисципліна "Розробка програмного забезпечення з відкритим кодом" є вибірковою дисципліною для спеціальності 113 - прикладна математика для освітньої програми "Прикладна математика", яка викладається в 3-му семестрі в обсязі 3-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Для більшості це буде несподіванкою, але комфорт нашого життя є високим у великій мірі завдячуючи програмним рішенням з відкритим кодом. Тобто таким, що за означенням доступні для аналізу, а інтеграція та копіювання яких обмежується ліцензією, що зазвичай вимагає лише згадки про використання відповідного програмного забезпечення, або у більш жорсткому випадку відкриття коду успадкованого програмного забезпечення. Наприклад, масштаб використання ПЗ з відкритим кодом можна побачити на прикладі Android смартфона відкривши меню "Налаштування > Про телефон > Нормативно-правова інформація > Правова інформація > Ліцензії третьої сторони". Розробка ПЗ з відкритим кодом зазвичай відбувається за участі спільноти і процес розробки (тікети, рецензування коду і т.д.) доступні для широкого загалу. Успіх ПЗ з відкритим кодом має глибокі психологічні корені: очевидно, що ми працюємо краще, коли наша робота доступна всім для аналізу. У курсі розглядаються просунуті засоби SCM, які є абсолютною необхідністю при розробці ПЗ з відкритим кодом, адже вони забезпечують чистоту історії коду, що є особливо важливою проблемою при великій кількості учасників в розробці, поширеній практиці рецензування коду. Також, розглядаються два достатньо великих, загальновідомих та відмінних проекти з відкритим кодом.
Мета та цілі дисципліни	Метою навчальної дисципліни є знайомство студентів з рухом розробки ПЗ з відкритим кодом, набуття базових вмінь, які є абсолютною необхідністю для участі в проектах з відкритим кодом.

Література для вивчення дисципліни	<p>Основна література:</p> <ol style="list-style-type: none"> 1. The Architecture of Open Source Applications (AOSA) Volume 1-2. - https://aosabook.org/en/ 2. Edwin Kempin. Git Concepts and Workflows. - Google Munich. - https://tinyurl.com/mwbtb6et 3. https://github.com/microsoft/vscode/wiki/How-to-Contribute 4. https://gerrit-documentation.storage.googleapis.com/Documentation/3.8.1/index.html 5. https://gerrit.googlesource.com/gerrit/+refs/heads/master/README.md <p>Додаткова література:</p> <ol style="list-style-type: none"> 6. Eric Steven Raymond. The Cathedral and the Bazaar. - http://www.catb.org/~esr/writings/cathedral-bazaar/ 7. Маніфест GNU, https://www.gnu.org/gnu/manifesto.html 8. https://github.com/chernivtsijs/made-in-ukraine 9. https://github.com/collections/made-in-ukraine
Обсяг курсу	<p>Загальний обсяг: 90 годин. Аудиторних занять: 32 год., з них 16 год. лекцій та 16 год. лабораторних робіт. Самостійної роботи: 58 год.</p>
Очікувані результати навчання	<p>Після завершення цього курсу студент буде знати:</p> <ul style="list-style-type: none"> • про місце програмного забезпечення з відкритим кодом у ІТ індустрії • загальноприйняті практики при розробці ПЗ з відкритим кодом • архітектуру деяких проектів з відкритим кодом <p>вміти:</p> <ul style="list-style-type: none"> • користуватися інструментами контролю версій • аналізувати документації проектів з відкритим кодом • аналізувати код проектів • збирати проект з вихідного коду • розширювати проект своїми змінами
Ключові слова	<p>ІТ, програмування, прикладна математика</p>
Формат курсу	<p>Очний</p>
Теми	<p>Подано нижче у таблиці «Схема курсу»</p>
Підсумковий контроль, форма	<p>Залік</p>
Пререквізити	<p>Для вивчення курсу студенти потребують базових знань з програмування.</p>
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	<p>Презентації, лекції, модульний контроль, індивідуальні завдання.</p>
Необхідне обладнання	<p>Комп'ютер, Internet.</p>
Критерії оцінювання (окремо для	<p>Під час семестру студент може отримати 100 балів з них:</p> <ul style="list-style-type: none"> • 20 балів за індивідуальне завдання • 20 балів за модульний контроль

<p>кожного виду навчальної діяльності)</p>	<ul style="list-style-type: none"> • 60 балів за індивідуальний проект <p>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.</p> <p>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 всіх видів письмових робіт та індивідуальних завдань, передбачених курсом.</p> <p>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.</p> <p>Політика виставлення балів. Враховуються бали набрані за індивідуальне завдання, модульний контроль та індивідуальний проект. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного 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.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Питання до модульного контролю.</p>	<ol style="list-style-type: none"> 1. Створення комітів. 2. Перегляд Git історії. 3. Робота з гілками. 4. Рекомендації щодо повідомлень комітів. 5. Додавання проміжних комітів в історії, видалення та об'єднання. 6. Редагування комітів. 7. Збирання VS Code. 8. VS Code розширення. 9. Збирання Gerrit. 10. Gerrit розширення.
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

Схема курсу “Розробка програмного забезпечення з відкритим кодом”

Ти ж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в інтернеті	Завдання, год.	Термін виконання
1	<p>Тема 1. Вступ Огляд проектів з відкритим кодом, GNU, Linux.</p>	<p>Лекція (2 год.)</p>	<p>[1],[6-9]</p>	<p>Опрацювання лекційного матеріалу (3 год.)</p>	<p>1 тиждень</p>
	<p>Тема 1. Вступ Пошук завдань для новачків.</p>	<p>Лабораторна робота (2 год.)</p>	<p>http://opensource.guide/how-to-contribute/#finding-a-project-to-</p>	<p>Виконання індивідуального проекту (3 год.)</p>	<p>1 тиждень</p>

			contribute-to		
2	Тема 2. SCM система Git: основи роботи Історичні SCM системи, централізовані та розподілені SCM системи. Репозиторій, робоча копія, внутрішня структура репозиторію (blob, tree, commit, refs, і т.д.). Внесення змін, перегляд історії, переміщення по історії, індекс.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 2. SCM система Git: основи роботи Індивідуальне завдання №1: Робота з SCM системою Git.	Лабораторна робота (2 год.)	[2]	Виконання індивідуального завдання (3 год.)	1 тиждень
3	Тема 3. SCM система Git: робота з гілками Створення, видалення гілок, переміщення між гілками, об'єднання гілок.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
	Тема 3. SCM система Git: робота з гілками Паралельне внесення змін у код проекту у різних гілках та об'єднання цих гілок.	Лабораторна робота (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
4	Тема 4. Хороші практики SCM Рекомендації щодо повідомлень комітів, переписування історії.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
	Тема 4. Хороші практики SCM Налаштування пре-коміт хуків, CI.	Лабораторна робота (2 год.)	[2], https://pre-commit.com	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
5	Тема 5. Керування історією (серіями	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу,	1 тиждень

	патчів) SCM Об'єднання, видалення, модифікація комітів.			виконання індивідуального завдання (3 год.)	
	Тема 5. Керування історією (серіями патчів) SCM Модульний контроль по Git.	Лаборатор на робота (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
6	Тема 6. VS Code Node.js, збирання, внесення змін, відлагодження, розширення.	Лекція (2 год.)	[3]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 6. VS Code Здача індивідуального завдання №1. Розбір завдань на індивідуальний проект.	Лаборатор на робота (2 год.)	[3]	Виконання індивідуального проекту (4 год.)	1 тиждень
7	Тема 7. Gerrit Налаштування, збирання з коду, високорівнева структура.	Лекція (2 год.)	[4-5]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 7. Gerrit Розбір завдань на індивідуальний проект.	Лаборатор на робота (2 год.)	[4-5]	Виконання індивідуального проекту (4 год.)	1 тиждень
8	Тема 8. Підсумкова лекція Огляд альтернативних проектів з відкритим кодом. Консультації щодо виконання індивідуальних проектів.	Лекція (2 год.)	https://github.com/explore . Debian	Виконання індивідуального проекту (4 год.)	1 тиждень
	Захист індивідуальних проектів.	Лаборатор на робота (2 год.)			1 тиждень