

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

Оновлено та затверджено
на засіданні
кафедри обчислювальної математики
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 6 від 21 грудня 2023р.)



Завідувач кафедри

 Роман ХАПКО

Силабус з навчальної дисципліни
«Розробка програмного забезпечення з відкритим кодом»,
що викладається в межах ОНП 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	vasy1.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-prykladna-matematyka-1-9
Інформація про дисципліну	Дисципліна «Розробка програмного забезпечення з відкритим кодом» є вибірковою дисципліною для спеціальності 113 - прикладна математика для освітньо-наукової програми «Прикладна математика», яка викладається в 3-му семестрі в обсязі 4,5 кредити (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 видима для всіх. У курсі розглядаються просунуті засоби SCM, які є 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 необхідними 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	Загальний обсяг: 135 годин. Аудиторних занять: 48 год., з них 32 год. лекцій та 16 год. лабораторних робіт. Самостійної роботи: 87 год.
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	ІТ, програмування, прикладна математика
Формат курсу	Очний
Теми	Подано нижче у таблиці «Схема курсу»
Підсумковий контроль, форма	Залік
Пререквізити	Для вивчення курсу студенти потребують базових знань з програмування.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентації, лекції, модульний контроль, індивідуальні завдання.
Необхідне обладнання	Комп'ютер, Internet.

Критерії оцінювання (окремо для кожного виду навчальної діяльності)

Оцінювання проводиться за 100-бальною шкалою.

Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національного шкалою
A	Відмінно	100 - 90	зараховано
B	Дуже добре	81- 89	
C	Добре	71 -80	
D	Задовільно	61 - 70	
E	Достатньо	51- 60	
FX (F)	Незадовільно	0 - 50	не зараховано

Під час семестру студент може отримати 100 балів з них:

- по 10 балів за завдання №1, 2, 3.
- 10 балів за модульний контроль
- по 30 балів за доповідь про розподілений проект та вирішення завдань по ньому

Критерії оцінювання індивідуальних завдань:

10 балів - Програма повністю реалізовує необхідний функціонал, осмислено спроектована з використанням архітектурних патернів, патернів проектування, принципів парадигми програмування, ідіом; імена в програмі є осмисленими, підтримується єдиний стиль кодування. Студент компетентно пояснює програму та може внести зміни.

8-9 балів - Програма не повно реалізовує необхідний функціонал. У архітектуру та код можна внести покращення. Студент компетентно пояснює програму та може внести зміни.

5-7 балів - Програма реалізовує більшість необхідного функціоналу. Архітектура та код можуть бути дещо хаотичними. Студент може пояснити програму, але з внесенням змін можуть бути труднощі.

2-4 бали - Програма реалізовує найпростіший юзкейс. Студент може пояснити програму, але з внесенням змін можуть бути труднощі.

1 бал - Програма наявна, студент може її запустити та продемонструвати, проте пояснення її роботи є незадовільним.

0 балів - програми немає або вона не запускається.

Критерії оцінювання доповіді:

15 балів - Тема висвітлюється повністю та цілісно, логічно та структуровано, враховуючи пов'язаний з нею контекст, причинно-наслідковід зв'язки. Наводяться приклади, покрокові інструкції. Впевнено відповідає на всі запитання, які пов'язані з тематикою доповіді та демонструє глибокі знання.

12-14 балів - Тема висвітлюється майже повністю та з незначними неточностями. Відповідає на більшість запитань, які пов'язані з тематикою доповіді.

5-11 балів - Тема висвітлюється не повністю та помилками, проте їх меншість та студент торкнувся основних понять.

1-4 бали - Студент володіє лише найпростішими поняттями з даної теми.

0 балів - Студент не підготував доповіді або тему доповіді не розкрито і під час захисту студент не може відповісти на жодне запитання за

	<p>тематикою доповіді.</p> <p>Критерії оцінювання фінального проекту: 15 балів - Вирішено складну проектну задачу або декілька простіших та наявний позитивний відгук спільноти. Студент компетентно пояснює суть змін. 12-14 балів - Вирішено проектну задачу. Студент компетентно пояснює суть змін. 5-11 балів - Проектна задача не вирішена, проте декілька проектних задач проаналізовано та проведено їх відлагодження. 1-4 бали - Проектна задача не вирішена, проте декілька проектних задач проаналізовано. 0 балів - Не має результатів по вирішенню проектних задач.</p> <p>Критерії оцінювання модульного контролю (15 завдань по 2б.): 2 бали: повністю вірна та повна відповідь; 1 бал: не повністю повна відповідь або відповідь з несуттєвими неточностями, проте зрозуміла суть; 0 балів: відповідь на питання неправильна.</p> <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; несвочасне 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.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

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	Тема 1. Вступ Огляд проектів з відкритим кодом, GNU, Linux.	Лекція (2 год.)	[1],[6-9]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 1. Вступ Пошук завдань для новачків.	Лабораторна робота (2 год.)	http://opensource.guide/how-to-contribute/#find-a-project-to-contribute-to	Виконання індивідуального проекту (6 год.)	1 тиждень
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.)		Опрацювання лекційного матеріалу (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 (3 год.)	1 тиждень
	Тема 2-3. SCM система Git: основи роботи та робота з гілками Паралельне внесення змін у код проекту у різних гілках та об'єднання цих гілок. Індивідуальне завдання	Лабораторна робота (2 год.)	[2]	Виконання індивідуального завдання (6 год.)	1 тиждень

	№1: Робота з SCM системою Git.				
4	Тема 4. Хороші практики SCM Рекомендації щодо повідомлень комітів, переписування історії.	Лекція (2 год.)		Опрацювання лекційного матеріалу (3 год.)	1 тиждень
5	Тема 5. Керування історією (серіями патчів) SCM Об'єднання, видалення, модифікація комітів.	Лекція (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
	Тема 4-5. Хороші практики SCM та керування історією Модульний контроль по Git.	Лабораторна робота (2 год.)	[2]	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
6	Тема 6. Розповсюджувачі ПЗ з відкритим кодом Портали, що розповсюджують ПЗ з відкритим кодом, проекти з трендів. Дистрибутиви Лінукс.	Лекція (2 год.)	SourceForge.net, GitHub, GitLab, opencores.org. https://distrowatch.com https://gitlab.com/Distroware/distroware.gitlab.io https://github.com/FabioLolix/LinuxTimeline	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
7	Тема 6. Розповсюджувачі ПЗ з відкритим кодом Проекти з трендів дистрибутивів Лінукс, критичні для релізу вади дистрибутивів.	Лекція (2 год.)	Debian popcorn https://bugs.debian.org/release-critical/	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
	Тема 2-5. Робота з SCM системою. Здача індивідуального завдання №1. Тема 6. Розповсюджувачі ПЗ з відкритим кодом	Лабораторна робота (2 год.)		Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень

	Індивідуальне завдання №2: Встановлення дистрибутиву Лінукс та робота з пакетами.				
6	Тема 7. Десктоп середовища Лінукс Огляд десктоп середовищ Лінукс та фреймворків і бібліотек на основі яких вони базуються.	Лекція (2 год.)	https://wiki.debian.org/DesktopEnvironment	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
9	Тема 8. Розповсюдження аплікацій на основі контейнеризації Snapcraft, Flatpak, AppImage	Лекція (2 год.)	https://snapcraft.io https://flatpak.org https://appimage.org	Опрацювання лекційного матеріалу, виконання індивідуального завдання (3 год.)	1 тиждень
	Тема 6-7. Дистрибутиви Лінукс. Здача індивідуального завдання №2. Індивідуальне завдання №3: Створення пакунків ПЗ: дистрибутивів та контейнеризованих пакетів.	Лабораторна робота (2 год.)	Документація відповідного дистрибутиву.	Опрацювання лекційного матеріалу, виконання індивідуального завдання (6 год.)	1 тиждень
10	Тема 9. Фундації проєктів з відкритим кодом GNU, Linux Foundation, Cloud Native Foundation, Eclipse Foundation, NumFOCUS, GNOME, KDE, freedesktop, Apache Software Foundation і т.д.	Лекція (2 год.)	Вебсайт відповідної фундації.	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
11	Тема 10. Проєкти з відкритим кодом FAANG компаній	Лекція (2 год.)	[3]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Теми 6-7. Дистрибутиви Лінукс. Здача індивідуального завдання №3.	Лабораторна робота (2 год.)	[3]	Виконання індивідуального проєкту (6 год.)	1 тиждень
12	Теми 6-10. Проєкт з	Лекція (2 год.)		Опрацювання лекційного	1 тиждень

	відкритим кодом Доповіді студентів про фінальний проект.			матеріалу (3 год.)	
13	Теми 6-10. Проект з відкритим кодом Доповіді студентів про фінальний проект.	Лекція (2 год.)		Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Теми 6-10. Проект з відкритим кодом Задача фінального проекту: вирішення вад.	Лабораторна робота (2 год.)		Виконання індивідуального проекту (6 год.)	1 тиждень
14	Тема 11. Спільнота, документація, правові аспекти README файл, INSTALL файл, інструкція контрибуції, мови розмітки документації, ліцензії програмного забезпечення, DCO, CLA.	Лекція (2 год.)	[3-9]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
15	Підсумкова лекція Огляд альтернативних проектів з відкритим кодом. Консультації щодо виконання індивідуальних проектів.	Лекція (2 год.)	https://github.com/explore , Debian	Виконання індивідуального проекту (6 год.)	1 тиждень
	Захист індивідуальних проектів.	Лабораторна робота (2 год.)		Повторення, погашення технічного боргу (3 год.)	1 тиждень
16	Розгляд подальших векторів розвитку.	Лекція (2 год.)			