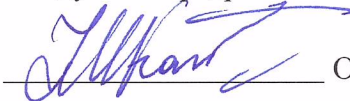


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Львівський національний університет імені Івана Франка**  
**Факультет електроніки та комп'ютерних технологій**  
**Кафедра оптоелектроніки та інформаційних технологій**

**Затверджено**

На засіданні  
кафедри оптоелектроніки та  
інформаційних технологій  
факультету електроніки та комп'ютерних  
технологій  
Львівського національного університету  
імені Івана Франка  
(протокол № 9 від 29 viii 2022 р.)

Завідувач кафедри:

  
Олег КУШНІР

**Силабус з навчальної дисципліни**  
**“ Життєвий цикл ПЗ”,**  
**що викладається в межах ОПП**  
**“ Інженерія програмного забезпечення ”**  
**першого (бакалаврського) рівня вищої освіти для здобувачів з**  
**спеціальності 121 – Інженерія програмного забезпечення**

Львів 2022 р.

<b>Назва дисципліни</b>	Життєвий цикл ПЗ
<b>Адреса викладання дисципліни</b>	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет електроніки та комп'ютерних технологій Кафедра оптоелектроніки та інформаційних технологій
<b>Галузь знань, шифр та назва спеціальності</b>	12 – Інформаційні технології 121 – Інженерія програмного забезпечення
<b>Викладачі дисципліни</b>	Катеринчук І.М., канд. фіз.-мат. наук, доцент Франів В.А., канд. фіз.-мат. наук, доцент
<b>Контактна інформація викладачів</b>	ivan.katerynychuk@lnu.edu.ua  <a href="mailto:volodymyr.franiv@lnu.edu.ua">volodymyr.franiv@lnu.edu.ua</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації в день проведення лекцій/лабораторних занять (за попередньою домовленістю): ауд. 214, корпус факультету електроніки та комп'ютерних технологій, вул. Ген. Тарнавського 107, м. Львів, 79011
<b>Сторінка курсу</b>	
<b>Коротка анотація дисципліни</b>	Дисципліна “Життєвий цикл ПЗ” є нормативною дисципліною з спеціальності 121 – Інженерія програмного забезпечення для освітньої програми “Інформаційні системи та технології”, яка викладається в 8-му семестрі в обсязі 4-ох кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
<b>Інформація про дисципліну</b>	Курс розроблено для ознайомлення студентів з основними поняттями та методами які використовуються у життєвому циклі ПЗ
<b>Мета та цілі дисципліни</b>	<i>Мета:</i> надати студентам основні поняття про життєвий цикл програмного забезпечення, яке використовуються в сучасних інформаційних технологіях <i>Цілі:</i> забезпечити знайомство студента з загальною теорією життєвого циклу програмного забезпечення і сформувати навички його практичного використання; навчити студента вибирати і обґрунтувати підхід до розробки програмного забезпечення та його підтримки.
<b>Література для вивчення дисципліни</b>	1) Життєвий цикл програмного забезпечення. Навчальний посібник / Є. В. Левус, Т. А. Марусенкова, О. О. Нитребич. Львів : Видавництво Львівської політехніки, 2017. 208 с.  2) Навчальний посібник з дисципліни «Технології розробки програмного забезпечення» для студентів спеціальності 123 «Комп'ютерна інженерія». – Полтава: ПолтНТУ, 2017. – 218 с.

<b>Обсяг курсу</b>	Загальний обсяг: 120 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 56 год.
<b>Очікувані результати навчання</b>	<p>У результаті вивчення даного курсу студент буде:</p> <p><b>знати:</b> основні поняття, визначення і проблеми курсу; вимоги до постановки основних задач життєвого циклу програмного забезпечення; призначення й особливості застосування різних життєвих циклів програмного забезпечення;</p> <p><b>вміти:</b> володіти основними інструментами та підходами для створення життєвого циклу ПЗ; реалізувати життєвий цикл ПЗ для тестового програмного забезпечення.</p> <p>Після вивчення даного курсу «Життєвий цикл ПЗ» здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.</p> <p>ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>ФК21. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи сучасні парадигми програмування.</p> <p>ПРН2. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.</p> <p>ПРН3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН4. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПРН6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПРН7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН10. Проводити передпроектне обстеження предметної області,</p>

	<p>системний аналіз об'єкта проектування.</p> <p>ПРН11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПРН22. Знати та вміти застосовувати методи та засоби управління проектами.</p> <p>ПРН24. Вміти проводити розрахунок економічної ефективності програмних систем.</p>		
<b>Ключові слова</b>	Програмне забезпечення, життєвий цикл, розробка.		
<b>Формат курсу</b>	Очний		
<b>Теми</b>	<p>Тема 1. Вихідні вимоги до програмного забезпечення.</p> <p>Тема 2. Підходи до розробки програмного забезпечення.</p> <p>Тема 3. Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ</p> <p>Тема 4. Аналіз прикладу життєвого циклу ПЗ при розробці ПЗ.</p> <p>Тема 5. Інтеграція. Розгортання. Постачання (CICD)</p> <p>Тема 6. Аналіз сучасних інструментів для впровадження CICD</p> <p>Тема 7. Версії програмного забезпечення.</p> <p>Тема 8. Видалення програмного забезпечення. Розробка документація</p>		
<b>Підсумковий контроль, форма</b>	Екзамен у кінці семестру		
<b>Пререквізити</b>	Для вивчення даного курсу студентам потрібні базові знання з курсів: - об'єктно-орієнтоване програмування.		
<b>Навчальні методи та техніки, які будуть використовуватися під час викладання курсу</b>	Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи (дискусія).		
<b>Необхідне обладнання</b>	Комп'ютер із необхідним програмним забезпеченням, доступ до Internet мережі.		
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> <li>• Змістовий модуль 1: 25% семестрової оцінки за виконані лабораторні роботи.</li> <li>• Змістовий модуль 2: 25% семестрової оцінки за виконані лабораторні роботи.</li> <li>• екзамен: 50% семестрової оцінки, максимальна кількість балів 50.</li> </ul> <p>Підсумкова максимальна кількість балів 100.</p>		
<b>Поточне тестування та самостійна робота</b>			
<b>Змістовий модуль 1</b>	<b>Змістовий модуль 1</b>		
	<table border="1"> <tr> <td style="text-align: center;">Піс ум- ков ий тест (екз)</td> <td style="text-align: center;">Сум а</td> </tr> </table>	Піс ум- ков ий тест (екз)	Сум а
Піс ум- ков ий тест (екз)	Сум а		

								аме н)	
T1	T2	T3	T4	T5	T6	T7	T8		
6	10	6	6	6	6	6	6	50	100
<p><b>Академічна доброчесність:</b> Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p><b>Відвідання занять</b> є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.</p> <p><b>Література.</b> 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><b>Політика виставлення балів.</b> Враховуються бали, набрані при поточному контролі та бали підсумкового тестування. 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.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>									
<p><b>Питання до заліку чи екзамену.</b></p> <p>Вихідні вимоги до програмного забезпечення?</p> <p>Основні принципи розробки програмного забезпечення?</p> <p>Agile?</p> <p>Порівняйте Scrum та Kanban.</p> <p>Які основні ролі існують у методології Scrum?</p> <p>Які основні ролі існують у методології Kanban?</p> <p>Наведіть приклад завдання для розробника.</p> <p>Наведіть приклад Bug репорту.</p> <p>Для чого використовуються розділи та з чого вони складаються.</p> <p>Порівняйте роль тестувальника та розробника у Scrum?</p> <p>Порівняйте роль тестувальника та розробника у Kanban?</p> <p>Які основні обов'язки обов'язки Scrum майстера?</p> <p>Опишіть обов'язки власника продукту.</p> <p>Як відбувається версіонування коду?</p> <p>Як відбувається процедура вирішення конфліктів у програмному кодї?</p> <p>Опишіть процес постійної інтеграції продукту.</p> <p>Опишіть процес постійної розробки коду.</p> <p>Опишіть процес постійної доставки коду.</p> <p>Які сучасні інструменти використовуються для CICD, їхні переваги та недоліки.</p> <p>Основні складові конвеєру по доставці програмного забезпечення?</p> <p>Конвеєр у Jenkins.</p> <p>Конвеєр у Kanban.</p> <p>Версіонування програмного забезпечення?</p>									

	Документація при розробці програмного забезпечення? Як відбувається видалення програмного забезпечення? Які наради відбуваються у Scrum? Основна ціль денної наради у Scrum?
<b>Опитування</b>	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

**Схема курсу “Методи обчислень”  
для студентів спеціальності 121 – Інженерія програмного забезпечення**

<b>Тиж.</b>	<b>Тема, план, короткі тези</b>	<b>Форма діяльності (заняття)* *лекція, самостійна, дискусія, групова робота)</b>	<b>Література.** * Ресурси в інтернеті</b>	<b>Завдання, год</b>	<b>Термін виконання</b>
1	Вихідні вимоги до програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
1	Лаб.1.Створення поверхневої архітектури програмного забезпечення, аналіз затребуваних ресурсів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
2	Вихідні вимоги до програмного забезпечення	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
2	Лаб.1 Створення розділів та завдань	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
3	Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
3	Лаб.2. Створення багів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
4	Вивчення обов'язків ролей у сучасних підходах до розробки ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
4	Лаб.2. Створення багів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
5	Створення багів	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
5	Лаб.3. Версіонування програмного коду	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
6	Аналіз прикладу життєвого циклу ПЗ при розробці ПЗ	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
6	Підсумкове заняття ЗМ 1	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
7	Версіонування програмного коду	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
7	Лаб.4. Вирішення конфліктів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
8	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
8	Лаб.4 Вирішення конфліктів документація	Лабораторна робота	Сайт курсу	2	кінець поточного тижня

9	Версії програмного забезпечення.	Лекція	[1], [2] Сайт курсу	2	Кінець поточного тижня
9	Лаб.5 Побудова CI/CD за допомогою Jenkins	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
10	Інтеграція. Розгортання. Постачання (CI/CD).	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
10	Лаб.5 Побудова CI/CD за допомогою Jenkins	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
11	Інтеграція. Розгортання. Постачання (CI/CD).	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
11	Лаб.5 Побудова CI/CD за допомогою Jenkins	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
12	Аналіз сучасних інструментів для впровадження CI/CD	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
12	Лаб.6 Побудова CI/CD за допомогою GitLab CI/CD	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
13	Аналіз сучасних інструментів для впровадження CI/CD	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
13	Лаб.6 Побудова CI/CD за допомогою GitLab CI/CD	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
14	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
14	Лаб.6 Побудова CI/CD за допомогою GitLab CI/CD	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
15	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Сайт курсу	2	кінець поточного тижня
15	Лаб.7 Розробка документації	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
16	Видалення програмного забезпечення. Розробка документація	Лекція	[1], [2] Лаб.6 Сайт курсу	2	кінець поточного тижня
16	Підсумкове заняття ЗМ 2	Лабораторна робота	Сайт курсу	2	кінець поточного тижня