

Чернівецький національний університет імені Юрія Федьковича

(повне найменування закладу вищої освіти)

Факультет математики та інформатики

(назва інституту/факультету)

Кафедра математичного моделювання

(назва кафедри)



“ЗАТВЕРДЖУЮ”

Декан

Мартинюк О.В.

2024 року

РОБОЧА ПРОГРАМА

навчальної дисципліни

Проектування програмних систем

(вказіть назву навчальної дисципліни (іноземною, якщо дисципліна викладається іноземною мовою))

обов'язкова

(вказати: обов'язкова)

Освітньо-професійна програма "Інформаційні технології та управління проектами"

(назва програми)

Спеціальність 122 – Комп'ютерні науки

(вказати: код, назва)

Галузь знань 12 - Інформаційні технології

(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)

(вказати: перший (бакалаврський)/другий (магістерський)/третій (освітньо-науковий))

факультет математики та інформатики

(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська

(вказати: на яких мовах читається дисципліна)

Чернівці 2024 рік

Робоча програма навчальної дисципліни «**Проектування програмних систем**» складена відповідно до освітньо-професійної програми «Інформаційні технології та управління проектами», затвердженої Вченою радою Чернівецького національного університету імені Юрія Федьковича (протокол № 7 від 29 квітня 2024 р.)

(назва освітньо-професійної програми, дата останнього затвердження)

Розробники: Піддубна Л.А., доцент, кандидат фізико-математичних наук
Перцов А.С., доцент, кандидат фізико-математичних наук
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Погоджено з гарантом ОП і затверджено на засіданні кафедри математичного моделювання

Протокол № 18 від “ 25 ” червня 2024 року

Завідувач кафедри

(підпис)

Черевко І.М.

(прізвище та ініціали)

Схвалено методичною радою факультету математики та інформатики

Протокол № 11 від “ 25 ” червня 2024 року

Голова методичної ради факультету

математики та інформатики

(підпис)

Сікора В.С.

(прізвище та ініціали)

1. Мета навчальної дисципліни Дисципліна “Проектування програмних систем” спрямована на узагальнене ознайомлення із інструментами, методиками, технологіями, системами розробки програмного забезпечення, вивчення систем управління проектними розробками цільового програмного забезпечення.

Знання і досвід, набуті в цій дисципліні, будуть корисними в майбутній практичній діяльності студентів при проектуванні чи розробці програм чи систем із залученням учнів/студентів навчальних закладів, де вивчається інформатика.

Метою дисципліни є опанування студентами теоретичних знань архітектури системного програмного забезпечення, побудови, функціонування, використання засобів операційних систем, технології контейнерів для реалізації упаковки, розгортання та функціонування програмного забезпечення.

Предмет дисципліни - вивчення принципів побудови, архітектури, основних функцій, режимів роботи, засобів операційних систем (ОС).

Завдання. В результаті вивчення дисципліни у студентів повинні сформуватися наступні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК8. Здатність генерувати нові ідеї (креативність)

ЗК11. Здатність приймати обґрунтовані рішення.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

ФК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

ФК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

ФК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

ФК17. Здатність здійснювати управління проектами на всіх стадіях проектного циклу, приймати ефективні рішення щодо їх ресурсного та організаційного забезпечення.

2. Результати навчання:

Завдання дисципліни. Знання і досвід, набуті при вивченні цієї дисципліни, будуть корисними в майбутній практичній діяльності студентів при проектуванні та розробці програм, систем, поглибленому вивченню інформатики, ведення позакласних гурткових знань з інформатики.

Для досягнення мети поставлені такі основні завдання:

- вивчення понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем;
- вивчення призначення засобів об’єктного підходу до проектування програмного забезпечення інформаційних систем;
- вивчення стандартів програмних засобів;
- вивчення методів створення вимог під час розроблення програмних продуктів;

- вивчення методів та засобів тестування програмне забезпечення сучасними засобами тестування програмних продуктів;
- вивчення стандартів якості під час створення програмного забезпечення інформаційних систем;
- вивчення інструментальних засобів та їх практичне застосування під час проектування програмне забезпечення інформаційних систем.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими програмними результатами навчання:

ПРН5. Проектувати, розробляти та аналізувати алгоритми розв’язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

ПРН10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проєктну документацію (техніко-економічне обґрунтування, технічне завдання, складних систем, методів структурного аналізу систем, об’єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем. бізнес-план, угоду, договір, контракт).

ПРН18. Здатність ефективно аналізувати вимоги до ІТ проєктів; брати участь у проектуванні архітектури, розробленні детальних планів інтеграції, тестування та впровадження проєкту; визначати стратегічні та тактичні аспекти виконання ІТ проєктів, враховуючи сучасні методики організації виконання проєктів та принципи гнучкого розроблення програмного забезпечення; здатність планувати, контролювати та вдосконалювати процеси розроблення програмного забезпечення.

3. Опис навчальної дисципліни

3.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	3	5	4	120	15			30	75		Екзамен

3.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем навчальних занять	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		лекц.	прак./семін.	лаб.	інд.	с.р.		лекц.	прак./сем.	лаб.	інд.	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Теми навчальних занять (назва теми й основні питання)	Змістовий модуль 1. Програмне забезпечення. Життєвий цикл та етапи розробки											
Тема 1. Базові питання. Види програмного забезпечення. Моделювання проєкту з UML: діаграми статичні та динамічні, логічні та фізичні. Види проєктування: архітектурне (верхній рівень) та деталізоване проєктування (класів, атрибутів, операцій), проєктування інтерфейсу користувача	11	1				10						
Тема 2. Парадигми проєктування: функціональна декомпозиція згори вниз, архітектура, орієнтована на дані, об'єктно-орієнтований аналіз та проєктування, подієво-керована архітектура. Життєвий цикл і етапи розробки програмного забезпечення.	13	1		2		10						
Разом за ЗМ1	24	2		2		20						
Теми навчальних занять (назва теми й основні питання)	Змістовий модуль 2. Сучасні методології розробки програмного забезпечення											
Тема 3 Ідентифікація класів предметної	13	1		2		10						

області. UML-діаграми ієрархії класів: моделювання підсистем, класів та зв'язків між ними												
Тема 4. Проектування сценаріїв реалізації варіантів використання на основі UML-діаграм послідовностей та комунікації Роль архітектури. Стандартні архітектури: клієнт-серверна та n-рівнева архітектура, Model View Controller	11	1		2		10						
Тема 5. Архітектурні моделі та патерни проектування (Abstract Factory, Facade, Decorator, Flyweight, Visitor, Observer, Proxy, Strategy, Chain of Responsibility)	16	2		4		10						
Разом за ЗМ 2	42	4		8		30						
Теми навчальних занять (назва теми й основні питання)	Змістовий модуль 3. Реалізація програмного забезпечення											
Тема 6. Вимоги до оформлення коду: стиль, розбиття на структуровані одиниці, найменування змінних, класів, об'єктів тощо Засоби автоматичної генерації програмного коду на основі трансформацій UML-модель – код	9	2		4		3						

ООП-мовою, повторне використання коду ПЗ.												
Тема 7. Налагодження: Точки зупинки (Breakpoints), Спостереження за змінними (Variable Watch), Виведення на консоль (Console Output), Налагоджувач (Debugger), Аналізатори коду (Code Analyzers).	15	2		6		7						
Тема 8. Керування конфігурацією програмного забезпечення та контроль версій.	13	2		4		7						
Тема 9. Забезпечення якості: тестування, верифікація, валідація Призначення, спільне та відмінності процесів тестування, верифікації, валідації Види тестів: модульні, інтеграційні, регресійні, системні, валідаційні Тестування методами білої та чорної скрині Розробка через тестування (Test- driven development)	10	2		4		4						
Тема 10. Додаткові техніки верифікації та валідації: інспекція коду, перевірка на відповідність	7	1		2		4						

стандартам і вимогам, оцінювання зручності використання та користувацького досвіду, перевірка продуктивності та масштабованості												
Разом за ЗМЗ	54	9		20		25						
Усього годин	120	15		30		75						

3.3. Тематика лабораторних занять

№	Назва теми	Кількість годин
1	Розробка технічного завдання	4
2	Побудова UsesCases діаграм	6
3	Застосування UML- мови для проектування ПС	7
4	Аналіз та підбір патернів, рефакторинг. Система контролю версій	5
5	Контроль якості ПС	8

3.4. Самостійна робота студента

№	Назва теми	кількість балів
1	Проблеми розробки складних програмних систем	2
2	Еволюція моделей життєвого циклу програмного забезпечення	2
3	Стандарт зрілості компанії-розробника Capability–Maturity–Model	2
4	Case–засоби та нотації моделювання програмних систем	2
5	Популярні та непопулярні патерни програмування	2
6	Оцінка якості програмного забезпечення (з т.з. маркетингу)	2

4. Освітні технології, методи навчання і викладання навчальної дисципліни

За джерелом передачі та сприймання студентами навчальної інформації:

- словесні (лекція, пояснення, бесіда, інструктаж);
- наочні (ілюстрація, демонстрація);
- практичні (завдання, проєкти).

За логікою передачі та сприймання студентами навчальної інформації:

- інформаційно-рецептивний;
- репродуктивний;
- проблемний;
- частково-пошуковий (евристичний).

За стимулюванням навчально-пізнавальної діяльності:

- методи стимулювання пізнавальних потреб та інтересів;
- методи стимулювання обов'язку та відповідальності.

5. Критерії та засоби оцінювання результатів навчання з навчальної дисципліни

Критерієм підсумкового оцінювання.

Система оцінювання рівня навчальних досягнень ґрунтується на принципах ECTS і є накопичувальною. Оцінювання знань здійснюється за 100-бальною шкалою. Результати роботи впродовж навчального семестру оцінюються під час поточного та модульного контролю в діапазоні загалом від 0 до 60 балів, а результати підсумкового контролю (екзамену) – від 0 до 40 балів.

Протягом семестру студенти виконують 5 лабораторні роботи, кожна з яких є продовженням попередньої. У результаті кожен студент виконує індивідуальний проєкт. Лабораторні роботи оцінюються відповідно балами: 15, 15, 15, 15 за повне виконання завдань.

Виконання лабораторних передбачає самостійного опрацювання додаткових інформаційних джерел і домашнього доопрацювання над завданнями, розпочатими в аудиторії. У випадку неістотних помилок при виконанні завдань знімається 1-2 бали, а істотних, необґрунтованого застосування методів чи невиконання завдань – 3-5 балів. У разі проходження сертифікованих курсів на навчальних платформах і вчасного подання сертифікатів можливе зарахування сертифікату замість лабораторної роботи з відповідної теми. Додатково можна отримати до 10 балів за виконання додаткових завдань.

Підсумковим контролем з дисципліни є залік у вигляді тестування у системі moodle. Варіант тесту містить 40 питань з однією правильною відповіддю, кожне з яких оцінюється в 1 бал.

Підсумкова оцінка виставляється за результатами суми балів набраних на змістовних модулях під час семестру та підсумковому модулі (екзамену).

Шкала оцінювання: національна та ECTS

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
Відмінно	A (90-100)	відмінно
Добре	B (80-89)	дуже добре
	C (70-79)	добре
Задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
Незадовільно	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим повторним курсом

6. Засоби оцінювання

Засобами оцінювання та демонстрування результатів навчання можуть бути:

- підготовлені звіти виконаних лабораторних чи дослідницьких робіт;
- сертифікати з проходження курсів.

7. Форми поточного та підсумкового контролю

Формами поточного контролю є лабораторні роботи.

Формою підсумкового контролю є екзамен.

8. Рекомендована література

1. Фаулер Мартін. Шаблони корпоративних додатків// Електронна книга
2. Scott Chacon and Ben Straub Pro Git book, – Apress // <https://git-scm.com/book/en/v2>
3. Pro Git. Режим доступу: <https://git-scm.com/book/uk/v2> .
4. Керівництва GitHub. Режим доступу: <https://guides.github.com/>
5. Кніберг Х. Scrum та XP // Електронна книга
6. Рубін К. Основи Scrum. Практичне керівництво для гнучкої розробки ПЗ / Рубін Кеннет С. // Електронна книга
7. Мартін Р. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І. Бондар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2019. — 448 с.
8. Мартін Роберт. Чиста архітектура: Мистецтво розроблення програмного забезпечення / пер. з англ. І. Бондар-Терещенко. — Харків : Вид-во «Ранок» : Фабула, 2019. — 368 с

9. Інформаційні ресурси

1.
https://stud.com.ua/97384/informatika/oglyad_suchasnih_tehnologiy_rozrobki_programnogo_zabezpechennya_ponyattya
2. <https://techexpert.ua/it-services/rozrobka-prykladnogo-pz/>
3. <https://armedsoft.com/ua/services/rozrobka-programnogo-zabezpechennya-ta-it-rishen>