

Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

СИЛАБУС
навчальної дисципліни

Проектування програмних систем

вибіркова

(вказати: обов'язкова)

Освітньо-професійна програма **Технології програмування та комп'ютерне моделювання**
(назва програми)

Спеціальність **113 Прикладна математика**
(вказати: код, назва)

Галузь знань **11 Математика**
(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)

Розробники: Скутар Ігор Дмитрович, асистент, канд. фіз.-мат. наук
(вказати авторів (викладач (ів)), їхні посади, наукові ступені, вчені звання)

Профайл викладача (-ів) <https://amit.chnu.edu.ua/pro-kafedru/spivrobotnyky/skutar-igor-dmytrovych/>

Контактний тел. **+38(050)9699747**

E-mail: **i.skutar@chnu.edu.ua**

Сторінка курсу в Moodle **<https://moodle.chnu.edu.ua/course/view.php?id=3147>**

Консультації **Згідно з розкладом консультацій**

1. Анотація дисципліни (призначення навчальної дисципліни). Дисципліна «Проектування програмних систем» є вибірковою дисципліною зі спеціальності 113 «Прикладна математика» для освітньої програми «Технології програмування та комп'ютерне моделювання», яка викладається в VIII семестрі в обсязі 5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS). Дисципліна «Проектування програмних систем» є логічним продовженням вивчення спеціальних дисциплін, здобуті знання знадобляться в подальшій практичній діяльності майбутнім фахівцям галузі ІТ. Курс спрямований на формування у студентів сучасного наукового світогляду та системи спеціальних знань щодо технологій та процесів розробки програмного забезпечення.

2. Мета навчальної дисципліни: забезпечення студентів необхідними теоретичними знаннями і практичними навиками використання сучасних технологій та підходів у програмуванні для створення програмного забезпечення та роботи в команді розробників.

3. Пререквізити. Програмування. Об'єктно-орієнтоване програмування. Основи інтернет-технологій. Сучасні СУБД. Серверна мова PHP. Технології програмування на Java.

4. Результати навчання (формулювання результатів навчання у вигляді переліку загальних та фахових компетентностей, програмних результатів відповідно до ОПП):

У результаті вивчення дисципліни студенти повинні **знати** основні поняття та твердження з програмного матеріалу даного курсу; **вміти** використовувати вивчений матеріал при написанні лабораторних проектів, застосовувати теоретичні знання на практиці.

Знання, які студент отримає в результаті вивчення даної дисципліни, відіграватимуть важливу роль у процесі його професійного формування та зростання, а також вони є необхідною основою для вивчення деяких інших фахових дисциплін.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

- ЗК01. Здатність учитися і оволодівати сучасними знаннями;
- ЗК02. Здатність застосовувати знання у практичних ситуаціях;
- ЗК03. Здатність генерувати нові ідеї (креативність);
- ЗК05. Здатність проведення досліджень на відповідному рівні;
- ЗК08. Знання та розуміння предметної області та розуміння професійної діяльності;
- ЗК09. Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності);
- ЗК10. Навички у використанні інформаційних і комунікаційних технологій;
- ЗК12. Визначеність і наполегливість щодо поставлених завдань і взятих обов'язків;

фахові компетентності:

- ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію;
- ФК05. Здатність проектувати бази даних, інформаційні системи та ресурси;
- ФК06. Здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків;
- ФК07. Здатність експлуатувати та обслуговувати програмне забезпечення автоматизованих та інформаційних систем різного призначення;
- ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення;
- ФК10. Здатність створення документів встановленої звітності, використання нормативно-правових документів;
- ФК11. Здатність до організації роботи колективу виконавців, приймання доцільних та економічно обґрунтованих організаційних та управлінських рішень, забезпечення безпечних умов праці. вміти працювати в команді розробників з проектування та розробки комп'ютеризованих систем, адаптуватися до різних практичних умов для ефективної праці в команді;

ФК16. Здатність використовувати набуті знання з фундаментальних та професійно-орієнтованих дисциплін для розуміння й аналізу нових технологічних рішень в галузі інформаційних технологій та застосовувати їх для розв'язання прикладних задач;

ФК18. Здатність до використання сучасних Web-технологій, методів і засобів розробки та супроводу Web-додатків;

та отримуються наступні **програмні результати навчання:**

ПРН11. Уміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символічних алгоритмів;

ПРН13. Використовувати в практичній роботі спеціалізовані програмні продукти та програмні системи комп'ютерної математики;

ПРН14. Виявляти здатність до самонавчання та продовження професійного розвитку;

ПРН15. Уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу;

ПРН16. Демонструвати навички взаємодії з іншими людьми, уміння працювати в команді;

ПРН18. Ефективно спілкуватися з питань інформації, ідей, проблем та рішень зі спеціалістами та суспільством загалом.

5. Опис навчальної дисципліни

5.1. 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	6	4	120	30	0	0	30	60	0	залік

5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усь го	у тому числі					усь ого	у тому числі				
		л	п	ла б	ін д	с. р.		л	п	ла б	ін д	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	
Змістовий модуль 1. Програмування на мові Python												
Тема 1. Метології розробки ПЗ		2		2		4						
Тема 2. Вимоги до програмного забезпечення		4		4		8						
Тема 3. UML		4		4		8						
Тема 4. SOLID-принципи ООД		4		4		8						
Разом за змістовим модулем 1		14		14		28						
Змістовий модуль 2. Big Data: великі дані, безмежні можливості												
Тема 1. Системи керування версіями		4		4		8						
Тема 2. Рефакторинг		4		4		8						
Тема 3. Патерни проектування		6		6		12						
Тема 4. Антипатерни		2		2		4						
Разом за змістовим модулем 2		16		16		32						

		3 0		3 0		6 0								
--	--	--------	--	--------	--	--------	--	--	--	--	--	--	--	--

5.3. Зміст завдань для самостійної роботи

№ з/п	Назва теми
1	Метології розробки ПЗ
2	Вимоги до програмного забезпечення
3	UML
4	SOLID-принципи ООД
5	Системи керування версіями
6	Рефакторинг
7	Патерни проектування
8	Антипатерни

5.4. Тематика лабораторних робіт

№ з/п	Назва теми
1	Метології розробки ПЗ
2	Вимоги до програмного забезпечення
3	UML
4	SOLID-принципи ООД
5	Системи керування версіями
6	Рефакторинг
7	Патерни проектування
8	Антипатерни

6. Система контролю та оцінювання. Види та форми контролю

Формами поточного контролю є:

1. Контрольне тестування з кожного модуля.
2. Вибіркове опитування на лекційних заняттях.

Формами підсумкового контролю є залік.

Засоби оцінювання

Засобами оцінювання та демонстрування результатів навчання можуть бути:

- тестування теоретичних знань;
- лабораторні роботи;

Критерії оцінювання результатів навчання з навчальної дисципліни

Залік. Підсумкова оцінка навчального курсу складається із сум оцінок за модульні контролю (60 балів) та залікової оцінки (40 балів). Це загалом складатиме 100 балів. Залік за рішенням кафедри “Прикладної математики та інформаційних технологій” (протокол №1 від 29 серпня 2019р.), полягає у захисті студентом розробленого протягом семестру проекту. **Залікова оцінка (40 балів) складається з 2-х складових частин:** особистого внеску у проект і результату виконання проекту командою. Кожна із складових частин оцінюється у **20 балів**.

Оцінювання проекту (20 балів). За дрібні помилки, які не привели до принципових змін в цілісній роботі проекту, знімається 1-5 балів. За збої і помилки в роботі проекту знімається 6-10 балів. Часткова робота проекту, в якому реалізовані окремі завдання кожного студента, відсутня цілісна робота проекту, оцінюється не вище 10 балів.

Оцінювання особистого внеску (20 балів). За дрібні недоробки у виконанні поставлених перед студентом у проекті завдань, які не привели до помилок чи цілісної роботи проекту, знімається 1-5 балів. За невірно реалізовану функціональність поставлених задач, помилки в роботі окремих модулів, знімається 6-10 балів. За незавершені задачі чи помилки, допущенні студентом, що суттєво вплинули на роботу проекту студент може отримати не більше 10 балів.

Підсумкова оцінка. Підсумкова оцінка виставляється за загальною сумою набраних студентом балів згідно із наступною таблицею:

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
80-89	B	добре	
70-79	C		
60-69	D		
50-59	E	задовільно	
35-49	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов’язковим повторним вивченням дисципліни	не зараховано з обов’язковим повторним вивченням дисципліни

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота		Підсумковий тест (залік)	Сума
Змістовий модуль 1	Змістовий модуль 2	40	100
30	30		

7. Рекомендована література

7.1. Базова (основна)

1. Martin Fowler, Kent Beck, John Brant, William Opdyke, Don Roberts, Erich Gamma. Refactoring: Improving the Design of Existing Code. - Addison-Wesley Professional, 1999, - 431p.
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software — Addison-Wesley Professional, 1994. — 416p.

3. Ларман, Крэг. Применение UML и шаблонов проектирования. 2-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2004. — 624 с. : ил. — Парал., тит. Англ.
4. Мартін Роберт. Чистий код. Створення і рефакторинг за допомогою Agile - Фабула, 2019. — 448 с.

8. Інформаційні ресурси

1. Рефакторинг.Гуру [Електронний ресурс]. – Режим доступу: URL: <https://refactoring.guru/>
2. Дизайн-патерни — просто, як двері [Електронний ресурс]. – Режим доступу: URL: <http://designpatterns.andriybuday.com>
3. GIT: Book [Electronic Resource]. – Mode of access: URL : <https://git-scm.com/book/uk/v2>
4. Система керування версіями. Вікіпедія [Електронний ресурс]. – Режим доступу: URL: <https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F%D0%B2%D0%B5%D1%80%D1%81%D1%96%D1%8F%D0%BC%D0%B8>