

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

«ЗАТВЕРДЖУЮ»

Заступник декана

з навчально-виховної роботи

Наталія ТМЄНОВА

« 29 » *серпня* 2026 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«УНІФІКОВАНЕ ПРОГРАМУВАННЯ»**

для студентів

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Інженерія програмного забезпечення
вид дисципліни	вільного вибору студента

Форма навчання	денна
Навчальний рік	2026/2027
Семестр	6
Кількість кредитів ECTS	6
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладачі: **Юрчук І.А.**, кандидат фізико-математичних наук, доцент

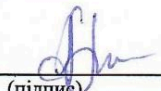
Пролонговано: на 20__/20__ н.р. _____ (підпис) _____ (ПІБ) «__»__ 20__р.

на 20__/20__ н.р. _____ (підпис) _____ (ПІБ) «__»__ 20__р.

КИЇВ – 2026

Розробник: **Юрчук Ірина Аркадіївна**, кандидат фізико-математичних наук,
доцент, доцент кафедри програмних систем і технологій

Завідувач кафедри
програмних систем і технологій

 проф. Олексій БИЧКОВ
(підпис)

Протокол № 18 від 17 червня 2026 року

Схвалено науково-методичною комісією факультету інформаційних
технологій

Протокол № 11 від 26 червня 2026 року

Голова науково-методичної комісії  Ганна КРАСОВСЬКА
(підпис)

© І.А. Юрчук, 2026 рік
© _____, 20__ рік
© _____, 20__ рік

1. Мета освітньої компоненти - засвоєння студентами базових знань щодо основних понять у галузі алгебри програм, тверджень та їх доведень, опанування аспектів денотаційної, операційної та алгебраїчної семантик, створення сценаріїв, які надають можливість стандартизувати основний та інфраструктурний синтаксиси, а також підходи мультипарадигмального програмування.

2. Попередні вимоги до опанування або вибору освітньої компоненти:

Для успішного вивчення дисципліни «Уніфіковане програмування» студенти повинні:

1. Знати:

1. Основні концепції процедурного та об'єктно-орієнтованого програмування;
2. Основні підходи та базові шаблони проєктування;
3. Основні структури даних;
4. Веб-технології, їх інструменти та середовища розробки.

2. Вміти:

1. Безпечно працювати з програмним кодом у системах контролю версій;
2. Створювати програмний код дотримуючись парадигм програмування та підходів до проєктування.

3. Володіти:

1. Базовими навичками застосування поширених інтегрованих середовищ розробки програмного забезпечення;
2. Базовими навичками застосування систем контролю версій, зокрема Git.

3. Анотація освітньої компоненти:

Освітня компонента формує у фахівців з Інженерії програмного забезпечення навички у доведенні чи спростуванні тверджень алгебри програм, створенні сценаріїв стандартизації денотаційної, операційної та алгебраїчної семантик, інфраструктурного синтаксису, а також мультипарадигмального програмування.

У результаті вивчення освітньої компоненти студенти повинні:

Знати:

1. основні поняття та твердження алгебри програм;
2. основні підходи до реалізації уніфікованої та мультипарадигмальної мов програмування;
3. основні підходи до уніфікації мов програмування високого рівня.

Вміти:

1. доводити основні твердження денотаційної, операційної та алгебраїчної семантик алгебри програм;
2. створювати сценарії, які надають можливість стандартизувати основний та інфраструктурний синтаксиси;
3. створювати сценарії мультипарадигмального програмування.

4. Завдання вивчення освітньої компоненти полягає у формуванні у студентів знань про основні підходи та методи алгебри програм та реалізації уніфікованої та мультипарадигмальної мов програмування. Дисципліна спрямована на досягнення компетентностей щодо здатності абстрактного мислення, застосовувати фундаментальні та дисциплінарні знання для успішного розв'язання завдання та обґрунтовано обирати інструментарій з розробки, дотримуючись законів алгебри програм.

5. Результати навчання за освітньою компонентою:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)	Форми (та / або методи і технології)	Методи оцінювання та пороговий	Відсоток у підсумковій оцінці з дисципліни
---	--------------------------------------	--------------------------------	--

Код	Результат навчання	викладання і навчання	критерій оцінювання ¹ (за необхідності)	
1.1	основні поняття алгебри програм;	лекція	виконання лабораторних робіт, STEM-проекту, іспит	3%
1.2	основні поняття та твердження денотаційної семантики;	лекція, лаборатор -не заняття		3%
1.3	основні поняття та твердження операційної семантики;	лекція, лаборатор -не заняття		3%
1.4	основні поняття та твердження алгебраїчної семантики;	лекція, лаборатор -не заняття		3%
1.5	основні аспекти створення сценаріїв стандартизації основного синтаксису;	лекція, лаборатор -не заняття		3%
1.6	основні аспекти створення сценаріїв стандартизації інфраструктурного синтаксису;	лекція, лаборатор -не заняття		3%
1.7	основні аспекти створення сценаріїв, які надають можливість об'єднання функціональної, імперативної та об'єктно-орієнтованої моделей;	лекція, лаборатор -не заняття		3%
1.8	основні аспекти створення сценаріїв уніфікації мов програмування високого рівня;	лекція, лаборатор -не заняття		3%
2.1	доводити основні твердження та теореми денотаційної, операційної та алгебраїчної семантик;	лаборатор -не заняття	виконання лабораторних робіт, STEM-проекту	до 36%
2.2	програмно реалізовувати сценарії стандартизації основного синтаксису;	лаборатор -не заняття		
2.3	програмно реалізовувати сценарії стандартизації інфраструктурного синтаксису;	лаборатор -не заняття		
2.4	програмно реалізовувати сценарії об'єднання імперативної та об'єктно-орієнтованої моделей;	лаборатор -не заняття		
2.5	програмно реалізовувати сценарії об'єднання функціональної та об'єктно-орієнтованої моделей;	лаборатор -не заняття		
2.6	програмно реалізовувати уніфікації мов програмування високого рівня;	лаборатор -не заняття		

¹ Критерії оцінювання див. пункт 7.

3	комунікація: здатність працювати в команді, вироблення у студентів практичних навиків проведення дослідження і оптимального групового вирішення актуальних проблем із застосуванням відповідних методів і прийомів.	лаборатор- не заняття, STEM-проєк т, аналітичне STEM-проєк т, досліджен ня, дискусія, вирішення конкретни х задач та ситуацій		до 20%
4	автономність: продемонструвати розуміння особистої відповідальності за професійні та/або управлінські рішення.		презентація , дискурс	до 20%

4. Співвідношення результатів навчання освітньої компоненти із програмними результатами навчання (необов'язково для вибіркових дисциплін, які не входять до блоків спеціалізації)

Результати навчання дисципліни (код)	1	1	1	1	1	1	1	1	2	2	2	2	2	2	3	4
Програмні результати навчання (назва)	1	2	3	4	5	6	7	8	1	2	3	4	5	6		
ПРН-25. Знати архітектуру програмного забезпечення та принципи проектування, володіти технологіями та методологіями розробки та реалізації програмного забезпечення та мережними технологіями, мати практичні навички з адміністрування комп'ютерних мереж в процесі розробки, розгортання та супроводження програмного забезпечення, розуміти і застосовувати програмне забезпечення мережевого обміну, обміну з сенсорами, віддаленої та розподіленої обробки інформації в інтересах інтернету речей.	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

ЗК 1, 2

ФК 1, 2, 4, 6, 8, 12, 13, 15, 17

7. Схема формування оцінки

7.1. Форми оцінювання студентів:

- семестрове оцінювання:

- **поточний контроль** шляхом виконання та захисту лабораторних робіт;
- **поточний контроль** шляхом виконання та захисту індивідуального STEM-проєкту;
- письмова **модульна контрольна робота**;

- підсумкове оцінювання (екзамен):

- **семестровий контроль** у формі екзамену.

Питома вага результатів навчання у підсумковій оцінці за умови її опанування на належному рівні така:

- ✓ результати навчання – **1.1 – 1.8 (знання)** – до 3% за кожен;
- ✓ результати навчання – **2 (вміння)** - до 36%;
- ✓ результати навчання – **3 (комунікація)** - до 20%
- ✓ результати навчання – **4 (автономність)** - до 20%

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності студентів оцінюються за 100-бальною шкалою. Робота в семестрі складається з одного змістовного модуля.

7.2. Організація оцінювання:

Обов'язковими для іспиту є виконання та захист студентом лабораторних робіт і індивідуального STEM-проєкту, що передбачені робочою програмою навчальної дисципліни, а також здача модульної контрольної роботи.

Оцінювання за формами контролю:

	Мін. – 36 балів	Макс. – 60 балів
Виконання та захист лабораторних робіт	12	20
Виконання та захист індивідуального STEM-проєкту	18	30
Модульна контрольна робота	6	10

Поточне семестрове оцінювання:

- оцінюванню підлягають модульна контрольна робота, звіти та захист лабораторних робіт та індивідуального STEM-проєкту;
- оцінюванню в індивідуальному STEM-проєкті підлягають програмна якість продукту, оформлена пояснювальна записка до нього та його захист відповідно до «Рекомендацій щодо оцінювання освітніх STEM-проєктів», затверджених на засіданні кафедри програмних систем і технологій, протокол № 16 від «01» травня 2026 р.;
- упродовж семестру орієнтовно на 11 тижні навчання проводиться письмова модульна контрольна робота із відкритими питаннями.

Оцінювання індивідуального STEM-проєкту:

<i>Критерії, що оцінюються</i>	Мін. – 18 балів	Макс. – 30 балів
--------------------------------	--------------------	---------------------

Програмна якість продукту: функціональна коректність, рівень проведеного тестування та якість коду	6	10
Захист індивідуального STEM-проєкту: відповіді на питання та демонстрація роботи продукту	6	10
Пояснювальна записка, яка містить опис призначення продукту, його архітектуру, функціонал, алгоритми роботи та інструкцію користувача у відповідності до стандартів	6	10

Для студентів, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум (36 балів), для допуску до іспиту обов'язковими є перездача модульних контрольних робіт, виконання та захист запланованих лабораторних робіт та індивідуального STEM-проєкту.

Студент має право захистити лабораторні роботи й індивідуальний STEM-проєкт і перескласти модульну контрольну роботу у терміни до та включно у день проведення консультації до іспиту.

Перескладання лабораторної роботи чи модульної контрольної роботи з метою підвищення оцінки не допускається.

Якщо студент не досяг мінімального межового рівня оцінки (60% від максимально можливої кількості балів, тобто 36 балів) разом із наданими спробами дозахистів лабораторних робіт та STEM-проєкту і перескладання модульної контрольної роботи, то студент не допускається до складання іспиту та отримує оцінку незадовільно (згідно шкали відповідності оцінок, див. п.7.2, оскільки сумарно набрані бали менше 60).

Підсумкове оцінювання (у формі іспиту):

- підсумковий контроль (іспит) проводиться у формі письмової роботи – 40 балів;
- екзаменаційний білет складається з двох частин: теоретична частина (2 завдання) та практична (2 завдання), які оцінюються 20 та 20 балів відповідно;
- максимальна кількість балів які можуть бути отримані студентом 40 балів на екзамені по 100-бальній шкалі;
- студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів;
- для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів.

7.3. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ЛАБОРАТОРНИХ ЗАНЯТЬ

№ п/п	Номер та назва теми	Кількість годин		
		Лекції	Лабора-торні	СР
Змістовний модуль 1. Уніфіковані теорії програмування				

1.	Тема 1. Вступ до уніфікованих теорій програмування. Денотаційна, операційна та алгебраїчна семантики	2		4
2.	Тема 2. Денотаційна семантика: рекурсія, передумови, постумови та предикатні трансформатори	6	6	14
3.	Тема 3. Алгебра програм: нормальна форма присвоєння, недетермінізм, обчислювальність та повнота	6	6	14
4.	Тема 4. Операційна семантика: коректність та бісимуляція. Послідовне та паралельне програмування	6	6	14
5.	Модульна контрольна робота		2	6
Змістовний модуль 2. Уніфікація процесу програмування				
6.	Тема 5. Уніфікована мова програмування (UPL) як стандартизація основного синтаксису	8	6	12
7.	Тема 6. Уніфікований синтаксис для інфраструктури. Фреймворки як засіб	6	4	12
8.	Тема 7. Мультипарадигмальне програмування як об'єднання функціональної, імперативної та об'єктно-орієнтованої моделей	8	6	12
9.	Тема 8. Уніфікація мов програмування високого рівня	8	4	12
	ВСЬОГО	40	40	100

Загальний обсяг 180 год., в тому числі:

Лекції – 40 год.;
Лабораторні заняття – 40 год.;
Самостійна робота – 100 год.

Теми лабораторних робіт

- 1-3. Теми 1-2. Програмна реалізація сценаріїв підтвердження та спростування тверджень денотаційної семантики.
4-6. Тема 3. Програмна реалізація сценаріїв підтвердження та спростування тверджень алгебраїчної семантики.
7-9. Тема 4. Програмна реалізація сценаріїв підтвердження та спростування тверджень операційної семантики.
10. **Модульна контрольна робота.**
11-12. Тема 5. Програмна реалізація сценаріїв стандартизації основного синтаксису.
13-14. Тема 6. Програмна реалізація сценаріїв стандартизації інфраструктурного синтаксису.
15-16. Тема 7. Програмна реалізація сценаріїв стандартизації об'єднання функціональної, імперативної та об'єктно-орієнтованої моделей.
17-18. Тема 8. Програмна реалізація сценаріїв уніфікації мов програмування високого рівня
19-20. Теми 1-8. Реалізація та захист індивідуального STEM-проекту.

Самостійна робота студента, її зміст та обсяг

№ п/п	Зміст самостійної роботи студента	Обсяг СРС (годин)
1	Підготовка до лекційних занять (теми 1-8)	30
2	Підготовка до лабораторних занять (згідно переліку)	24
3	Підготовка до виконання індивідуального STEM-проєкту	40
4	Підготовка до модульної контрольної роботи	6
Всього		100

РЕКОМЕНДОВАНА ЛІТЕРАТУРА:

Основна:

1. Рекомендації щодо оцінювання освітніх STEM-проєктів [Електронний ресурс]. – Режим доступу: <https://pst.knu.ua/academics/bachelor/evaluation-stem-projects/>
2. Baader F., Snyder W. Unification Theory. Handbook of Automated Reasoning. — Elsevier Science Publishers, 2001. - pp. 445–533.
3. Dowek G. Higher-order unification and matching. Handbook of automated reasoning. — Elsevier Science Publishers, 2001. – pp. 1009–1062.
4. Hoare C. A. R., Jifeng He. Unifying Theories of Programming. — Oxford University Computing Laboratory, 2014. — p. 53.
5. Woodcock J., Cavalcanti A. A tutorial introduction to designs in Unifying Theories of Programming. Integrated Formal Methods. — Lecture Notes in Computer Science, 2004. — pp. 40–66.

Додаткова:

1. Dunne S. Recasting Hoare and He's Unifying Theories of Programs in the Context of General Correctness. In A. Butterfield and C. Pahl, editors, IWFM'01: 5th Irish Workshop in Formal Methods, BCS Electronic Workshops in Computing, Dublin, Ireland, July 2001 The Internet & The Web, Theories & Literature Review, Preprint, June 2023, 7 p.
2. Hirasawa S. and Honda H. Unified Programming Environment for Heterogeneous Distributed Parallel Systems, 2008 International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems, Hilo, HI, USA, 2008, pp. 24-31.
3. Li X., Shi Z., Pan J. and Tian Y. A Unified Programming Model over Heterogeneous Data Planes, 2021 7th International Conference on Computer and Communications (ICCC), Chengdu, China, 2021, pp. 1331-1335.
4. Xiong Y. A Unified Programming Model For Heterogeneous Computing With CPU And Accelerator Technologies: An Axiomatic Approach, 2024 17th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 2024, pp. 1-5.
5. Zhou M., Quan Z. and Huang X. Unified program cross-architecture migration framework model, 2022 29th Asia-Pacific Software Engineering Conference (APSEC), Japan, 2022, pp. 427-431.

Додаткові ресурси:

Інформаційні ресурси

1. Carbon, a new programming language from Google, aims to be C++ successor, Bradshaw Kyle (19 July 2022). URL: <https://9to5google.com/2022/07/19/carbon-programming-language-google-cpp/>
2. Newman P. Unified Programming Language (8 January 2005). URL: <https://wiki.tcl-lang.org/page/Unified+Programming+Language>