

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 20__ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи побудови компіляторів

для студентів

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	бакалавр
освітня програма	Програмна інженерія
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	6
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: **к.ф.-м.н., доцент Ченцов О.І.** (лекції, лабораторні заняття)
асистент Супрун О.О. (лабораторні заняття)

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

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

Розробник: Ченцов Олексій Ілліч, к. ф.-м. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 20__ р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 202__ року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«_____» _____ 202__ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 202__ року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета дисципліни – опанування основ побудови компіляторів мов програмування. Вивчення устрою компіляторів, а також принципів лексичного, синтаксичного та семантичного аналізу.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. **Знати:** основи організації обчислювальної техніки, елементарні поняття з обраних розділів дискретної математики, зокрема, з теорії множин, теорії алгоритмів, графів, основи об'єктно-орієнтованого програмування на прикладі мов програмування C++ та/або Java.
2. **Вміти:** програмувати на мові програмування C/C++ або Java.
3. **Володіти:** елементарними навичками налагодження та тестування програмного забезпечення.

3. Анотація навчальної дисципліни. Навчальна дисципліна «Основи побудови компіляторів» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти у галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення в рамках освітньої програми «Програмна інженерія».

Дана дисципліна є обов'язковою навчальною дисципліною в рамках освітньої програми «Програмна інженерія». Викладається у 6 семестрі в **обсязі – 120 год. (4 кредити ECTS)** зокрема: лекції – 28 год., лабораторні – 28 год., самостійна робота – 62 год., консультації – 2 год. У курсі передбачено 2 змістові частини, одна контрольна робота. Завершується дисципліна – **іспитом**.

В результаті вивчення навчальної дисципліни студент повинен

знати: різновиди мовних процесорів та загальну структуру компілятора, підходи до лексичного, синтаксичного та семантичного аналізу та відповідні розділи теорії автоматів та формальних мов.

вміти: розробляти системи лексичного та синтаксичного аналізу для мов програмування різного ступеня абстракції, опрацьовувати вихідний код аналізуючої частини сучасних мовних процесорів.

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Основи побудови компіляторів» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області розробки компіляторів відповідно до освітньої кваліфікації бакалавр з інженерії програмного забезпечення. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність розробляти архітектури, модулі та компоненти програмних систем (СК03).
- Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (СК13).
- Здатність до алгоритмічного та логічного мислення (СК14).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати загальну структуру компілятора, підходи до лексичного, синтаксичного та семантичного аналізу та відповідні розділи теорії автоматів та формальних мов	Лекції	Контрольна робота, 52% балів, тест, іспит	63%
РН2.1	Вміти розробляти системи лексичного та синтаксичного аналізу для мов програмування різного ступеня абстракції	Лабораторні заняття, самостійна робота	Захист лабораторної роботи	15%
РН2.2	Вміти опрацювати вихідний код аналізуючої частини компілятора	Лабораторні заняття, самостійна робота	Захист лабораторної роботи	8%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм	Лабораторні заняття	Захист лабораторної роботи	4%
РН4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Контрольна робота, тест, захист лабораторної роботи, іспит	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Програмні результати навчання	Результати навчання дисципліни				
	РН1.1	РН2.1	РН2.2	РН3.1	РН4.1
ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	+		+		+
ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.		+		+	+
ПРН15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження	+	+		+	

програмного забезпечення.					
ПРН16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.				+	

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольні роботи: РН 1.1, РН 4.1 – 20 балів/12 балів.
2. Захист лабораторних робіт 1-3: РН2.1, РН2.2, РН 3.1, РН 4.1 – 40 балів/20 балів.
3. Оцінка якісних характеристик лабораторних робіт: 20 балів (оцінюється лектором).
Обов'язковим є виконання лабораторної роботи 1.

Для сприяння самостійній роботі студентів передбачені додаткові бали за отримання сертифіката з онлайн-курсу [Компілятори](#) – 10 балів або за раннє виконання наборів тестових завдань курсу – 3 бали за окремий набір тестових завдання, 10 за всі набори.

Підсумкове оцінювання:

- максимальна кількість балів, які можуть бути отримані студентом: 20 балів;
- результати навчання, які будуть оцінюватись: РН1.1, РН 4.1;
- форма проведення і види завдань: усно-письмова робота – письмова робота у вигляді google-форм, усний захист роботи, з можливістю додаткових питань.
- Види завдань: письмові завдання з відповіддю безпосередньо в полі форми або з можливістю надсилання фотокопії. На виконання кожного блоку роботи встановлені індивідуальні часові обмеження.

Студент може бути не допущеним до іспиту, якщо під час семестру набрав менше ніж 40 балів.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота: до 15 тижня семестру.
2. Лабораторна робота 1: до 6 тижня семестру.
3. Лабораторна робота 2: до 10 тижня семестру.
4. Лабораторна робота 3: до 15 тижня семестру.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

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.

№ лекції	Назва теми/лекції	Кількість годин		
		Лекції	Лабораторні заняття	Самост. робота
Частина 1. Лексичний аналіз.				
1	Тема 1. Загальні відомості про компілятори.	2	2	4
2	Тема 2. Призначення та устрій лексичного аналізатора.	2	2	4
3	Тема 3. Регулярні вирази та мови.	2	2	4
4	Тема 4. Скінченні автомати. Алгоритм Хопкрофта.	2	2	6
Всього по частині 1		8	8	18
Частина 2. Синтаксичний аналіз.				
5	Тема 5. Формальні мови та граматики.	2	2	4
6	Тема 6. Синтаксичний аналізатор	2	2	4
7	Тема 7. Низхідний синтаксичний розбір. Лекція 7. Метод рекурсивного спуску.	2	2	4
8	Тема 7. Низхідний синтаксичний розбір. Лекція 8. LL-граматики.	2	2	4
9	Тема 8. Автомати з магазинною пам'яттю.	2	2	4
10	Тема 9. Висхідний синтаксичний розбір. Лекція 10. Загальний устрій LR-аналізаторів.	2	2	4
11	Тема 9. Висхідний синтаксичний розбір. Лекція 11. Канонічний LR(1)- та LALR(1)-аналізатори	2	2	4
Всього по частині 2		14	14	28
Частина 3. Семантичний аналіз.				
12	Тема 10. Призначення та складові семантичного аналізу	2	2	4
13	Тема 11. Атрибутні граматики	2	2	6
14	Тема 12. Перевірка типів	2		6
Контрольна робота			2	
Всього по частині 3		6	6	16
ВСЬОГО		28	28	62

Загальний обсяг – 120 год., в тому числі:

Лекцій – 28 год.
Лабораторні роботи – 28 год.
Самостійна робота – 62 год.
Консультації – 2 год.

Теми, винесені на самостійне вивчення:

1. Концепції побудови ефективних мовних процесорів [д1,д3,д4].
2. Системи автоматичного породження лексичних аналізаторів [1].
3. Проблеми синтезу та аналізу в скінчених автоматах – повторення. Теорема Майхіля–Нероуда [д1,д2].
4. Проблеми синтаксичного аналізу. Шляхи підвищення ефективності аналізаторів [д1,д3].
5. Синтаксичний розбір на основі уасс (bison) [д1].
6. Універсальні розпізнавачі: СУК, алгоритм Ерлі[д3].
7. Синтаксичний розбір неоднозначних граматик[1].
8. Реалізація L-атрибутних синтаксично-керованих визначень[1].
9. Породження проміжного коду[1].

Умови лабораторних робіт:

- **Лабораторна робота 1:** Розробка лексичного аналізатора.
- **Лабораторна робота 2:** Розробка синтаксичного аналізатора.
- **Лабораторна робота 3:** Розробка інтерпретатора.

Деталізовані умови лабораторних робіт розміщено за посиланням:

- <https://drive.google.com/drive/folders/1hXWgK0VOw6-phwr0DhdiYWjbRC-sThyc>

9. Рекомендовані джерела:

Основні:

1. *Aho A. Compilers: Principles, Techniques, and Tools, 2nd ed.* / Aho A., Lem M., Sethi R., Ullman J. — Addison Wesley, 2007. — 1007 p.
2. *Hopcroft J. Introduction to Automata Theory, Languages, and Computation, 3rd ed.* / John Hopcroft, Rajeev Motwani, Jeffrey Ullman. — Prentice Hall, 2006. — 750 p.

Додаткові:

1. Методичні рекомендації до лабораторного практикуму побудови мовних процесорів з курсу «системне програмування» / В.В. Волохов, Б.І. Бойко, В.Ф. Кузенко, С.С. Шкільняк. – К.: Київський національний університет імені Тараса Шевченка, 2001. – 52 с.
2. *Ахо А. Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т1. М. Мир. 1978.
3. *Льюис Ф., Стирнз Р., Розенкранц Д.* Теоретические основы построения компиляторов. — М. Мир, 1979.
4. *Прайт Т.* Языки программирования: разработка и реализация. — М.: Мир, 1979.
5. *Reinhard W., Seidl H., Hack S.* Compiler Design: Syntactic and Semantic Analysis. — Springer, 2013 — 225p.

10. Додаткові ресурси

Приклади контрольних завдань, перелік контрольних тем, що виносяться на іспит, розміщено за посиланнями:

- <https://drive.google.com/drive/folders/1dpQlolCCA7WhXsc1gHCMKOgXWLacbOvK>
- <https://drive.google.com/drive/folders/1F7Erzvj0ovcFW2HCTaQsrrzHTQQk43cU>