



ПРЕДМЕТНО-ОРІЄНТОВАНІ МОВИ ПРОГРАМУВАННЯ: РОЗРОБКА ТА РЕАЛІЗАЦІЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	очна (денна)
Рік підготовки, семестр	3 курс осінній семестр
Обсяг дисципліни	4,0 кредитів (120 годин) 36 лек. 18 лаб. 66 СРС
Семестровий контроль/ контрольні заходи	Залік, мкр
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент, Стативка Юрій Іванович, statyvka-yu@iit.kpi.ua Лабораторні: к.т.н., доцент, Стативка Юрій Іванович, statyvka-yu@iit.kpi.ua
Розміщення курсу	https://classroom.google.com/c/NDk4NTE2NzYwOTUw https://ecampus.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

В курсі розглядаються як фундаментальні принципи створення і імплементації предметно-орієнтованих мов програмування, так і практичні аспекти автоматизації розробки трансляторів з використанням новітніх технологій та інструментів.

Значна увага приділяється проектуванню мов програмування, їх специфікації, математичним основам представлення та опрацювання формальних мов та їх імплементації.

Розглядаються загальні та спеціалізовані засоби побудови компіляторів та інтерпретаторів: алгоритми лексичного, синтаксичного та семантичного аналізу, генерування коду (низького, проміжного чи високого рівня), засобів виконання коду цільової мови, такі засоби автоматизації, як Yacc, ANTLR, BNFC тощо.

Метою дисципліни є опанування студентами теоретичних знань та набуття практичного досвіду проектування і імплементації предметно-орієнтованих мов програмування .

Предмет дисципліни — методи та алгоритми побудови, специфікації та імплементації предметно-орієнтованих мов програмування.

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

загальні:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 1);
- здатність застосовувати знання у практичних ситуаціях (ЗК 2);
- здатність спілкуватися державною мовою як усно, так і письмово (ЗК 4);

фахові:

- Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем (ФК 3);
- здатність застосовувати теоретичні та практичні основи методології та технології моделювання для дослідження характеристик і поведінки складних об'єктів і систем, проводити обчислювальні експерименти з обробкою й аналізом результатів (ФК 7);
- 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 (ФК 8);
- 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 (ФК 10).

Після засвоєння навчальної дисципліни студенти мають продемонструвати такі програмні результати навчання:

- застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук (ПР 1);
- використовувати сучасний математичний апарат неперервного та дискретного аналізу, лінійної алгебри, аналітичної геометрії, в професійній діяльності для розв'язання задач теоретичного та прикладного характеру в процесі проектування та реалізації об'єктів інформатизації. (ПР 2);
- 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 (ПР 5);
- 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 (ПР 9);
- володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт) (ПР 11).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити дисципліни. Знання, отримані при вивченні дисциплін: «Алгоритмізація та програмування» «Об'єктно-орієнтоване програмування», «Дискретна математика», «Програмування алгоритмічних структур».

Постреквізити дисципліни. Отримані знання при вивченні дисципліни «Предметно-орієнтовані мови програмування: розробка та реалізація» можуть бути використані при вивченні таких дисциплін: «Моделювання систем», «Проектування інформаційних систем», які викладаються в наступних семестрах. Компетенції, отримані студентами в процесі вивчення дисципліни можуть бути використані при виконанні дипломної роботи.

3. Зміст навчальної дисципліни

Тема 1 Основи теорії формальних граматики.

Тема 2 Лексичний та синтаксичний аналіз.

Тема 3 Трансляція та виконання програм

Тема 4 Автоматизація побудови трансляторів

4. Навчальні матеріали та ресурси

Основна література

1. Медведєва В.М. Транслятори: лексичний та синтаксичний аналізатори [Текст] : навч. посіб. / В.М. Медведєва, В.А. Третьак. – К. : НТУУ «КПІ», 2012. – 148с.
2. Reghizzi S. C. Formal Languages and Compilation / S. C. Reghizzi, L. Breveglieri, A. Morzenti. – Cham: Springer Nature Switzerland AG, 2019. – 499 с. – (Third Edition). – (Texts in Computer Science).
3. Aho, Alfred, Lam, Monica, Sethi, Ravi, Ullman, Jeffrey Compilers: Principles, Techniques, and Tools, 2nd edition. - Addison Wesley, 2006. - 1040 p.
4. Mogensen T. Æ. Introduction to Compiler Design / Torben Ægidius Mogensen. – Cham: Springer International Publishing AG, 2017. – (Second Edition). – (Undergraduate Topics in Computer Science).

Додаткова література

5. Martin Fowler, Rebecca Parsons Domain Specific Languages. - Addison-Wesley Signature Series, 2010, 640p.
6. Хантер Р. Проектирование и конструирование компиляторов. – Пер. с англ. - М. : Финансы и статистика 1984, 217 с.
7. Ахо А. Ульман Д. Теория синтаксического анализа, перевода и компиляции. -М.: Мир, 1978.
8. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. Пер с англ. -М.: Мир, 1975.
9. Лебедев В.Н. Введение в системы программирования. -М.: Статистика, 1975.- 310с.
10. Методические указания к разделу “Разработка и реализация транслятора” курса “Лингвистическое обеспечение САПР”/ Сост. В.Н. Медведєва - Киев: КПИ, 1990.- 72с.

11. Методические указания к разделам “Языки САПР” и “Организация таблиц” курса “Лингвистическое обеспечение САПР”/ Сост. В.Н. Медведева - Киев: КПИ, 1990.- 40 с.
12. Методичні вказівки до лабораторних робіт з курсу «Лінгвістичне забезпечення» для студентів напряму «Комп’ютерні науки» / Укл.: В.М. Медведева, П.О. Петін.- К.: НТУУ «КПІ», 1997.- 72с.
13. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов. : М. : Мир 1988. – 192 с.
14. Хантер Р. Проектирование и конструирование компиляторов. – Пер. с англ. - М. : Финансы и статистика 1984, 217 с.
15. Рейерд-Смит В. Дж. Теория формальных языков. Вводный курс. Пер. с англ.-М.: Радио и связь, 1988.
16. Бек Л. Введение в системное программирование. Пер. с англ. -М.: Мир, 1988.
17. Льюис Ф., Розенкранц Д., Стирнз Р. Теоретические основы проектирования компиляторов : Пер с англ . -М.: Мир , 1979.

Інформаційні ресурси

18. <http://dinosaur.compilertools.net/>
19. <https://wwwantlr.org/>
20. <https://www.haskell.org/happy/>
21. <https://github.com/antlr/antlr4/blob/master/doc/getting-started.md>
22. <https://github.com/antlr/antlr4/blob/4.6/doc/index.md>

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Тема 1 Основи теорії формальних граматики

Лекція 1. Вступ. Предметно-орієнтовані мови програмування. Компіляція.

Предметно-орієнтовані мови програмування (DSL). Класифікація DSL. Компіляція. Основні блоки та структура компілятора.

Лекція 2. Елементи теорії формальних граматики.

Ланцюжки, операції над ними. Мова. Генеративна граматика. Класифікація граматики Хомського. Виведення. Сентенційна форма. Рекурсивні правила.

Лекція 3. Форми подання граматики.

Нотація Бекуса-Наура. Розширена нотація Бекуса-Наура. Синтаксичне дерево. Синтаксичні діаграми. Однозначність граматики. Варіанти граматики арифметичного виразу.

Тема 2 Лексичний та синтаксичний аналіз

Лекція 4. Функції лексичного аналізатора. Розбір до роздільника

Лексичний аналізатор. Призначення, вхід, вихід. Реалізація лексичного аналізатора методом розбору до роздільника.

Лекція 5. Реалізація лексичного аналізатора за допомогою діаграми станів.

Діаграма станів. Алгоритм лексичного аналізу за допомогою діаграми станів.

Лекції 6. Скінченні автомати

Визначення, правила функціонування, форми подання. Перетворення. Програмна реалізація скінченних автоматів.

Лекція 7. Призначення та види синтаксичного аналізу. Рекурсивний спуск

Призначення, види синтаксичного розбору. Низхідний розбір. Алгоритм рекурсивного спуску. Перетворення граматики для реалізації розбору без повернень та зациклювання.

Лекція 8. Автомат з магазинною пам'яттю

Визначення, правила функціонування, форми подання.

Лекція 9 Використання автомата з магазинною пам'яттю для синтаксичного розбору.

Приклади побудови фрагментів граматичних процесорів з використанням магазинного автомата.

Тема 3 Трансляція та виконання програм

Лекція 10. Проміжні форми подання програм.

Призначення проміжних форм, їх види. Польський інверсний запис. Алгоритм Дейкстри побудови ПОЛІЗ. Пріоритети.

Лекція 11. Переклад у ПОЛІЗ.

Переклад у ПОЛІЗ оператора присвоювання. Пріоритети.

Лекція 12. Виконання ПОЛІЗ-програм

Виконання ПОЛІЗ-програм з операторами присвоювання. Контроль типів.

Лекція 13. Переклад у ПОЛІЗ складених операторів.

Переклад у ПОЛІЗ операторів розгалуження, повторення тощо.

Лекція 14. Трансляція у систему команд JVM (Java Virtual Machine).

JVM: архітектура, система команд, виконання.

Лекція 15. Трансляція у CIL (Common Intermediate Language).

Мова CIL. Засоби .NET Framework для роботи з CIL. JIT-компілятор.

Тема 4 Автоматизація розробки трансляторів

Лекція 16. Засоби автоматизації розробки трансляторів

Засоби автоматизації розробки трансляторів: Yacc, Bison, Lex, Flex, Narry, ANTLR4.

Лекція 17. Генератор парсерів ANTLR4

Структура, встановлення, використання.

Лекція 18. Застосування ANTLR4 для розробки транслятора.

Побудова та обхід AST.

6. Самостійна робота студента

Тема 1 Основи теорії формальних граматики

Спеціалізовані транслятори, додаткові блоки аналізу. Виведення та вивідність. Сентенційна форма. Класифікація граматики. Інші способи подання граматики. Операції над ланцюжками, алфавітами, множинами ланцюжків. Граматика логічного виразу.

Література [1 С. 9-34], [2, 3] [4 С. 28-53]

Тема 2 Лексичний та синтаксичний аналіз

Побудова таблиці лексем спрощеної мови програмування. Скінченні автомати. Регулярні вирази.

Література [1, С. 50-53, 57-73].

Тема 3 Трансляція та виконання програм

Побудова тетраді при синтаксичному розборі “згори вниз”. Генерація четвірок при рекурсивному спуску.

Література [1, ст.91-96].

Тема 4 Автоматизація розробки трансляторів

Встановлення та перший сеанс роботи з ANTLR.

Засоби обходу AST.

Література [20-21].

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог при вивченні дисципліни:

1. Кодекс честі: <http://kpi.ua/code>;
2. Правила внутрішнього розпорядку: <http://kpi.ua/admin-rule>;
3. Положення про організацію освітнього процесу в КПІ ім. Ігоря Сікорського: <https://kpi.ua/regulations>.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Система рейтингових (вагових) балів та критерії оцінювання

1) Робота на лекціях

На лекціях може бути проведено бліцопитування студентів. Такі опитування проводяться на довільних лекціях протягом семестру, наприкінці лекції. Ваговий бал за правильну відповідь - 1. Максимальна кількість балів, що може отримати кожен студент за семестр — 5.

2) Лабораторні практикуми

Максимальна кількість балів за усі виконані комп'ютерні практикуми дорівнює 58 балів. Розподіл балів серед лабораторних практикумів наступний:

№ з/п	Назва лабораторного практикуму	Кількість балів
1	Специфікація мови програмування	10
2	Лексичний аналіз програм	20
3	Синтаксичний аналіз методом рекурсивного спуску	10
4	Трансляція у ПОЛІЗ та виконання програм	20

5	Автоматизація побудови компіляторів	15
Всього:		75

Критерії оцінювання:

Виконання лабораторних робіт комп'ютерного практикуму:

- виконана (і захищена) своєчасно у повному обсязі – максимальна кількість балів згідно номера роботи;
- виконана із запізненням – знімається 10 – 30% від максимальної кількості балів в залежності від терміну запізнення;
- виконана не самостійно – знімається 50% від максимальної кількості балів.

3) Творче завдання

Студент може обрати додаткове завдання творчого характеру, результати виконання якого можуть бути опубліковані у наукових виданнях, або повідомлені на студентській науковій конференції з публікацією тез. Максимальна кількість балів за виконання творчого завдання — 5 балів.

4) Модульна контрольна робота

Максимальна кількість балів за модульну контрольну роботу дорівнює 15 балів.

Якість виконання роботи:

- усі завдання виконані правильно та повно – 15 балів,
- у відповідях допущені несуттєві неточності – 10 балів,
- половина відповідей правильна – 7 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 3 бали,
- менше половини відповідей правильна – 0 балів.

Умови позитивної проміжної атестації

Для отримання „зараховано” з першої проміжної атестації студент матиме не менше ніж 10 балів.

Для отримання „зараховано” з другої проміжної атестації студент матиме не менше ніж 22 бали.

Умови допуску до залікової контрольної роботи

Необхідною умовою допуску до залікової контрольної роботи є зарахування усіх лабораторних робіт комп'ютерного практикуму та виконання модульної контрольної роботи, а також стартовий рейтинг (R_c) не менше 40 балів.

Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{лек}} + r_{\text{практ}} + r_{\text{тв}} + r_{\text{мод}} = 5 + 75 + 5 + 15 = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
R < 40 є незараховані роботи комп'ютерного практикуму або не виконані інші умови допуску до екзамену	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено доцент, к.т.н., доцент, Стативка Юрій Іванович

Ухвалено кафедрою ЦТЕ (протокол № 1 від 01.07.2022р)

Погоджено Методичною комісією факультету (протокол № 10 від 04.07.2022 р.)