

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
Кафедра теорії та технології програмування



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

СИСТЕМНЕ ПРОГРАМУВАННЯ

для студентів денної форми навчання

галузь знань **12 Інформаційні технології**
спеціальність **124 Системний аналіз**
освітній рівень **бакалавр**
освітня програма **Системний аналіз**
вид дисципліни **вибіркова**

Форма навчання	денна
Навчальний рік	2022/2023
Семестр	7
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: к.ф.-м.н., доц. Волохов В.М.

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

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

КИЇВ – 2020

Розробник: Волохов Віктор Миколайович, к.ф.-м.н., доцент кафедри «Теорії та технології програмування»

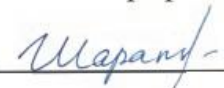
ЗАТВЕРДЖЕНО

Зав. кафедри теорії та технології програмування


_____ (підпис) (Нікітченко М.С.)
_____ (прізвище та ініціали)

Протокол №1 від «28» 08 2020 р.

Схвалено Гарантом освітньо-професійної програми першого рівня вищої освіти

«Системний аналіз»  М.М. Шарапов

«28» серпня 2020 року

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

Протокол від «28» серпня 2020 року №1

Голова науково-методичної комісії _____ (Омельчук Л.Л.)
(підпис) _____ (прізвище та ініціали)

«28» серпня 2020 року

ВСТУП

1. Мета дисципліни. Навчальна дисципліна «Системне програмування» є складовою освітньо-професійної програми підготовки фахівців за освітньо-кваліфікаційним рівнем «бакалавр» галузі знань «12 Інформаційні технології» зі спеціальності – «124 Системний аналіз» освітньо-професійної програми «Системний аналіз». Основна мета програми: освоїти загальні принципи побудови структури мовних процесорів: трансляторів та інтерпретаторів. Розглянути основні блоки мовного процесора. Визначити алгоритми конструювання основних блоків мовного процесора. Реалізувати на практиці запропоновані алгоритми та отримати практичні навички проектування та програмування блоків (модулів) мовного процесора.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* загальні принципи проектування алгоритмів обробки нечислової інформації; мови програмування C, C++ та Python на базовому рівні; технології та методи проектування та програмування.

2. *Вміти:* розробляти специфікації з урахуванням встановлених вимог; демонструвати процеси та результати професійної діяльності.

3. *Володіти елементарними навичками:* програмування мовою C, C++ та Python з використанням інструментальних середовищ розробки програмного забезпечення.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Системне програмування» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань «12 Інформаційні технології» зі спеціальності «124 Системний аналіз», освітньо-професійної програми – «Системний аналіз».

Предмет навчальної дисципліни «Системне програмування» включає в себе розгляд теоретичних аспектів проектування та створення мовних процесорів мов програмування, вивчення відповідних класів граматики, опанування алгоритмів та їх програмування з метою отримання практичних навичок реалізації мовних процесорів.

Викладається у 7 семестрі 4 курсу в **обсязі – 120 год (4 кредити ECTS)** зокрема: *лекції – 28 год., лабораторні – 14 год., самостійна робота – 76 год, консультації – 2 год.* У курсі передбачено 2 частини, 1 контрольна робота та 4 лабораторних роботи. Завершується дисципліна – **іспитом в 7 семестрі.**

4. У результаті вивчення навчальної дисципліни студент повинен:

знати: засоби проектування систем; стандарти моделювання, методи аналізу потреб; методи розробки програмного забезпечення; принципи проектування користувацьких інтерфейсів; інструментальні засоби мови програмування; передові технології; мови програмування C, C++, Java та Python;

вміти: розробляти та аналізувати граматики мов програмування розробляти, аналізувати та реалізувати алгоритми побудови компонентів мовного процесора, розробляти специфікації з урахуванням встановлених вимог; роз'яснювати і представляти проекти / розробки замовникам з використанням сучасних технологій розробки програмних систем.

Зв'язок з іншими дисциплінами. Дисципліна «Системне програмування» є базовою для засвоєння всіх інших курсів та спецкурсів програмиського спрямування, окремих розділів теорії алгоритмів та математичної логіки.

Завдання (навчальні цілі): набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно освітньої кваліфікації «Бакалавр з комп'ютерних наук».

1. Зокрема:

- здатність застосовувати знання у практичних ситуаціях;
- здатність оцінювати та забезпечувати якість виконуваних робіт;
- здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних;
- здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

7. Результати навчання за дисципліною:

Результат навчання (РН) (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			
РН 1.1	Знати та освоїти основні алгоритми розробки мовних процесорів	Лекція	Тест, 60% правильних відповідей, іспит	14%
РН 1.2	Знати принципи роботи Java-орієнтованого інструментального комплексу розробки програмного забезпечення.	Лекція, лабораторне заняття	Тест, 60% правильних відповідей, іспит	10%
РН 1.3	Знати та освоїти принципи функціонування JVM – середовища.	Лекція, лабораторне заняття,	Захист лабораторної роботи, іспит	10%
РН 2.1	Вміти працювати з Java-орієнтованим інструментальним комплексом розробки програмного забезпечення.	Лабораторне заняття, самостійна робота	Захист лабораторної роботи, іспит	20%
РН 2.2	Вміти працювати пакетом програм розробки компонент мовного процесора	Лабораторне заняття, самостійна робота	Захист лабораторної роботи	20%
РН 2.3	Вміти самостійно описувати синтаксис мови програмування з орієнтацією на відповідний клас граматик.	Лабораторні роботи, самостійна робота	Захист лабораторної роботи	20%
РН 4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Захист лабораторної роботи	6%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 2.2	РН 2.3	РН 4.1
Програмні результати навчання <i>(з опису освітньої програми)</i>							
ПР10. Знати архітектуру сучасних обчислювальних систем і комп'ютерних мереж.	+	+	+	+			
ПР13. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.					+	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота (тест): РН 1.1, РН 1.2 - 12 балів/ 7,2 балів
2. Лабораторна робота (2 проекти): РН 2.1 - 28 балів/ 14 балів.
3. Лабораторна робота (1 проект): РН 2.2 - 10 балів/ 5,0 балів.
4. Лабораторна робота (1 проект): РН 2.3 -10 балів/ 6 балів

- підсумкове оцінювання (у формі іспиту) вказується:

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН 1.3, РН 2.1
- форма проведення і види завдань: письмова з режимом дистанційного навчання

Види завдань: 4 теоретичних та 4 письмових завдання

Критерії оцінювання екзаменаційної роботи

Завдання	Вид завдання	Максимальний відсоток (бал)	Всього відсотків (балів)
Завдання 1, 2, 3, 4	Письмове запитання з теорії курсу	по 6% (1.6 балів)	50% (30 балів)
Завдання 5, 6, 7,8	Письмове завдання: розв'язування задач на основі теорії курсу	по 5% (2 бали)	50% (10 балів)
Всього			100% (40 балів)

Запитання для підготовки до іспиту

1. Аспекти (парадигми) мов програмування.
2. Класифікація мов програмування за критерієм «Прагматика».

3. Семантика та синтаксис мов програмування;
4. Означення породжуючої граматик.
5. Класифікація граматик за Хомським.
6. Автоматна характеристика граматик.
7. Термінологія в області формальних мов та граматик.
8. Структура мовного процесора типу транслятор та типу інтерпретатор.
9. Характеристика основних блоків мовного процесора (вхід – вихід).
10. Лексичний аналіз в мовних процесорах. Поняття лексеми.
11. Означення скінченого автомата. Недетерміновані та детерміновані скінчені автомати.
12. Алгоритм пошуку недосяжних станів скінченого автомата.
13. Алгоритм пошуку тупикових станів скінченого автомата.
14. Еквівалентні автомати. Мінімальний еквівалентний скінчений автомат.
15. Праволінійні та ліволінійні граматики.
16. Алгоритм побудови скінченого автомата на основі праволінійної граматики.
17. Алгоритм побудови праволінійної граматики на основі скінченого автомата.
18. Регулярні множини та регулярні вирази. Алгебра регулярних виразів. Тотожності в алгебрі регулярних виразів.
19. ПОЛІЗ регулярного виразу. Інтерпретація ПОЛІЗ регулярного виразу.
20. Програмування скінчених автоматів.
21. Методика розробки лексичних аналізаторів мов програмування на основі скінчених автоматів.
22. Інструментальні комплекси автоматизації побудови лексичних аналізаторів мов програмування: LEX, FLEX, BISON, JCC.
23. Контекстно-вільні граматики. Безпосереднє виведення та виведення в граматиці G .
24. Стратегії виведення в граматиці G . Дерево виведення ланцюжка w в граматиці G .
25. Аналіз π ланцюжка w в граматиці G . Відтворення синтаксичного дерева на основі аналізу π ланцюжка w (лівостороння стратегія виводу).
26. Алгоритм пошуку ε -нетерміналів.
27. Алгоритм пошуку недосяжних нетерміналів.
28. Алгоритм пошуку непродуктивних правил.
29. Алгоритм пошуку ліво- (право-) рекурсивних нетерміналів.
30. Означення $LL(k)$ -граматики. Конструктивні означення $LL(k)$ -граматики.
31. Наслідки означення $LL(k)$ -граматики.
32. Означення множини $FIRST_k(w)$. Алгоритм пошуку $FIRST_k(w_1w_2)$.
33. Алгоритм пошуку $FIRST_k(A)$, $A \in N$.
34. Сильні $LL(k)$ -граматики.
35. Означення множини $FOLLOW_k(w)$. Алгоритм пошуку $FOLLOW_k(A)$, $A \in N$.
36. Таблиця управління $LL(1)$ -синтаксичним аналізом.
37. Алгоритм $LL(1)$ -синтаксичного аналізу.
38. Методика побудови синтаксичних аналізаторів мов програмування за умови, що побудувати $LL(1)$ -граматику неможливо (використання сильної $LL(2)$ -граматики).
39. Методика розробки синтаксичних аналізаторів на основі синтаксичних діаграм для $LL(1)$ -граматик.

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою 24 балів

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 20 балів. Студент допускається до іспиту за умови виконання 70% передбачених планом лабораторних робіт.

Контрольні запитання до частини 1.

1. Блочна структура мовного процесора типу інтерпретатор.
2. Алгоритм пошуку недосяжних станів скінченого автомата.
3. Алгоритм мінімізації детермінованого скінченого автомата.
4. Алгоритм розв'язування системи лінійних рівнянь в алгебрі регулярних виразів.
5. Алгоритм інтерпретації ПОЛІЗ регулярного виразу.
6. Методика програмування лексичних аналізаторів на основі скінчених автоматів.

Основні теоретичні питання

1. Скінченні автомати (СА). Детерміновані та недетерміновані СА. Повністю визначені СА. Недетермінізм в скінчених автоматах.
2. Побудувати скінченний автомат M , такий що
$$L(M) = L(M1) \cup L(M2),$$
де $M1$ та $M2$ - скінченні автомати.
3. Означення безпосереднього виведення та виведення в граматиці G .
4. Означення ϵ - нетермінала. Алгоритм пошуку ϵ - нетерміналів для КС-граматики.
5. Означення мовного процесора. Типи мовних процесорів. Вхідні та вихідні дані інтерпретатора.
6. Регулярні множини та регулярні вирази. Тотожності в алгебрі регулярних виразів. Співвідношення між регулярними множинами та регулярними виразами.
7. Алгоритм побудови скінченого автомата на основі праволінійної граматики.
8. Поняття фази мовного процесора. Одно- та багато-прохідні мовні процесори.
9. Обернений польський запис для арифметичних виразів (ПОЛІЗ). Алгоритм перетворення арифметичного виразу в форму оберненого польського запису.
10. Означення недосяжного стану в скінченому автоматі. Алгоритм пошуку недосяжних станів в скінченому автоматі.
11. Означення породжуючої граматики G . Структура правил в схемі P для різних типів граматик.
12. Означення тупикового стану для скінченого автомата (СА). Алгоритм пошуку множини тупикових станів СА.
13. Вхідні та вихідні дані синтаксичного аналізатора. Способи завдання синтаксичної структури програми.
14. Конфігурації скінченого автомата (СА). Початкова та заключна конфігурація СА. Такт роботи СА. Мова, котру розпізнає (сприймає) СА.
15. Означення еквівалентності двох скінчених автоматів (СА). Алгоритм побудови мінімального СА для даного детермінованого СА.
16. Побудувати скінченний автомат M , такий що
$$L(M) = L(M1) * L(M2),$$
де $M1$ та $M2$ - скінченні автомати.
17. Структура мовного процесора інтерпретуючого типу (інтерпретатора).
18. Найменший розв'язок рівняння $X = \alpha X + \beta$, де α та β - регулярні вирази. Системи лінійних рівнянь з регулярними коефіцієнтами. Метод Гауса розв'язування лінійних рівнянь з регулярними коефіцієнтами.
19. Побудувати скінченний автомат M , такий що
$$L(M) = \{ L(M1) \},$$
де $M1$ - скінченний автомат.
20. Інтерпретація ПОЛІЗ для регулярного виразу.
21. Означення рекурсивного нетермінала в граматиці G . Алгоритм пошуку ліворекурсивних нетерміналів.
22. Блок лексичного аналізу в мовному процесорі. Вхідні та вихідні дані блока лексичного аналізу.
23. Типи граматик та відповідні їм автомати.

24. Аналіз π , побудова аналізу π на основі лівосторонньої стратегії виводу.
25. Алгоритм відтворення синтаксичного дерева на основі аналізу π .
26. Означення LL(k)-граматики. Наслідки означення.
27. Алгоритм побудови множини First.
28. Алгоритм побудови множини Follow.
29. Алгоритм побудови таблиці управління LL(1) – синтаксичним аналізатором.
30. Алгоритм LL(1) – синтаксичного аналізатора.

Основні алгоритми

1. Алгоритм пошуку недосяжних станів скінченого автомата.
2. Алгоритм пошуку тупикових станів скінченого автомата.
3. Алгоритм побудови мінімального еквівалентного даному скінченого автомата.
4. Методика та алгоритми програмування лексичних аналізаторів на основі скінчених автоматів.
5. Алгоритм перетворення виразу у ПОЛІЗ.
6. Алгоритм інтерпретації ПОЛІЗ регулярного виразу.
7. Типи граматик та відповідні їм автомати.
8. Аналіз π , побудова аналізу π на основі лівосторонньої стратегії виводу.
9. Алгоритм відтворення синтаксичного дерева на основі аналізу π .
10. Означення LL(k)-граматики. Наслідки означення.
11. Алгоритм побудови множини First.
12. Алгоритм побудови множини Follow.
13. Алгоритм побудови таблиці управління LL(1) – синтаксичним аналізатором.
14. Алгоритм LL(1) – синтаксичного аналізатора.

Основні теоретичні питання

1. Алгоритм побудови таблиці управління LL(1) – синтаксичного аналізатора;
2. Алгоритм LL(1) – синтаксичного аналізатора (магазинний автомат);
3. Методика побудови синтаксичного аналізатора за умови невідповідності декількох правил LL(1) – умові.
4. Метод рекурсивного спуску побудови синтаксичного аналізатора для класу LL(1) – граматик.

Типове завдання контрольної роботи №2.

1. Означення LL(k) – граматики. Наслідки означення LL(k) – граматики;
2. Алгоритм побудови таблиці управління LL(1) – синтаксичного аналізатора;
3. Умови, за яких граматики G буде LL(1) – граматикою.
4. Для граматики з наступною схемою правил знайти ϵ -нетермінали:

$$S \rightarrow AB \mid aAB \qquad C \rightarrow DA \mid aAB$$

$$A \rightarrow qwertyB \mid AB \mid \epsilon \qquad D \rightarrow Sa \mid \epsilon$$

$$B \rightarrow CDA \mid aAB$$

5. Сформулювати алгоритм пошуку множини Follow $k(A_i)$, $A_i \in N$.
Для граматики $G = \{N, \Sigma, P, S\}$ з схемою P:

$$S \rightarrow A B \mid \epsilon \qquad A \rightarrow C b \mid a$$

$$B \rightarrow b B \mid \epsilon \qquad C \rightarrow SA$$

знайти множини Follow $_2(A_i)$, $A_i \in N$, де $N = \{S, A, B, C\}$, $\Sigma = \{a, b\}$.

6. Для вище означеної граматики побудувати МП-автомат, який моделює лівосторонню стратегію виводу ω в граматиці G.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота (тест): до 9 тижня семестру.
2. 3. Лабораторна робота 1,2 (проект): до 6 тижня семестру.

4. *Лабораторна робота 3 (проект): до 8 тижня семестру.*

5. *Лабораторна робота 4 (проект): до 9 тижня семестру.*

Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 5 балів за кожну. Термін перескладання визначається викладачем.

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 лекції	Кількість годин		
		Лекції	Лабор. занять	Сам. р-та
Частина 1. Теорія та практика побудови мовних процесорів. (Лексичний аспект)				
1	Тема 1. Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора. Структура мовного процесора типу транслятор та типу інтерпретатор. Основні блоки мовного процесора.. Перегляди мовного процесора.	2		8
2	Тема 2. Лексичний аналіз. Функції лексичного аналізатора. Вхідні та вихідні структури даних лексичного аналізатора. Скінчені автомати. Способи завдання скінчених автоматів. Недетерміновані та детерміновані скінчені автомати. Алгоритми перетворення недетермінованого скінченого автомата в детермінований	2	2	8
3	Тема 3. Регулярні множини та регулярні вирази. Основні тотожності над регулярними виразами. Регулярні множини та скінчені автомати. Алгоритм перетворення регулярного виразу у ПОЛІЗ. Інтерпретація ПОЛІЗ регулярного виразу. Приклади.	2	2	8
4	Тема 4. Праволінійні граматики та скінчені автомати. Побудова скінченого автомата на основі праволінійної граматики. Система лінійних рівнянь в алгебрі регулярних виразів. Метод Гаусса. Приклади.	2		8
5	Тема 5. Програмування скінчених автоматів. Прямий та непрямий лексичний аналіз в мовних процесорах. Побудова лексичних аналізаторів на основі скінчених автоматів. Інструментальні системи побудови лексичних аналізаторів. Приклади.	4	2	8
Всього за частиною 1		12	6	45
Частина 2. Теорія та практика побудови мовних процесорів. (Синтаксичний аспект)				
6	Тема 6. Порождуючі граматики. Класифікація граматик по Хомському. Порождуючі граматики та еквівалентні їм класи автоматів. Аналіз ланцюжка W в граматиці G . Побудова дерева виводу на основі лівостороннього аналізу ланцюжка W . Приклади	2	2	9
7	Тема 7. Магазинні автомати. Побудова МП-автомата, що моделює лівосторонній вивід ланцюжка W в граматиці G . Приклади.	4	2	9
8	Тема 8. Означення $LL(k)$ -граматики. Властивості $LL(k)$ -граматик. Приклади. Алгоритми побудови множин $FIRST(k)$ та $FOLLOW(k)$. Приклади.	4	2	9
9	Тема 9. Перевірка $LL(1)$ -умови для довільної KS -граматики. Побудова $LL(1)$ -таблиці для управління роботою $LL(1)$ -синтаксичним аналізатором. Приклади	4	2	9
Контрольна робота		2		
Всього за частиною 2		16	8	36
ВСЬОГО		28	14	76
Іспит				

Загальний обсяг 120 год., в тому числі:

Лекцій – **28 год.**

Лабораторні заняття - **14 год.**

Самостійна робота - **76 год.**

Консультації – **2 год.**

Завдання для самостійної роботи. Проектування та програмування основних функцій для реалізації лабораторної роботи Програмування основних модулів лабораторної роботи. Підготовка тестів. Тестування.

Виконання лабораторних робіт 1-5.

Умови лабораторних робіт:

Лабораторна робота 1: Побудова лексичного аналізатора.

Лабораторна робота 2: Розробити та реалізувати представлення скінченого автомата.

Лабораторна робота 3: Реалізувати лексичний аналізатор мови програмування.

Лабораторна робота 4: Потoki. Взаємодія потоків та задач.

9. Рекомендовані джерела:

Основна

1. Richard Antony. System Programming. 1st Edition. Morgan Kaufman. ISBN 9780128007297, 2015. – 548 p.
2. Cooper, Keith D., and Torczon. Engineering a Compiler, Morgan Kaufmann, 2004, ISBN 1-55860-699-8.
3. Ахо А. Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Т1. М. Мир. 1978.
4. Грис Д. Построение компиляторов для ЦЭВМ. М. Мир. 1976.
5. Льюис Ф., Стирнз Р., Розенкранц Д. Теоретические основы построения компиляторов. М. Мир. 1979.
6. Вирт Н. Систематическое программирование. Введение. М.Мир. 1977.

Додаткова

1. Агафонов В.Н. Синтаксический анализ языков программирования. Новосибирск. Изво НГУ. 1981.
2. Братчиков И.А. Синтаксис языков программирования. М. Наука. 1975.
3. Вайнгартен Ф. Трансляция языков программирования. М. Мир. 1977.
4. Вирт Н. Систематическое программирование. Введение. М.Мир. 1977.
5. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки, программирование. Киев. Наукова думка. 1974.
6. Ингерман П. Синтаксически ориентированный транслятор. М. Мир. 1969.
7. Лебедев В.Н. Введение в системы программирования. М. Статистика. 1975.
8. Миккиман У., Хорнинг Дж., Уортман Д. Генератор компиляторов. М. Статистика. 1980.
9. Пратт Т. Языки программирования: разработка и реализация. М. Мир. 1979.
10. Чантер Р. Проектирование и конструирование компиляторов. М. финансы и статистика. 1984.
11. Бек Д. Введение в системное программирование. М. Мир. 1988.