

Чернівецький національний університет імені Юрія Федьковича
(повне найменування вищого навчального закладу)

Факультет математики та інформатики

(назва інституту / факультету)

Кафедра прикладної математики та інформаційних технологій

(назва кафедри)

СИЛАБУС

навчальної дисципліни

Системне програмування

(назва навчальної дисципліни)

обов'язкова

(вказати: обов'язкова / вибіркова)

Освітньо-професійна програма Інформаційні технології та управління проектами
(назва програми)

Спеціальність 122 – Комп'ютерні науки
(вказати: код, назва)

Галузь знань 12 – Інформаційні технології
(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)
(вказати: перший бакалаврський/другий магістерський)

Факультет математики та інформатики
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська
(вказати: на якій мові читається дисципліна)

Розробники: Сопронюк Т. М., доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Профайл викладача: <http://pm.fmi.org.ua/employees/23588>

Контактний тел.: 0663141113

E-mail: t.sopronyuk@chnu.edu.ua

Сторінка курсу в Moodle: <http://e-learning.fpm.chnu.edu.ua/course/view.php?id=399>

Консультації Онлайн-консультації: П'ятниця з 13.00 до 14.20

1. Анотація дисципліни (призначення навчальної дисципліни).

У курсі вивчаються елементи теорії формальних мов (форми Бекуса-Наура, регулярні вирази, формальні граматики, зокрема, граматики Хомського) та теорії скінченних автоматів (розпізнавачі, недетерміновані скінченні автомати, алгоритми перетворення недетермінованого скінченного автомата в детермінований, алгоритми вилучення недосяжних станів, алгоритми мінімізації та інші), які використовуються при лексичному і синтаксичному аналізі.

2. Мета навчальної дисципліни: студенти повинні опанувати основні принципи побудови компіляторів, елементи теорії формальних мов (регулярні вирази, формальні граматики, зокрема, граматики Хомського, розпізнавачі) та теорію скінченних автоматів.

3. Пререквізити. Для ефективності засвоєння курсу здобувач вищої освіти має вивчити дисципліну «Програмування», «Об'єктно-орієнтоване програмування».

4. Результати навчання.

У результаті вивчення навчальної дисципліни студент повинен

знати: типи мовних процесорів, основні фази компіляції, роботу з хеш-таблицями, поняття автоматних і неавтоматних мов та засоби їх представлення, алгоритми перетворення різних форм подання формальних мов, алгоритми перетворення автоматів (недетермінованого скінченного автомата в детермінований, алгоритми мінімізації та інші);

вміти: розв'язувати задачу належності для граматик, що розпізнають; генерувати ланцюжки граматик, що породжують; застосовувати алгоритми перетворення скінченних автоматів, праволінійних граматик та регулярних виразів.

Студенти повинні оволодіти програмним матеріалом, застосувати вивчені алгоритми до модельних прикладів, запрограмувати частину алгоритмів, виконати контрольні роботи.

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

загальні компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК8. Здатність генерувати нові ідеї (креативність)

фахові компетентності:

ФК1. Здатність до математичного формулювання та дослідження неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування

ФК3. 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.

ФК8. 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.

та отримуються наступні **програмні результати навчання:**

ПРН1. 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.

ПРН13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

5. Опис навчальної дисципліни

5.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин					Вид підсумкового контролю	
			кредитів	годин	змістових модулів	лекції	практичні	семінарські	лабораторні	самостійна робота		індивідуальні завдання
Денна	3	6	4	120	2	30	-	-	30	60	-	Залік
Заочна	-	-	-	-	-	-	-	-	-	-	-	-

5.2. Дидактична карта навчальної дисципліни

Назви змістових модулів і тем	Кількість годин												
	денна форма						Заочна форма						
	усьог	у тому числі					усьог	у тому числі					
		о	л	п	лаб	інд		с.р.	л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13	
Змістовий модуль 1. Обчислювальні та регулярні вирази. Вирази з метасимволами													
Тема 1. Розробка мовних процесорів мов програмування (загальний огляд) Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора (лексичний аналіз, робота з таблицями, синтаксичний аналіз, генерація проміжного коду, оптимізація коду, генерація машинного коду, аналіз помилок). Спрощена модель компілятора. Проходи компілятора.	16	4	-	4		8							
Тема 2. Організація таблиць імен Таблиці розміщення. Схеми хешування. Хешування зі списками. Алгоритм обчислення адреси хеш-таблиці розміщення в одновимірному випадку. Первинні і вторинні функції розміщення.	12	3	-	3		6							

Тема 3. Формальні мови і граматики Алфавіт. Ланцюжки. Означення формальної мови. Способи визначення мов. Граматики, що породжують і розпізнають. Задача належності. Регулярні операції над мовами. Метамова БНФ. Розширення БНФ. Граматики Хомського. Спосіб визначення мови за допомогою граматик. Вивід ланцюжка в граматиці G і його аналіз. Ієрархія граматик Хомського.	16	4	-	4		8							
Тема 4. Регулярні множини і регулярні вирази Регулярні множини та регулярні вирази. Тотожності над регулярними виразами. Системи з регулярними коефіцієнтами. Алгоритми перетворення праволінійної граматики в регулярний вираз і навпаки.	16	4	-	4		8							
Разом за змістовим модулем 1	60	15	-	15		30							
Змістовий модуль 2. Скінченні автомати і контекстно-вільні граматики													
Тема 5. Скінченні автомати Розпізнавачі (структура, конфігурація). Мова, що дозволяється розпізнавачем. Способи завдання скінченних автоматів. Детерміновані та недетерміновані скінченні автомати. Функція переходів.	16	4	-	4		8							

<p>Алгоритм вилучення недосяжних станів ДСК. Розширена (узагальнена) функція переходів. Перевірка еквівалентності станів. Побудова таблиці нееквівалентності станів. Перевірка еквівалентності регулярних мов. Алгоритм перетворення недетермінованого скінченного автомата в детермінований. Мінімізація скінчених автоматів.</p>												
<p>Тема 6. Побудова спрощеного лексичного аналізатора Етапи побудови лексичного аналізатора. Зв'язок між регулярними множинами, скінченими автоматами та праволінійними граматиками. Побудова лексичних аналізаторів на основі скінчених автоматів. Алгоритм побудови недетермінованого скінченного автомата за регулярним виразом. Програмування скінчених автоматів та лексичних аналізаторів на прикладі лексичного аналізатора, що розпізнає одну лексему – дійсне число.</p>	12	3	-	3		6						
<p>Тема 7. Синтаксичний аспект в мовах програмування Породжуючі граматика. Контекстно-вільні граматика. Дві</p>	16	4	-	4		8						

ідеї аналізу. Дерево виводу. Лівостороння та правостороння стратегії виводу. Означення та властивості LA(1)-граматики. Множини FIRST і FOLLOW та алгоритми їх побудови. Ліворекурсивні та розширені рекурсії. Синтаксичні діаграми. Застосування алгоритму LA(1)-аналізу.											
Тема 8. Автоматні мови та регулярні вирази Регулярні і нерегулярні мови. Лема про накачку. Приклади доведення нерегулярності мов. Алгоритм перетворення НСА в регулярний вираз за допомогою рекурентних формул. Алгоритм перетворення ДСА в регулярний вираз методом вилучення станів.	16	4	-	4		8					
Разом за змістовим модулем 2	60	15	-	15		30					
Усього годин за 4-й семестр	120	30	-	30		60					

Теми лабораторних занять

Модуль 1. Обчислювальні та регулярні вирази. Вирази з метасимволами (40 балів)

Елемент модуля	Кількість балів	Час виконання
Лабораторна робота №1: Розпізнавання ланцюжків по заданих шаблонах в текстових файлах	12	6 годин
Домашнє завдання №1: Виконання всіх етапів компіляції для оператора присвоєння	6	2 години

Домашнє завдання №2: Граматики, що розпізнають ланцюжки	3	1 година
Лабораторна робота №2: Створення діалогових вікон Find, Replace, Delete з шаблонами для обробки текстових документів	10	6 годин
Домашнє завдання №3: Граматики Хомського, регулярні вирази і скінченні автомати	9	3 години

Модуль 2. Скінченні автомати і контекстно-вільні граматики (50 балів)

Елемент модуля	Кількість балів	Час виконання
Домашнє завдання №4: Моделювання скінченних автоматів	4	1 година
Лабораторна робота №3: Побудова скінченного автомата і праволінійної граматики	16	8 годин
Лабораторна робота №4: Побудова аналізаторів методом рекурсивного спуску	10	5 годин
Домашнє завдання №5: Побудова регулярного виразу по скінченному автомату	5	2 години
Підсумкова контрольна робота	10	

Підсумковий модуль. Залік (тести - 15 балів)

Система контролю та оцінювання

Види та форми контролю

Під час проведення лекцій використовуються пасивний та активний методи навчання. Консультаційна робота. Під час виконання студентами лабораторних робіт використовується активні методи навчання. Проведення модульних контрольних робіт та навчальна робота під час прийому лабораторних і домашніх робіт.

Методи контролю

1. Контрольна робота.
2. Оцінювання на лабораторних заняттях в обох модулях.
3. Тестові завдання на заліку засобами системи Moodle.

Захист та критерії оцінювання лабораторної роботи

- Здача лабораторної роботи проводиться під час заняття згідно з календарними планом.
- Для захисту лабораторної роботи кожен студент має самостійно виконати лабораторну роботу і здати її викладачу **на занятті**.
- Не допускається заочне прийняття програм (електронною поштою) без запуску програм з різними вхідними даними.

- Під час здачі програми викладач зобов'язаний перевіряти здатність студента орієнтуватися у власній програмі, пропонуючи йому виконати нескладні зміни, розраховані на 5-10 хвилин поточного заняття.
- При необхідності виконання частини завдання або усього завдання у **робочому** зошиті, бали виставляти у зошиті, вказуючи число і підпис.
- Під час здачі лабораторної роботи студент повинен
 - вміти пояснити постановку задач, які розв'язувались в лабораторній роботі; алгоритм розв'язування задач; програмну реалізацію завдання;
 - продемонструвати розуміння програми та обґрунтувати зроблені висновки;
 - відповісти на питання, які належать до виконання лабораторної роботи та додаткові теоретичні питання, якщо розданий перелік таких питань.
- Якщо студент не розуміє алгоритму розв'язання задачі, не орієнтується в програмній реалізації, але є у наявності правильно виконувана програма, то робота зараховується не більше як на 30%.
- Якщо студент розуміє задачу і алгоритм її виконання, але не орієнтується (слабо орієнтується) в практичній частині (програмній реалізації), то оцінка знижується до 50%.
- Якщо програма не працює, або працює частково і студент може пояснити алгоритм, роботу оцінювати частково, в залежності від об'єму і якості коду.
- Кількість балів за лабораторну роботу визначає викладач в процесі здачі. Оцінка повідомляється студенту.
- За невчасний захист лабораторних робіт у межах модуля допускається знімати по одному балу за кожне прострочене заняття, якщо робота оцінюється до 10 балів, і по 1,5-2 бали, якщо робота оцінюється в межах від 11 до 20 балів, але не більше половини балів.

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота									контрольн а	Залік	Сума
Змістовий модуль №1					Змістовий модуль № 2						
T1	T2	T3	T4	T4	T5	T6	T7	T8			
12	6+3	10	9	4	16	10	5	10	10	15	100

T1, T2 ... T8 – теми змістових модулів.

Критерії оцінювання результатів навчання з навчальної дисципліни

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	задовільно	
50-59	E		
35-49	FX	незадовільно з можливістю повторного	не зараховано з можливістю

		складання	повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Рекомендована література

1. Alfred Aho, Jeffrey Ullman, Ravi Sethi, Monica Lam. Compilers: Principles, Techniques, and Tools 2nd Edition. – Addison Wesley, 2006. – 1040 p.
2. Jeffrey E. F. Friedl. Mastering Regular Expressions, Third Edition. – O'Reilly Media, 2006. – 517 p.
3. John E. Hopcroft, Jeffrey D. Ullman, Rajeev Motwani. Introduction to Automata, Theory, Languages and Computation 2nd Edition. – Addison-Wesley, 2000. – 521 p.
4. Системне програмування: Методичні рекомендації та завдання для лабораторних робіт/ Укл. Т.М. Сопронюк.– Чернівці: ЧНУ, 2003. – 33 с.
5. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
6. Сопронюк Т.М. Елементи теорії формальних мов: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с

Інформаційні ресурси

1. <http://www-db.stanford.edu/~ullman/ialc.html> – Introduction to Automata Theory, Languages, and Computation. Slides and Lecture Notes. (Stanford University)
2. <http://www.unicyb.kiev.ua/Library/PROG/Zmist.htm> – А.Б.Ставровський. Посібник з програмування (факультет кібернетики Київського національного університету)
3. http://uk.wikipedia.org/wiki/Формальні_граматики
4. <http://courses.cs.vt.edu/~cs4114/lectures/> – Formal Languages and Automata Theory Course Lecture notes
5. http://khpri-iip.mipk.kharkiv.edu/library/datastr/book_sod/kgsu/oglav.html – Мова програмування C++. Динамічні структури даних

Додаток

Методичне забезпечення

1. Системне програмування: Методичні рекомендації та завдання для лабораторних робіт/ Укл. Т.М. Сопронюк.– Чернівці: ЧНУ, 2003. – 33 с.
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. Сопронюк Т.М. Елементи теорії формальних мов: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
4. Тестові завдання (Система Moodle)
5. Презентації лекцій (Система Moodle)
6. Відео-лекції на Google диску
7. Сертифікат про закінчення курсів професора Стенфордського університету Дж.Ульмана “Automata” (<https://drive.google.com/file/d/0B-cxrxmP0J7XRUI2YINGem1wb3c/view?resourcekey=0--NXsRIO20hZHdxuVKIDQ4Q>)