



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

ЗАТВЕРДЖЕНО

В.О. Зав. кафедри «Теорії та технології програмування»

\_\_\_\_\_ (Панченко Т.В.)  
(підпис) (прізвище та ініціали)

Протокол № \_\_\_\_ від «\_\_\_\_» \_\_\_\_\_  
20\_\_ р.

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

---

Протокол від «\_\_\_\_» \_\_\_\_\_ 20\_\_ року №\_\_

Голова науково-методичної комісії \_\_\_\_\_ (Хусаїнов Д.Я.)  
(підпис) (прізвище та ініціали)

«\_\_\_\_» \_\_\_\_\_ 20\_\_ року

## ВСТУП

**1. Мета дисципліни.** Навчальна дисципліна “Системне програмування” є складовою освітньо-професійної програми підготовки фахівців за освітньо-кваліфікаційним рівнем «бакалавр» галузі знань “Математика та статистика” з напряму підготовки „Інформатика”, спеціальності - 113 «Прикладна математика».

Дана дисципліна базова нормативна за спеціальністю “Прикладна математика”.

Викладається у 6 семестрі 3 курсу в обсязі – 216 год.

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

1. *Знати:* загальні принципи проектування алгоритмів обробки несислової інформації; мови програмування C++ та C# 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++ та C# 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 11 „ Математика та статистика ” зі спеціальності 113 „Прикладна математика”, освітньо-професійної програми – „Прикладна математика”.

Предмет навчальної дисципліни „Системне програмування” включає в себе розгляд теоретичних аспектів проектування та створення мовних процесорів, вивчення відповідних класів граматик, опанування алгоритмів та їх програмування з метою отримання практичних навиків реалізації мовних процесорів.

Дана дисципліна є обов’язковою навчальною дисципліною за програмою “ Прикладна математика ”.

Викладається у 6 семестрі 3 курсу в обсязі – 150 год.

**(5 кредитів ECTS)** зокрема: *лекції – 34 год., лабораторні – 38 год., консультації – 2 год., самостійна робота – 76 год.* У курсі передбачено **3 змістових модулів** та **2 модульні контрольні роботи**. Завершується дисципліна – **екзаменом в 6 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 Java;

**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;

**5. Зв’язок з іншими дисциплінами.** Дисципліна “Системне програмування” є базовою для засвоєння всіх інших курсів та спецкурсів програміського спрямування, окремих розділів теорії алгоритмів та математичної логіки.

**6. Завдання (навчальні цілі):** набуття знань, умінь та навичок, формування компетентностей на рівні новітніх досягнень у програмуванні, відповідно до кваліфікації фахівця з інформаційних технологій:

- здатність проектувати та розробляти програмне забезпечення із застосуванням об'єктно-орієнтованого (Java – технології) програмування та проектування та реалізації алгоритмів, які є основою для розробки мовних процесорів;
- застосовувати знання у практичних ситуаціях;
- знання та розуміння предметної області та розуміння професійної діяльності;
- здатність вчитися й оволодівати сучасними знаннями;
- здатність генерувати нові ідеї (креативність);
- здатність розробляти й управляти проектами;
- здатність оцінювати та забезпечувати якість виконуваних робіт.

**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 оволодіти основні алгоритми розробки мовних процесорів	Лекція	Тест, 60% правильних відповідей	19%
РН 1.2	Знати принципи роботи Java-орієнтованого інструментального комплексу розробки програмного забезпечення.	Лекція		
РН 1.3	Знати та оволодіти принципи функціонування JVM – середовища.	Лекція	Тест, 60% правильних відповідей	18%
РН 2.1	Вміти працювати з Java-орієнтованим інструментальним комплексом розробки програмного забезпечення.	Лабораторна робота, самостійна робота	Захист лабораторної роботи	21%
РН 2.2	Вміти працювати пакетом програм розробки компонент мовного процесора	Лабораторна робота, самостійна робота	Захист лабораторної роботи	21%
РН 2.3	Вміти самостійно описувати синтаксис мови програмування з орієнтацією на відповідний клас граматики.	Лабораторна робота, самостійна робота	Захист лабораторної роботи	21%

\* заповнюється за необхідністю, наприклад для практик, лабораторних курсів тощо.

\*

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 2.2	PH 2.3
Програмні результати навчання <i>(з опису освітньої програми)</i>						
ПР9. Використовувати інструментальні засоби розробки програмного забезпечення	+	+		+		+
ПР10. Проектування та розробка основних алгоритмів мовного процесора	+			+	+	+
ПР11. Володіти навичками використання пакету програм з інтеграцією власних алгоритмів.		+	+		+	
ПР 12. Вміння самостійно описати синтаксис (фрагмент синтаксису мови програмування) на прикладі SQL-мови.				+	+	+

### Контроль знань і розподіл балів, які отримують студенти.

Контроль здійснюється за модульно-рейтинговою системою.

У змістовий модуль 1 (ЗМ1) входять теми 1- 5, у змістовий модуль 2 (ЗМ2) – теми 6 - 17. Обов'язковим для заліку у кожному семестрі є отримання студентом-магістром протягом 1 семестру не менше 21 балу.

*Оцінювання за формами контролю: (як приклад)*

	ЗМ1		ЗМ2	
	Min. – 10 балів	Max. – 30 бали	Min. – 11 балів	Max. – 30 балів
Модульна контрольна робота	5	10	5	10
Лабораторна робота	5	15	6	15
Активна робота	0	5	0	5

„3” – мінімальна/максимальна оцінку, яку може отримати студент.  
<sup>1</sup> – мінімальна/максимальна залікова кількість робіт чи завдань.

Для студентів, які набрали сумарно меншу кількість балів ніж *критично-розрахунковий мінімум – 21 бал* для допуску до заліку обов'язково повинні перескласти контрольні роботи та здати заплановані лабораторні роботи. Студент має право на одне перескладання контрольної роботи із можливістю отримання максимально 5 балів за кожен. Термін перескладання визначається викладачем.

*Оцінювання за формами контролю: (як приклад)*

	ЗМ1		ЗМ2	
	Min. – 10 балів	Max. – 30 бали	Min. – 11 балів	Max. – 30 балів
Модульна контрольна робота	5	10	5	10
Лабораторна робота	5	15	6	15
Активна робота	0	5	0	5

„3” – мінімальна/максимальна оцінку, яку може отримати студент.  
<sup>1</sup> – мінімальна/максимальна залікова кількість робіт чи завдань.

У випадку відсутності студента з поважних причин відпрацювання та перездачі МКР здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

**При простому розрахунку отримаємо:**

	Змістовий модуль1	Змістовий модуль2	іспит	Підсумкова оцінка
<i>Мінімум</i>	15	14	31	60
<b>Максимум</b>	<b>30</b>	<b>30</b>	<b>40</b>	<b>100</b>

Контроль знань студентів здійснюється за модульно-рейтинговою системою. Результати навчальної діяльності студентів оцінюються за 100-бальною шкалою. Робота в кожному семестрі поділяється на два змістових модуля. При виставленні балів за змістовий модуль враховується: оцінка за модульну контрольну роботу - 10 балів, виконання студентом модульних лабораторних робіт – 10 балів/одна робота, робота на самостійній роботі - 5 балів. Підсумковий контроль проводиться у формі письмового заліку – 30 балів. Підсумкова оцінка  $100=2*(10+2*10+5)+30$ .

Якщо студент з поважних причин, які підтверджено документально, був відсутній при написанні модульної контрольної роботи, він має право на одне перескладання з можливістю отримання максимальної кількості балів. Термін перескладання визначається викладачем.

За активну роботу студента на практичних заняттях та на самостійній роботі до семестрової оцінки може бути додано до 5 балів.

Якщо впродовж семестру студент не з'являвся на заняття (не залежно від причин), не має модульних оцінок, у відповідних графах „Відомості обліку успішності КМСОНП” виставляються „0”, а у графі іспиту – відмітка про недопуск.

Студент допускається до складання заліку, якщо кількість набраних ним балів за семестр становить не менше 30 балів.

Залік вважається не зданим, якщо сумарна кількість балів з дисципліни у семестрі складає менше 60 балів.

**При цьому, кількість балів:**

- **1-34** відповідає оцінці «незадовільно» з обов'язковим повторним вивченням дисципліни;
- **35-59** відповідає оцінці «незадовільно» з можливістю повторного складання;
- **60-64** відповідає оцінці «задовільно» («достатньо»);
- **75 - 84** відповідає оцінці «добре»;
- **85 - 89** відповідає оцінці «добре» («дуже добре»);
- **90 - 100** відповідає оцінці «відмінно».

**Шкала відповідності (за умови іспиту)**

За 100 – бальною шкалою	За національною шкалою	
90 – 100	5	відмінно
85 – 89	4	добре
75 – 84		
65 – 74	3	задовільно
60 – 64		
35 – 59	2	незадовільно
1 – 34		

## ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### **Змістовий модуль 1. Об'єктно-орієнтоване програмування. Оглядові лекції з мови програмування Java.**

**Тема 1.** Поняття об'єкта в мові програмування Java SE 8. Структура об'єкта. Відкрита та закрита компоненти об'єкта. Методи - члени об'єкта. Опис та означення функцій - членів об'єкта. Визначення та опис методів класу. Властивості методів у залежності від атрибутів, які їм надаються. Приклади. (4 год.).

Поняття класу як приватного типу . Об'єкт типу. Синтаксис визначення типу та об'єкта типу. Конструктори. Метод finalize(). Методи класу. Сигнатура методу. Приклади [1], [2].

#### **Тема 2.**

Перевантаження методів. Роль суперкласу при перевантаженні та перевизначенні методів. Приклади. [1], [2]. (4 год.).

#### **Тема 3.**

Виключення та їх обробка. Ієрархія виключень. Побудова системи власних виключень. Приклади. [1], [2]. (4 год.).

**Тема 4.** Колекції. Інтерфейси колекцій та відповідна ієрархія класів. Клонування та серіалізація об'єктів. Приклади. [1], [2]. (4 год.).

**Тема 5.** Узагальнені масиви. Об'єктні оболонки. Поняття рефлексії. Роль JVM при реалізації рефлексії. Методи рефлексії. Приклади. [1], [2]. (4 год.).

### **Змістовний модуль 2. Розробка мовних процесорів мов програмування (лексичний аспект).**

**Тема 6.** Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора. Структура мовного процесора типу транслятор та типу інтерпретатор. Основні блоки мовного процесора. Основні блоки мовного процесора. Перегляди мовного процесора. Приклади [3], [4], [5]. (5 год.).

**Тема 7.** Лексичний аналіз. Функції лексичного аналізатора. Вхідні та вихідні структури даних лексичного аналізатора. Скінчені автомати. Приклади [3], [4], [5]. (7 год.).

**Тема 8.** Способи завдання скінчених автоматів. Недетерміновані та детерміновані скінчені автомати. Алгоритми перетворення недетермінованого скінченого автомата в детермінований. Приклади [3], [4], [5]. (5 год.).

**Тема 9.** 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 [3], [4], [5]. (6 год.).

**Тема10.** Праволінійні граматики та скінчені автомати. Побудова скінченого автомата на основі праволінійної граматики. Система лінійних рівнянь в алгебрі регулярних виразів. Метод Гаусса. Приклади [3], [4], [5]. (6 год.).

**Тема 11.** Програмування скінчених автоматів. Прямий та непрямий лексичний аналіз в мовних процесорах. Побудова лексичних аналізаторів на основі скінчених автоматів. Інструментальні системи побудови лексичних аналізаторів. Приклади [3], [4], [5]. (8 год.).

### **Змістовний модуль 3. Розробка мовних процесорів мов програмування (синтаксичний та семантичний аспекти).**

**Тема 12.** 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 [3], [4], [5]. (6 год.).

**Тема 13.** Магазинні автомати. Побудова МП-автомата, що моделює лівосторонній вивід ланцюжка W в граматиці G. Приклади [3], [4], [5]. (6 год.).

**Тема 14.** Означення LL(k)-граматики. Властивості LL(k)-граматик. Приклади. Множини FIRST(k) та FOLLOW(k). Приклади [3], [4], [5]. (6 год.).

**Тема 15.** Перевірка LL(1)-умови для довільної КС-граматики. Побудова LL(1)-таблиці для управління роботою LL(1)-синтаксичним аналізатором. Приклади [3], [4], [5]. (6 год.).

**Тема 16.** Семантичний аспект в мовах програмування та в мовних процесорах. Методи денотаційної та операційної семантики. Метод семантичних підпрограм. Приклад. [4] - [6] (6 год.).

**Тема 17.** Семантичний терм програми. Семантичний терм програми в формі ПОЛІЗ, в формі семантичного дерева та в формі трійок. Приклад. Машинно-орієнтовані мови програмування. Асемблери - мовні процесори для машинно-орієнтованих мов програмування. Структура асемблера, проходи та бази даних [4] - [6] (6 год.).

## **ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І СЕМІНАРСЬКИХ ЗАНЯТЬ СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

№ лекції	Назва лекції (теми)	Кількість годин		
		Лекції	Лабор. заняття	Самост. робота
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>Шостий семестр</b>				
<b>Змістовий модуль 1. Об'єктно-орієнтоване програмування. Оглядові лекції з мови Java</b>				
1	Поняття об'єкта в мові програмування . Java SE 6, Java SE 7. Методи - члени об'єкта. Опис та означення функцій - членів об'єкта	2		3
2	Перевантаження операцій. Пріоритети при перевантаженні операцій. Виключення та їх обробка.	2	2	3
3	Клонування та серілізація.	2		3
4	Інтерфейси. Колекції	2	2	3
5	Узагальнені масиви. Об'єктні оболонки. Поняття рефлексії. Роль JVM при реалізації рефлексії.	2		3
<b>Модульна контрольна робота 1</b>				
<b>Змістовий модуль 2. Розробка мовних процесорів мов програмування (лексичний аспект)</b>				
6	Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора. Лексичний аналіз.	2	2	3



1	2	3	4	5
7	Функції лексичного аналізатора. Вхідні та вихідні структури даних лексичного аналізатора. Скінчені автомати.	2		3
8	Способи завдання скінчених автоматів. Недетерміновані та детерміновані скінчені автомати. Алгоритми перетворення недетермінованого скінченого автомата в детермінований.	2	2	3
9	Побудова мінімального скінченого автомата еквівалентного даному. Алгоритми.	2		4
10	Регулярні множини та регулярні вирази. Основні тотожності над регулярними виразами. Регулярні множини та скінчені автомати.	2	2	4
11	Програмування скінчених автоматів. Прямий та непрямий лексичний аналіз в мовних процесорах. Побудова лексичних аналізаторів на основі скінчених автоматів	2		4
<b>Змістовий модуль 3. Розробка мовних процесорів мов програмування (синтаксичний та семантичний аспекти)</b>				
12	Породжуючі граматики. Класифікація граматик по Хомському. Породжуючі граматики та еквівалентні їм класи граматик.	2	2	2
13	Магазинні автомати. Побудова МП-автомата, що моделює лівосторонній вивод ланцюжка $W$ в граматиці $G$ .	2		2
14	Означення LL(k)-граматики. Властивості LL(k)-граматик. Алгоритми побудови множин FIRST(k) та FOLLOW(k). Програмна реалізація алгоритмів пошуку множин FIRST(1) та FOLLOW(1).	2	2	2
15	Перевірка LL(1)- умови для довільної КС- граматики. Побудова LL(1)- таблиці для управління роботою LL(1)- синтаксичним аналізатором.	2		2
16	Методи денотаційної та операційної семантики. Метод семантичних підпрограм. Атрибутний метод визначення семантики програм. Синтезовані та успадковані атрибути. Порядок та правила обчислення атрибутів	2	2	2
	<b>Модульна контрольна робота 2</b>	1		
	<b>Іспит за VI семестр</b>			

**Загальний обсяг 216 годин, в тому числі:**

- Лекцій – 34 год.,
- Лабораторні заняття – 34 год.,
- Самостійна робота – 64 год.

### **Змістовий модуль 1. Об'єктно-орієнтоване програмування. Оглядіві лекції з мови програмування Java.**

**Лекція 1.** Поняття об'єкта в мові програмування Java SE 8. Структура об'єкта. Відкрита та закрита компоненти об'єкта. Методи - члени об'єкта. Опис та означення функцій - членів об'єкта. Визначення та опис методів класу. Властивості методів у залежності від атрибутів, які їм надаються. Приклади. (2 год.).

Поняття класу як приватного типу . Об'єкт типу. Синтаксис визначення типу та об'єкта типу. Конструктори. Метод finalize(). Методи класу. Сигнатура методу. Приклади [1].

**Завдання для самостійної роботи.** Ознайомлення з середовищем NetBeans IDE: Створення проекту. Компоненти проекту. Збереження проекту. Пакети проекту [1]. – 3 год.

**Лекція 2.** Перевантаження операцій. Пріоритети при перевантаженні операцій. Виключення та їх обробка. Приклади [1]. (2 год.).

**Завдання для самостійної роботи.** Ознайомлення з середовищем NetBeans IDE: створення проекту. Компоненти проекту. Збереження проекту. Пакети проекту [1]. – 3 год.

**Лабораторне заняття 1.** Мова програмування JAVA. Виконання лабораторної роботи №1. Вивчення класів, пов'язаних з потоковим вводом/виводом даних. [1]. – 2 год.

**Лекція 3.** Рефлексія в Java. Приклади [1]. Поняття рефлексії. Роль JVM при реалізації рефлексії. Методи рефлексії. Приклади. (2 год.).

**Завдання для самостійної роботи.** Проектування методів проекту для лабораторної роботи №1. Тестування методів. Інтеграція до проекту [1]. – 3 год.

**Лекція 4.** Інтерфейси. Колекції. Приклади. [1] (2 год.).

**Завдання для самостійної роботи.** Проектування методів проекту для лабораторної роботи №1. Тестування методів. Інтеграція до проекту [1]. – 3 год.

**Лабораторне заняття 2.** Постановка лабораторної роботи №1: «Робота з файлами з використанням базових класів мови Java». Аналіз задачі, розробка проекту та кодування функцій (методів) проекту. [1]. – 2 год.

**Лекція 5.** Узагальнені масиви. Об'єктні оболонки. Поняття рефлексії. Роль JVM при реалізації рефлексії. Приклади [1]. (2 год.).

**Завдання для самостійної роботи.** Вивчення матеріалу для виконання лабораторної роботи №2: способи визначення та програмування скінчених детермінованих та недетермінованих скінчених автоматів [1]. – 3 год.

### **Контрольні запитання до змістового модуля 1.**

1. Поняття класу. Синтаксис визначення класу. Синтаксис визначення методу класу.
2. Відкрита та закрита частина класу. Конструктори класу. Створення класів без можливостей створення об'єктів відповідного класу.
3. Статичні класи. Статичні методи класів.
4. Терміни overload та override: перевантаження методів та їх переписування.
5. Базові та похідні класи.
6. Абстрактні класи. Поліморфізм.
7. Інтерфейси. Конструктори для похідних класів.
8. Поняття виключення. Синтаксис обробки виключення.
9. Проектування власних виключень та їх використання.

### **Типове завдання модульної контрольної роботи №1.**

1. Внутрішні локальні класи. Синтаксис. Приклади.
2. Класи. Визначення класу. Базовий клас та інтерфейси.
3. Не використовуючи інших методів (окрім конструктора, довжина рядка та доступ до i-го символу ) запрограмуйте наступні методи класу **String** :
  - 3.1. Boolean **startsWith(String prefix, int toffset)**

Метод повертає true, якщо prefix є префіксом рядка. Що починається з позиції toffset, інакше - false.

3.2. int **compareTo**(String anotherString) //лексикографічне порівняння.  
Метод повертає 0 - рядки рівні, >0 - перший рядок «більший», <0- перший рядок «менший»

3.3. int **indexOf**(String str)  
Метод повертає індекс першого входження str у рядок.

4. Програміст визначив список об'єктів наступним чином (на перспективу):

ArrayDeque <Object> testDeque;

Фактично в цей список були записані екземпляри класів Date та екземпляри інших класів. Визначте конструкцію управління для обробки лише даних типу Date.

## **Змістовний модуль 2. Розробка мовних процесорів мов програмування (лексичний аспект).**

**Лекція 6.** Поняття мовного процесора. Типи мовних процесорів. Основні фази мовного процесора. Структура мовного процесора типу транслятор та типу інтерпретатор. Основні блоки мовного процесора. Приклади [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Підготовка тестових автоматів для Лабораторної роботи №2. Тестування програми з використанням створених (описаних) автоматів [2], [3], [4]. – 3 год.

**Лабораторне заняття 3.** Програмування методів (функцій), компіляція проекту, інтеграція та виявлення логічних помилок. [2], [3], [4]. – 2 год.

**Лекція 7.** 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. Приклади [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Підготовка тестових автоматів для Лабораторної роботи №2. Тестування програми з використанням створених (описаних) автоматів [2], [3], [4]. – 3 год.

**Лекція 8.** Алгоритми перетворення недетермінованого скінченого автомата в детермінований. Побудова мінімального скінченого автомата еквівалентного даному. Приклади [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Підготовка тестових автоматів для Лабораторної роботи №2. Тестування програми з використанням створених (описаних) автоматів [2], [3], [4]. – 4 год.

**Лабораторне заняття 4.** Програмування методів (функцій), компіляція проекту, інтеграція та виявлення логічних помилок. [2], [3], [4]. – 2 год.

**Лекція 9.** Побудова мінімального скінченого автомата еквівалентного даному. Алгоритми.

**Завдання для самостійної роботи.** Опис класів лексем для лабораторної роботи №3. Побудова відповідних скінчених автоматів [2], [3], [4]. – 4 год.

**Лекція 10.** 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], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Опис класів лексем для лабораторної роботи №3. Побудова відповідних скінчених автоматів [2], [3], [4]. – 4 год.

**Лабораторне заняття 5.** Задача лабораторної роботи №2. Постановка задачі для Лабораторного практикуму №3 : „Реалізація лексичного аналізатора для відомих мов програмування” [2], [3], [4]. – 2 год.

**Лекція 11.** Програмування скінчених автоматів. Прямий та непрямий лексичний аналіз в мовних процесорах. Побудова лексичних аналізаторів на основі скінчених автоматів. Приклади [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Програмування та тестування створеного програмного забезпечення. Підготовка тестів [2], [3], [4]. – 4 год.

### **Контрольні запитання до змістового модуля 2.**

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  $\epsilon$  - нетерміналу. Алгоритм пошуку  $\epsilon$  - 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 грамастик.
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 скінчений 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  $\alpha$  та  $\beta$  - 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.

### **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.

### **Змістовний модуль 3. Розробка мовних процесорів мов програмування (синтаксичний та семантичний аспекти).**

**Лекція 12.** Порождуючі граматики. Класифікація граматик по Хомському. Порождуючі граматики та еквівалентні їм класи граматик. Приклади [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Вивчення функціональних можливостей середовища MS Visual Studio в обсязі: виконання функцій Debug, побудова компонент проекту, компіляція елементів проекту [2], [3], [4]. – 2 год.

**Лабораторне заняття 6.** Постановка задачі щодо виконання лабораторних робіт у другому семестрі. Проект компілятора. Структура проекту. Модель користувача. Інтеграція модуля користувача до проекту. Виконання. [2], [3], [4]. – 2 год.

**Лекція 13.** Магазинні автомати. Побудова МП-автомата, що моделює лівосторонній вивод ланцюжка  $W$  в граматиці  $G$ . Аналіз ланцюжка  $W$  в граматиці  $G$ . Побудова дерева виводу на основі лівостороннього аналізу ланцюжка  $W$ . Приклади [2], [3], [4]. - 2 год.

**Завдання для самостійної роботи.** Вивчення структури глобальної специфікації на інтерфейси та структури даних проекту [2], [3], [4]. – 2 год.

**Лекція 14.** Означення  $LL(k)$ -граматики. Властивості  $LL(k)$ -граматик. Алгоритми побудови множин  $FIRST(k)$  та  $FOLLOW(k)$ . Програмна реалізація алгоритмів пошуку множин  $FIRST(1)$  та  $FOLLOW(1)$ . [2], [3], [4]. - 2 год.

**Завдання для самостійної роботи.** Проектування та програмування основних функцій для реалізації лабораторної роботи [2], [3], [4]. – 2 год.

**Лабораторне заняття 7.** Постановка задачі Лабораторна робота №4: „Програмування основних алгоритмів аналізу контекстно- вільних граматики” Виконання. [2], [3], [4]. – 2 год.

**Лекція 15.** Перевірка LL(1)- умови для довільної КС- граматики. Побудова LL(1)- таблиці для управління роботою LL(1)- синтаксичним аналізатором. Методика розробки синтаксичних авналідаторів: метод рекурсивного спуску (метод синтаксичних діаграм).

**Завдання для самостійної роботи.** Програмування основних модулів лабораторної роботи. Підготовка тестів. Тестування [2], [3], [4]. – 2 год.

**Лекція 16.** Методи денотаційної та операційної семантики. Метод семантичних підпрограм. Атрибутний метод визначення семантики програм. Синтезовані та успадковані атрибути. Порядок та правила обчислення атрибутів Приклад [2], [3], [4]. – 2 год..

**Завдання для самостійної роботи.** Програмування основних модулів лабораторної роботи. Підготовка тестів. Тестування [2], [3], [4]. – 2 год.

**Лабораторне заняття 8.** Здача лабораторної роботи №5. Ознайомлення з постановкою лабораторної роботи №6: „Програмування алгоритмів побудови таблиць управління синтаксичного аналізатора”. [2], [3], [4]. – 2 год.

**Лекція 17.** Машинно-орієнтовані мови програмування. Асемблери - мовні процесори для машинно-орієнтованих мов програмування. Структура асемблера, проходи та бази даних [4] - [6] – 2 год Приклад [2], [3], [4]. – 2 год.

**Завдання для самостійної роботи.** Програмування основних модулів лабораторної роботи. Підготовка тестів. Тестування [2], [3], [4]. – 2 год.

### **Контрольні запитання до змістового модуля 3.**

1. Типи граматики та відповідні їм автомати.
2. Аналіз  $\pi$ , побудова аналізу  $\pi$  на основі лівосторонньої стратегії виводу.
3. Алгоритм відтворення синтаксичного дерева на основі аналізу  $\pi$ .
4. Означення LL(k)- граматики. Наслідки означення.
5. Алгоритм побудови множини First.
6. Алгоритм побудови множини Follow.
7. Алгоритм побудови таблиці управління LL(1) – синтаксичним аналізатором.
8. Алгоритм LL(1) – синтаксичного аналізатора.
9. Алгоритм побудови таблиці управління LL(k) – синтаксичним аналізатором ( $k>1$ ).
10. Алгоритм LL(k) – синтаксичного аналізатора ( $k>1$ ).
11. Поняття семантичного терму програми. Співвідношення між семантичним та синтаксичним термами програми.
12. Види семантичного терму програми.
13. ПОЛІЗ для конструкцій управління.
14. Перегляди Асемблера. Таблиці управління на першому перегляді Асемблера.
15. Структура об'єктного модуля програми.
16. Оверлейна структура \*.exe – файла. Виконання оверлейного модуля. Управління на рівні операційної системи.
17. Стратегії підвантаження сторінок віртуальної пам'яті в оперативну пам'ять.
18. Стратегії вивантаження сторінок з оперативної пам'яті.
19. Машинно-незалежні алгоритми оптимізації об'єктного коду програми.
20. Машинно-залежні алгоритми оптимізації об'єктного коду програми.



## Рекомендована література

1. Java 2. <https://docs.oracle.com/javase/8/docs/api/> Опис класів Java SE.
2. Java 2. <http://www.oracle.com/technetwork/java/javase/documentation/index.html>
3. Презентаційні матеріали до оглядових лекцій з мови Java.
4. Ахо А. Ульман Дж. Теория синтаксического анализа, перевода и компиляции. Т1. М. Мир. 1978.
5. Грис Д. Построение компиляторов для ЦЭВМ. М. Мир. 1976.
6. Льюис Ф., Стирнз Р., Розенкранц Д. Теоретические основы построения компиляторов. М. Мир. 1979.
7. Вирт Н. Систематическое программирование. Введение. М.Мир. 1977.
8. В.В. Волохов, Б.І. Бойко, В.Ф. Кузенко, С.С. Шкільняк. Методичні рекомендації до лабораторного практикуму побудови мовних процесорів з курсу „Системне програмування” – К. Київський національний університет імені Тараса Шевченка, 2001. – 52 с.

## Додаткова література

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



**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**«Системне програмування», що виносить на самостійну роботу студентів**  
**3 курсу спеціальності «Прикладна математика» у другому семестрі 2017-2018**  
**навчального року**

Напрямок підготовки 6.040301 «Прикладна математика»  
(шифр і назва напрямку підготовки)

Спеціалізація:

- Обчислювальна математика;
- Дослідження операцій;
- Моделювання та оптимізація систем.

**На першому етапі** (24 січня – 6 лютого 2018 р.) студенти мають:

- вивчити запропоновані матеріали, які викладено у презентаційному файлі: Оглядові лекції з мови програмування Java;
- виконати інсталяцію на власному ПЕОМ (Desktop або Note) IDE (Integrated Development Environment) – середовища розробки програмного забезпечення з використанням мови програмування Java;
- Ознайомитися з технологією виконання робіт з використанням встановленого IDE-середовища.

Вище перераховані роботи необхідно виконати до **6 лютого 2018 р.**

**На другому етапі** (7 лютого – 20 лютого 2018 р.) кожен студент має опанувати питання винесені на самостійну роботу на поглибленому рівні. Підтвердженням його роботи в межах цього етапу є виконання лабораторної роботи виконаної протягом другого етапу. Загальну умову лабораторної роботи наведено у **п.4 Додатку 1.**

### ***Зміст***

***Змістовий модуль 1. Об'єктно-орієнтоване програмування. Оглядові лекції з мови програмування Java.***

**Тема 1.** Поняття об'єкта в мові програмування Java SE 8. Структура об'єкта. Відкрита та закрита компоненти об'єкта. Методи - члени об'єкта. Опис та означення функцій - членів об'єкта. Визначення та опис методів класу. Властивості методів у залежності від атрибутів, які їм надаються. Приклади. (7 год.). (4 год.).

Поняття класу як приватного типу . Об'єкт типу. Синтаксис визначення типу та об'єкта типу. Конструктори. Метод finalize(). Методи класу. Сигнатура методу. Приклади [1].

**Тема 2.**

Перевантаження методів. Роль суперкласу при перевантаженні та перевизначенні методів. Приклади. [1] (7 год.).

**Тема 3.**

Виключення та їх обробка. Ієрархія виключень. Побудова системи власних виключень. Приклади [1]. (5 год.).

**Тема 4.** Колекції. Інтерфейси колекцій та відповідна ієрархія класів. Клонування та серіалізація об'єктів. Приклади[1] (7 год.).

**Тема 5.** Узагальнені масиви. Об'єктні оболонки. Поняття рефлексії. Роль JVM при реалізації рефлексії. Методи рефлексії. Приклади [1]. (6 год.).

### Контрольні запитання до змістового модуля 1.

1. Поняття класу. Синтаксис визначення класу. Синтаксис визначення методу класу.
2. Відкрита та закрита частина класу. Конструктори класу. Створення класів без можливостей створення об'єктів відповідного класу.
3. Статичні класи. Статичні методи класів.
4. Адресна арифметика. Використання адресної арифметики при проектування методів класу.
5. Терміни `overload` та `override`: перевантаження методів та їх переписування.
6. Базові та похідні класи.
7. Абстрактні класи. Поліморфізм.
8. Конструктори для похідних класів.
9. Поняття виключення. Синтаксис обробки виключення.
10. Проектування власних виключень та їх використання.
11. Поняття інтерфейсу у мові Java.

### Типове завдання модульної контрольної роботи №1.

1. Внутрішні локальні класи. Синтаксис. Приклади.
2. Класи. Визначення класу. Базовий клас та інтерфейси.
3. Не використовуючи інших методів (окрім конструктора, довжина рядка та доступ до *i*-го символу ) запрограмуйте наступні методи класу `String` :
  - 3.1. Boolean `startsWith(String prefix, int toffset)`  
Метод повертає `true`, якщо `prefix` є префіксом рядка. Що починається з позиції `toffset`, інакше - `false`.
  - 3.2. int `compareTo(String anotherString)` //лексикографічне порівняння.  
Метод повертає 0 - рядки рівні, >0 - перший рядок «більший», <0- перший рядок «менший»
  - 3.3. int `indexOf(String str)`  
Метод повертає індекс першого входження `str` у рядок.
4. Програміст визначив список об'єктів наступним чином (на перспективу):

```
ArrayDeque <Object> testDeque;
```

Фактично в цей список були записані екземпляри класів `Date` та екземпляри інших класів. Визначте конструкцію управління для обробки лише даних типу `Date`.

### Додаток 1

1. Презентація курсу лекцій #1-5 на сайті факультету <http://csc.knu.ua/uk/filer/canonical/1516622226/582/> або за запитом на адресу: [yvn@uiccnnet.com.ua](mailto:yvn@uiccnnet.com.ua).
2. <https://docs.oracle.com/javase/tutorial/java/index.html>. Оригінальний курс по мові Java.
3. <http://www.oracle.com/technetwork/java/javase/downloads/index.html>: Java SE Downloads and NetBeans with JDK 8.
4. Лабораторний практикум: <http://tp.infosoft.ua/sp/lr1.htm>.  
Студент навчальної групи виконує лабораторну роботу за номером = (номер у списку групи) % 12 +1, тобто перший у списку виконує роботу №2, другий - №3, .... Дванадцятий - №1 і т. д.