

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Прикладне програмування Java»

Лектор курсу			Ільїн Олег Олександрович, доктор технічних наук, професор кафедри комп'ютерних наук		Контактна інформація лектора (e-mail), сторінка курсу в Moodle		e-mail: oleg.ilin@duikt.edu.ua сторінка курсу в Moodle – https://dn.dut.edu.ua/course/view.php?id=853	
Галузь знань			12 «Інформаційні технології»		Рівень вищої освіти		бакалавр	
Спеціальність			122 Комп'ютерні науки		Семестр		1, 2	
Освітня програма			Штучний інтелект		Тип дисципліни		Обов'язкова	
Обсяг:	Кредитів ECTS	Годин	За видами занять:					
			Лекцій	Семінарських занять	Практичних занять	Лабораторних занять	Самостійна підготовка	
	10	300	72	-	90	-	138	
АНОТАЦІЯ КУРСУ								
Взаємозв'язок у структурно-логічній схемі								
Освітні компоненти, які передують вивченню			<ol style="list-style-type: none"> 1. Групова динаміка і комунікації 2. Психолінгвістика 3. Іноземна мова 4. Дискретні структури 5. Вища математика 6. 3D-моделювання та дизайн 7. Застосування інформаційно-комунікаційних технологій 					
Освітні компоненти для яких є базовою			<ol style="list-style-type: none"> 1. Бази даних 2. Об'єктно-орієнтовані методи розробки веб-додатків 3. Алгоритмізація та програмування 4. Інформаційна безпека комп'ютерних систем 					
Мета курсу:	Отримання теоретичних знань і практичних навичок програмування на мові Java.							
Компетенції відповідно до освітньої програми								
Soft- skills / Загальні компетентності (ЗК)					Hard-skills / Спеціальні компетентності (СК)			
ЗК1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК2. Здатність застосовувати знання у практичних ситуаціях. ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово. ЗК5. Здатність спілкуватися іноземною мовою. ЗК6. Здатність вчитися й оволодівати сучасними знаннями. ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.					СК8. 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.			

<p>ЗК8. Здатність генерувати нові ідеї (креативність).</p> <p>ЗК9. Здатність працювати в команді.</p> <p>ЗК11. Здатність приймати обґрунтовані рішення.</p> <p>ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p> <p>ЗК13. Здатність діяти на основі етичних міркувань.</p> <p>ЗК14. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.</p> <p>ЗК15. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.</p>	<p>СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.</p> <p>СК16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.</p>
---	--

Програмні результати навчання (ПР)

<p>ПР1. 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.</p> <p>ПР9. 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.</p> <p>ПР16. Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.</p>
--

ОРГАНІЗАЦІЯ НАВЧАННЯ

Тема, опис теми	Вид заняття	Оцінюван ня за тему	Форми і методи навчання/питання до самостійної роботи
-----------------	-------------	------------------------	--

Розділ (модуль) 1. Основи програмування на мові Java

<p>Тема 1. Робота з Eclipse. Перша програма.</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Призначення мов програмування 2. Особливості мови Java 3. Ознайомлення із поняттями клас, пакет, проект. 4. Структура програми на Java. Перша програма "Hello, World!" 5. Оголошення змінних 6. Типи даних String, int і double 7. Перетворення (casting) типів int та double
--

8. Конкатенація String

Вміти:

1. Встановлювати JDK, IDE Eclipse
2. Порядок запуску програми з командного рядку та через IDE Eclipse
3. Способи детектування, виявлення та виправлення похибок
4. Виконувати структуру програми на Java. Код першої програми "Hello, World!"
5. Оформлювати код та коментарі згідно code convention.
6. Оголошувати змінні
7. Робити претворення типів int та double

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 1.1 Ведення до мови програмування Java	Лекція 1 2 год		Лекція-візуалізація
Заняття 1.2 Створення та запуск програм на Java	Практичне заняття 1 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 1. Загальні принципи розробки програм, в тому числі і на Java	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 2. Представлення чисел у двійковій, шістнадцятковій та вісімковій системах числення.

Знати:

1. Системи числення: двійкова система числення, вісімкова система числення і шістнадцяткова система числення.
2. Принцип роботи з двійковими числами, сфери їх застосування.
3. Застосування в програмуванні, особливо у взаємодії з бітами та байтами.

Вміти:

1. Конвертувати дані з однієї системи числення в іншу.
2. Застосовувати метод toBinaryString() з класу Integer для перетворення числа у дійковий рядок.
3. Перетворювати двійковий рядок у число за допомогою методу parseInt() з класу Integer.

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 2.1 Представлення чисел у двійковій, шістнадцятковій та вісімковій системах числення	Лекція 2 2 год		Лекція-візуалізація
Заняття 2.2 Застосування бінарних чисел у програмах	Практичне заняття 2 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 2. Бінарні числа	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 3. Операції порівняння. Типи даних boolean. Оператори if та if-else. Логічні функції.

Знати:

1. Типи розгалужень. Програмні конструкції для розгалужень у Java
2. Операції порівняння. Приклади застосування
3. Логічні операції. Приклади застосування
4. Поняття потоку виконання програми. Блок-схема прямолінійного потоку та потік із розгалуженнями.
5. Типи розгалужень. Програмні конструкції для розгалужень у Java
6. Конструкція для умовного переходу if, if-else. Синтаксис та принцип застосування. Приклади застосувань: проста імітація випадкового процесу, сортування двох та трьох чисел

Вміти:

1. Здійснювати логічні операції у відповідних виразах
2. Здійснювати операції порівняння
3. Використовувати конструкцію розгалуження потоку виконання if, if-else.
4. Використовувати фігурні дужки для формування багаторядкових блоків коду
5. Використовувати генератор випадкових чисел
6. Будувати програми, що імітують прості випадкові процеси (підкидування монетки, грального кубіку тощо)

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 3.1 Логічні операції та операції порівняння. Розгалуження потоку виконання програми.	Лекція 3 2 год		Лекція-візуалізація
Заняття 3.2 Здійснення операцій порівняння, логічних операцій, конструкцій розгалуження потоку виконання у програмах на Java	Практичне заняття 3 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

Тема 3. Базові конструкції розгалуження потоку виконання програми. Принципи застосування в програмах імітаційного моделювання	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 4. Цикли. Оператор for. Область видимості змінної. Вкладені цикли. Оператори while</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Конструкція для циклічного виконання коду операторів for, while і do-while. Синтаксис та принцип застосування. Приклад застосувань: генерація ступенів двійки 2. Область видимості змінної 3. Особливості використання фігурних дужок в кодї Java для створення багаторядкових блоків коду 4. Конструкція вкладених циклів <p>Вміти:</p> <ol style="list-style-type: none"> 1. Використовувати конструкції для циклічного виконання коду for, while і do-while 2. Застосовувати вкладені цикли в програмах 3. Будувати програми для обчислення значень функцій для аргументів на певному інтервалі та виведення результату у табличному вигляді у консоль *(на прикладі розрахунку ступенів двійки) <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p><i>Рекомендовані джерела: 1-7</i></p>			
Заняття 4.1 Основи Циклів в Java: синтаксис циклів for, while та do-while	Лекція 4 2 год		Лекція-візуалізація
Заняття 4.2 Застосування конструкцій циклів для вирішення програмних завдань на Java	Практичне заняття 4 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 4. Використовувати циклічні конструкції для ефективного вирішення завдань, зокрема для виведення крапкових трикутників та вирішення математичних задач, що можуть виникнути у шкільних підручниках	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 5. Види масивів, способи їх опису. Одновимірні масиви та двовимірні. Поле length. Обробка масивів</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Поняття структури даних. Поняття масиву 2. Математичні абстракції та аналоги в мові програмування Java. Масив як математична множина 3. Порівняння ефективності різних способів обробки множини значень одного типу в кодї Java 			

4. Підтримка масивів в Java. Представлення масиву у пам'яті. Особливості копіювання масивів в Java
5. Розв'язання задач із одновимірними та двовимірними масивами. Типові шаблони застосування
6. Типові помилки при роботі з масивами, шляхи виявлення та подолання в Java
7. Способи оброблення масивів
8. Функція поля length

Вміти:

1. Створювати масиви різної довжини, ініціалізувати дані під час декларування
2. Застосовувати масив у програмах для збереження множин значень, в т.ч. компонентів векторів
3. Працювати із аргументами командного рядка, які зберігаються у масиві
4. Заповнювати масив випадковими значеннями із певного діапазону
5. Обробляти значення масиву у циклі за різними математичними формулами (в т.ч. векторні операції)
6. Копіювати масив в інший масив, розбивати масив на кілька масивів, копіювати масиви з кількох в один
7. Виводити елементи масиву у консоль в рядок, в стовпчик
8. Знаходити мінімальний/максимальний елемент у масиві
9. Бачити похибки у роботі із масивами, виявляти їх у коді та виправляти
10. Знаходити довжину масиву

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 5.1 Введення у масиви. Створення та застосування	Лекція 5 2 год		Лекція-візуалізація, експрес-опитування студентів
Заняття 5.2 Застосування одновимірних масивів	Практичне заняття 5 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 5. Масиви. Базові операції та шаблони застосування одновимірних масивів	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 6. Оператори ділення за модулем %, інкремента, декремента, операцій з присвоєнням, тернарний оператор, логічна функція XOR, short-circuit логічні функції

Знати:

1. Функцію оператора ділення за модулем %
2. Поняття інкременту та декремента
3. Поняття операцій з присвоєнням

4. Синтаксис та застосування тернарного оператора
5. Дію логічного оператора XOR
6. Поняття short-circuit

Вміти:

1. Застосовувати оператор ділення за модулем
2. Впроваджувати інкремент та декремент в залежності від умови задачі
3. Виконувати операції з присвоюванням
4. Перетворювати умовний оператор if-else в тернарний
5. Імплементувати логічний оператор XOR

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 6.1 Застосування оператору switch-case, циклу for-each, операторів з присвоєнням, тернарного оператора та ключових слів break і continue	Лекція 6 2 год		Лекція-візуалізація
Заняття 6.2 Застосування оператору switch-case, циклу for-each, операторів з присвоєнням, тернарного оператора та ключових слів break і continue	Практичне заняття 6 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 6. Виконання завдань за допомогою switch-case, обробка масиву через цикл for-each	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 7. Оператори switch-case, break, continue, for-each

Знати:

1. Область видимості, синтаксис та застосування оператору switch-case
2. Значення ключових слів break і continue
3. Застосування циклічного оператора for-each для обробки масиву

Вміти:

1. Застосовувати оператор switch-case
2. Додавати ключові слова break та continue в блоках коду
3. Обробляти масив за допомогою оператора for-each

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 7.1 Оператори switch-case, break, continue, for-each	Лекція 7 2 год		Лекція-візуалізація
Заняття 7.2 Опрацювання оператора switch-case, ключових слів break, continue і циклу for-each	Практичне заняття 7 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 7. Оператори switch-case, break, continue, for-each	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 8. Ознайомлення із Java API , пакетом java.lang, класом String та методами length, toLowerCase, toUpperCase, charAt, endsWith, startsWith класу String. Ознайомлення із Java API та методами indexOf, replace, split, substring класу String</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Застосування імпортів для роботи з іншими пакетами 2. Застосування методів класу java.util.Arrays під час роботи з масивами. 3. Методи toString, sort, binarySearch класу java.util.Arrays 4. Декларування класів java.util.ArrayList та java.util.LinkedList 5. Поняття алгоритму динамічного масиву та зв'язаного списку 6. Переваги ArrayList і LinkedList над статичним масивом 7. Методи size, add, get, remove, indexOf, contains класу ArrayList. 8. Методи peek, peekFirst, peekLast, addFirst і addLast класу LinkedList. <p>Вміти:</p> <ol style="list-style-type: none"> 1. Імпортувати пакети 2. Здійснювати сортування масиву методом sort з класу java.util.Arrays та реалізовувати бінарний пошук через метод binarySearch 3. Оголошувати та ініціалізовувати ArrayList та LinkedList 4. Використовувати динамічний масив для більш ефективного написання коду <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 8.1 Особливості застосування масивів у задачах комп'ютерних наук	Лекція 8 2 год		Лекція-візуалізація
Заняття 8.2 Застосування масивів у задачах комп'ютерних наук	Практичне заняття 8 4 год	4	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

Тема 8. Використання одномірних масивів у класичних задачах комп'ютерних наук	Самостійна робота	2	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 9. Поняття інкапсуляції, структура класу. Створення власних класів. Оголошення та ініціалізація полів. Оголошення методів. Оголошення та створення об'єктів. Оператор new. Літерал null. Архітектурний шаблон Model-View-Controller (MVC) для побудови власних класів. Ознайомлення із конструкторами, передаванням в метод посилань на об'єкти</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Поняття об'єкту та класу, UML-діаграма класів 2. Поняття інкапсуляції, структура класу (поля та методи), область їх видимості 3. Змінна об'єктного типу 4. Поняття та призначення конструкторів, їх різновиди 5. Призначення модифікаторів видимості 6. Структуру і компоненти методів, як оголошувати методи <p>Вміти:</p> <ol style="list-style-type: none"> 1. Розробляти клас певної сутності на Java, UML 2. Створювати об'єкти за допомогою конструкторів, створювати власні конструктори 3. Ініціалізувати конструктори 4. Реалізовувати архітектурний шаблон MVC при проектуванні класів 5. Звертатися до методів через об'єкт або через самий клас <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 9.1 Класи та об'єкти	Лекція 9 2 год		Лекція-візуалізація, експрес-опитування студентів
Заняття 9.2 Розробка класів та створення об'єктів	Практичне заняття 9 4 год	4	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 9. Класи та об'єкти	Самостійна робота	2	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 10. Ключове слово static. Змінні об'єкту та класу.			

Знати:

1. Як оголошуються статичні методи і змінні
2. Як викликаються статичні методи і змінні
3. Як оперувати змінними або методами, якщо вони статичні

Вміти:

1. Звертатися до статичних або нестатичних методів і полей
2. Аналізувати пріоритетність надання елементам статичності або нестатичності

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 10.1 Ключове слово “static”. Змінні об’єкту та класу	Лекція 10 2 год		Лекція-візуалізація
Заняття 10.2 Присвоєння статичності методам і змінним	Практичне заняття 10 2 год	2	Усне опитування, навчальна дискусія, розв’язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 10. Ключове слово “static”. Змінні об’єкту та класу	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 11. Метод main(). Аргументи в методі main()

Знати:

1. Синтаксис методу main()
2. Аргументи командного рядка
3. Перевірка наявності аргументів
4. Обробка аргументів
5. Робота з командним рядком (Command Line Interface, CLI)

Вміти:

1. Декларувати метод main()
2. Розуміти те, що args є масивом рядків (String[]), в якому можна передавати аргументи командного рядка при запуску програми
3. Перевіряти наявність переданих аргументів, щоб уникнути помилок під час їх обробки
4. Навички обробки переданих аргументів та взаємодії з ними в коді програми
5. Валідувати та перетворювати аргументи з рядкового представлення в відповідні типи даних (наприклад, з рядка в ціле число або число з плаваючою комою)
6. Можливість виводити повідомлення про наявність аргументів, їх кількість та зміст для зручності користувача або для дебагування

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 11.1 Особливості передавання в метод main аргументів та їх обробка у кодї	Лекція 11 2 год		Лекція-візуалізація
Заняття 11.2 Запуск та передача аргументів в програму з програмного рядка	Практичне заняття 11 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 11. Перетворення значень аргументів у інші типи у програмі	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 12. Особливості передавання в метод в якості аргументів примітивів та посилань Знати: 1. Передачу примітивів (за значенням) 2. Передачу посилань на об'єкти 3. Передачу аргументів в статичні методи 4. Масиви в якості параметру 5. Внесення параметрів в конструктор для ініціалізації об'єкту Вміти: 1. Передавати типи посилань та примітиви в якості аргументів в методи 2. Передавати параметри в конструктор для ініціалізації об'єкту 3. Передавати параметри різних типів в методи Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 12.1 Особливості передавання в метод в якості аргументів примітивів та посилань	Лекція 12 2 год		Лекція-візуалізація
Заняття 12.2 Створення методів з передаванням аргументів в параметри	Практичне заняття 12 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

Тема 12. Особливості передавання в метод в якості аргументів примітивів та посилань	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 13. Типи методів. Конструктори. Перезавантажені конструктори</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Нестатичні методи 2. Статичні методи 3. Методи з параметрами, додавання параметрів в методи 4. Методи, що повертають значення 5. Методи без повернення значень (void) <p>Призначення та основні характеристики конструкторів.</p> <ol style="list-style-type: none"> 7. Конструктор з параметрами, додавання параметрів в конструктор 8. Процес перезавантаження конструкторів 9. Використання ключого слова this для виклику конструктора з іншого конструктора <p>Вміти:</p> <ol style="list-style-type: none"> 1. Оголошувати методи, визначати тип повернення і додавати параметри 2. Створювати базові конструктори (без параметрів) та конструктори з параметрами 3. Розрізняти конструктори від інших методів в класі 4. Визначати та використовувати різні конструктори в класах з різною кількістю та типами параметрів 5. Визначати та використовувати статичні і нестатичні методи в класах <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p><i>Рекомендовані джерела: 1-7</i></p>			
Заняття 13.1 Типи методів. Конструктори. Перезавантажені конструктори	Лекція 13 2 год		Лекція-візуалізація, експрес-опитування студентів
Заняття 13.2 Створення класів з методами і конструкторами, виконання операції перезавантаження	Практичне заняття 13 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 13. Типи методів. Конструктори. Перезавантажені конструктори	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 14. Перезавантаження методів

Знати:

1. Поняття перевантаження методів: принцип реалізації на прикладі програм для перетворення чисел з hex у dec системи числення, пошуку максимального значення
2. Ситуація неоднозначного виклику методів: причина та шляхи вирішення
3. Переваги використання методів
4. Принципи розробки програм із методами: підходи top-down та bottom-up
5. Поняття локальної змінної у Java, область її видимості

Вміти:

1. Здійснювати повернення значень із методу і складних випадках (за допомогою switch-case та кількох операторів return)
2. Реалізовувати перевантажені методи, виявляти таку необхідність відповідно до поставлених задач
3. Розширювати та звужувати область видимості змінних у методах, не допускати повторного декларування
4. Розробляти програми із методами способами top-down або bottom-up

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 14.1 Особливості розробки методів, їх перезавантаження	Лекція 14 2 год		Лекція-візуалізація
Заняття 14.2 Застосування блоків ініціалізації, створення перезавантаження методів	Практичне заняття 14 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 14. Перезавантаження методів	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 15. Ініціалізація змінних. Блоки ініціалізації. Порядок ініціалізації

Знати:

1. Знати основні способи ініціалізації змінних в Java
2. Концепцію блоків ініціалізації
3. Способи ініціалізації змінних
4. Порядок виконання конструкторів, блоків ініціалізації та інших частин коду при створенні об'єкта класу
5. Життєвий цикл локальних змінних
6. Ініціалізація глобальних змінних

Вміти:

<p>1. Розрізняти ініціалізацію статичних та нестатичних змінних</p> <p>2. Використовувати блоки ініціалізації для складних ініціалізаційних дій, які не можуть бути виконані в одному рядку коду</p> <p>3. Використовувати блоки ініціалізації для ініціалізації змінних або виконання інших дій під час завантаження класу в пам'ять</p> <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 15.1 Ініціалізація змінних. Блоки ініціалізації. Порядок ініціалізації	Лекція 15 2 год		Лекція-візуалізація
Заняття 15.2 Застосування блоків ініціалізації	Практичне заняття 15 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 15. Ініціалізація змінних. Блоки ініціалізації. Порядок ініціалізації	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 16. Пакети. Import. Static import. Клас Math</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Принцип застосування математичних функцій з бібліотеки Math 2. Поняття пакет 3. Імпорт класів з пакетів 4. Статичний імпорт методів, використання статичних методів класу без повторного вказування його імені 5. Комбінування імпорту та статичного імпорту 6. Загальні конвенції при створенні пакету <p>Вміти:</p> <ol style="list-style-type: none"> 1. Створювати власні пакети 2. Декларувати пакет у верхній частині файлу класу 3. Імпортувати клас з іншого пакету для використання його в поточному класі 4. Визначити ситуації, коли статичний імпорт може бути зручним <p>Застосовувати математичні функції із бібліотеки Math</p> <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 16.1 Пакети. Import. Static import. Клас Math	Лекція 16 2 год		Лекція-візуалізація

Заняття 16.2 Імпортування пакетів, статичний імпорт, виконання математичних завдань за допомогою класу Math	Практичне заняття 16 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 16. Пакети. Import. Static import. Клас Math	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 17. Поняття наслідування. Відношення has-a, is-a. UML-діаграми. Класи Calendar, GregorianCalendar, Date</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Поняття суперкласу і підкласу 2. Ключове слово super, створення об'єктів із класів-нащадків 3. Поняття перевизначення методів: суть та механізм застосування, відмінність від перевантаження 4. Відношення has-a та is-a 5. Класова діаграма 6. Працювання з датами та часом в Java класи: Calendar, GregorianCalendar, Date 7. Обмежувати наслідування <p>Вміти:</p> <ol style="list-style-type: none"> 1. Проектувати ієрархію наслідування, розробляти вміст суперкласу та класів – нащадків, зображати класи та зв'язки у вигляді UML-діаграми 2. Використовувати ключове слово super 3. Обмежувати області видимості у класах-нащадках за допомогою модифікаторів видимості protected 4. Обмежувати наслідування та перевизначення методів за допомогою ключового слова final 5. Перевизначають методи toString(), equals() 6. Будувати UML-діаграми 7. Використовувати методи для працювання з датами та часом <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 17.1 Поняття наслідування	Лекція 17 2 год		Лекція-візуалізація
Заняття 17.2 Поняття наслідування. Відношення has-a, is-a. UML-діаграми. Класи Calendar, GregorianCalendar, Date	Практичне заняття 17 4 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

Тема 17. Поняття наслідування. Відношення has-a, is-a. UML-діаграми. Класи Calendar, GregorianCalendar, Date	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 18. Модифікатори доступу Знати: 1. Завдання модифікаторів доступу 2. Класові модифікатори доступу 3. Комбінування модифікаторів доступу 4. Вплив модифікаторів на наслідування Вміти: 1. Використовувати для визначення видимості класів, методів, полів і інших елементів програми 2. Використовувати комбінації модифікаторів, наприклад, public з final, private з static, тощо. Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 18.1 Модифікатори доступу	Лекція 18 2 год		Лекція-візуалізація
Заняття 18.2 Застосування модифікаторів доступу	Практичне заняття 18 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 18. Модифікатори доступу	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Розділ (модуль) 2. Особливості об'єктно-орієнтованого програмування на мові Java			
Тема 19. Ключове слово this Знати: 1. Вказання на поточний об'єкт 2. Уникнення конфліктів імен 3. Виклик конструктора 4. Передачу поточного об'єкта			

<p>5. Ланцюговий виклик методів 6. Використання в анонімних класах 7. Використання власного поточного об'єкта 8. Обмеження використання Вміти: 1. Звертатися до полів та методів класу 2. Викликати конструктори через ключове слово this() 3. Передавати поточний об'єкт Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 Рекомендовані джерела: 1-7</p>			
Заняття 19.1 Ключове слово this	Лекція 19 2 год		Лекція-візуалізація
Заняття 19.2 Застосування this в ООП	Практичне заняття 19 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 19. Ключове слово this	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 20. Ключове слово super. Клас Object Знати: 1. Виклик конструктора батьківського класу 2. Методи класу Object: toString(), equals(), hashCode(), getClass(), wait(), notify(), notifyAll() 3. Використання в методах Вміти: 1. Використовувати super() для виклику конструктора батьківського класу в конструкторі підкласу 2. Використовувати super для взаємодії з методами батьківського класу в методах підкласу 3. Використання та перевизначення методів класу Object, таких як toString(), equals(), hashCode(), getClass(), wait(), notify(), notifyAll() 4. Використання методу finalize() для виконання операцій перед знищенням об'єкта збиральником сміття Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 Рекомендовані джерела: 1-7</p>			
Заняття 20.1 Ключове слово super. Клас Object	Лекція 20		Лекція-візуалізація

	2 год		
Заняття 20.2 Реалізація ключового слова super. Використання методів класу Object	Практичне заняття 20 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 20. Ключове слово super. Клас Object	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 21. Перевизначення методів. Метод toString() Знати: 1. Розуміння концепції перевизначення методів 2. Сигнатура перевизначеного методу 3. Анотація @Override 4. Роль методу toString() Вміти: 1. Перевизначати методи, використовувати анотацію @Override 2. Перевизначати метод toString() класу Object для отримання "представлення" об'єкту у текстовому вигляді Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 21.1 Перевизначення методів. Метод toString()	Лекція 21 2 год		Лекція-візуалізація
Заняття 21.2 Перевизначення методів. Перевизначення Методу toString()	Практичне заняття 21 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 21. Перевизначення методів. Метод toString()	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 22. Поліморфні посилання. Поліморфізм. Метод equals()

Знати:

1. Поліморфні посилання
2. Характеристика поліморфізму
3. Перевірка типу (instanceof)
4. Кастинг посилань батьківського класу на посилання його підкласу
5. Метод equals()
6. Перевизначення методу equals()

Вміти:

1. Використовувати поліморфні посилання для зручного виклику методів підкласів через посилання на батьківський клас
2. Перевіряти тип об'єкта перед використанням оператора instanceof та кастуванням
3. Перевизначати метод equals() для порівняння об'єктів на рівність відповідно до конкретних вимог класу
4. У перевизначеному equals(), перевіряти на null та клас об'єкта для коректного порівняння

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 22.1 Поліморфні посилання. Поліморфізм. Метод equals()	Лекція 22 2 год		Лекція-візуалізація
Заняття 22.2 Розробка ієрархії класів на основі принципу успадкування та створення відповідних об'єктів із застосуванням принципу поліморфізму	Практичне заняття 22 2 год	3	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 22. Наслідування та поліморфізм	Самостійна робота	3	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 23. Covariant returns

Знати:

1. Поняття Covariant returns (Коваріантні повернення)

Вміти:

1. Використовувати оператор instanceof для перевірки, чи об'єкт є екземпляром певного класу чи інтерфейсу
2. Використовувати коваріантні повернення для забезпечення зручності при використанні методів підкласів через посилання на батьківський клас

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 23.1 Covariant returns	Лекція 23 2 год		Лекція-візуалізація
Заняття 23.2 Виконання Covariant returns	Практичне заняття 23 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 23. Covariant returns	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 24. Приведення типів. Оператор instanceof</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Поняття приведення типів 2. Означення: оператор instanceof 3. Уникання похибки ClassCastException <p>Вміти:</p> <ol style="list-style-type: none"> 1. Опрацьовувати використання явного та неявного приведення типів відповідно до ситуації 2. Імплементувати оператор instanceof для перевірки типу об'єкта перед приведенням 3. Знаходити та обробляти винятки, якщо приведення типів не є можливим 4. Ефективно використовувати приведення типів для роботи з об'єктами різних типів в межах ієрархії <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
Заняття 24.1 Приведення типів. Оператор instanceof	Лекція 24 2 год		Лекція-візуалізація
Заняття 24.2 Реалізація приведення типів у конкретних завданнях. Використання оператора instanceof для визначення типів об'єктів та запобігання помилкам приведення	Практичне заняття 24 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 24. Приведення типів. Оператор instanceof	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 25. Ключове слово final Знати: 1. Поняття ключового слова final Вміти: 2. Використовувати ключового слова final Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 25.1 Ключове слово final	Лекція 25 2 год		Лекція-візуалізація
Заняття 25.2 Дослідження впливу ключового слова final на методи та класи, розглядаючи обмеження змін та наслідування	Практичне заняття 25 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 25. Ключове слово final	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 26. Абстрактні класи Знати: 1. Поняття абстрактного класу, призначення, особливості в кодї 2. Успадкування абстрактного класу 3. Використання абстрактного класу в абстрактних шаблонах Вміти: 1. Використовувати абстрактні класи, коли є спільна логіка для групи класів, але не завжди є зміст створювати об'єкти самого абстрактного класу. 2. Застосовувати абстрактні класи для визначення абстрактних шаблонів, які можуть бути реалізовані в підкласах Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 26.1 Абстрактні класи	Лекція 26 2 год		Лекція-візуалізація
Заняття 26.2 Розробка абстрактних класів	Практичне заняття 26 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

Тема 26. Абстрактні класи	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 27. Інтерфейси. Інтерфейси Comparable, Comparator</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Поняття інтерфейсу, призначення, особливості у кодї 2. Інтерфейси Comparable, Cloneable, Comparator: призначення та принцип використання <p>Вміти:</p> <ol style="list-style-type: none"> 1. Реалізовувати інтерфейси та використовувати їх у класах 2. Визначати метод compareTo() інтерфейсу Comparable, який дозволяє класу здійснювати порівняння самого себе з іншими об'єктами 3. Визначати метод compare() інтерфейсу Comparator, який дозволяє визначити власний порядок порівняння для класів, навіть якщо вони не реалізують Comparable 4. Створювати копії об'єктів за допомогою інтерфейсу Cloneable <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p><i>Рекомендовані джерела: 1-7</i></p>			
Заняття 27.1 Інтерфейси. Інтерфейси Comparable, Comparator. Внутрішні класи (регулярні, локальні, анонімні, статичні)	Лекція 27 2 год		Лекція-візуалізація
Заняття 27.2 Розробка інтерфейсів, перевизначення методів інтерфейсів, внутрішні класи	Практичне заняття 27 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 27. Інтерфейси. Інтерфейси Comparable, Comparator. Внутрішні класи (регулярні, локальні, анонімні, статичні)	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
<p>Тема 28. Внутрішні класи (регулярні, локальні, анонімні, статичні)</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Види внутрішніх класів (звичайні, статичні, анонімні, локальні) та область їх застосування 2. Особливості доступу до членів зовнішнього класу для кожного типу внутрішнього класу <p>Вміти:</p> <ol style="list-style-type: none"> 1. Створювати внутрішні та анонімні класи 2. Розуміти ситуації, в яких кожен тип внутрішнього класу може бути найбільш підходящим 			

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 28.1 Внутрішні класи, анонімні класи та їх застосування	Лекція 28 2 год		Лекція-візуалізація
Заняття 28.2 Написання внутрішніх класів	Практичне заняття 28 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 28. Внутрішні класи, анонімні класи та їх застосування	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 29. Клас Arrays Знати: 1. Клас Arrays та його методи 2. Сортування масивів методом sort() 3. Порівняння масивів equals() 4. Бінарний пошук binarySearch() Вміти: 1. Сортувати, порівнювати, копіювати та виконувати інші операції з масивами за допомогою класу Arrays 2. Використання бінарного пошуку та інших алгоритмів 3. Використовувати методи класу Arrays для роботи з масивами Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 29.1 Клас Arrays	Лекція 29 2 год		Лекція-візуалізація
Заняття 29.2 Опрацювання масиву методами класу Arrays. Вирішення програмних задач пов'язаних з масивами методами Arrays	Практичне заняття 29 2 год	2	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 29. Вирішення програмних задач пов'язаних з масивами методами Arrays	Самостійна робота	1	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

			літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 30. Exceptions Знати: 1. Поняття обробка винятків (exceptions) 2. Ієрархія винятків 3. Ключові слова try, catch, finally 4. Обробка багатьох винятків 5. Використання блоку throws 6. Створення власних винятків 7. Використання блоку try-with-resources Вміти: 1. Використовувати конструкції try, catch та finally для ефективної обробки винятків 2. Обробляти багато типів винятків та використовувати блок throws 3. Створювати та використовувати власні винятки Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 30.1 Exceptions	Лекція 30 2 год		Лекція-візуалізація
Заняття 30.2 Опрацювання Exceptions	Практичне заняття 30 2 год	3	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 30. Створення власних Exceptions	Самостійна робота	3	Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 31. Функціональні інтерфейси та лямбда-вирази

Знати:

1. Визначення функціонального інтерфейсу та його обмежень (один абстрактний метод)
2. Синтаксис лямбда-виразів
3. Обробка винятків в лямбда-виразах

Вміти:

1. Використовувати вбудовані функціональні інтерфейси
2. Визначати лямбда-вирази для реалізації абстрактного методу функціонального інтерфейсу
3. Використовувати функціональні інтерфейси для зручної передачі функціональності
4. Використовувати лямбда-вирази у потоках
5. Застосовувати анотацію `@FunctionalInterface`
6. Створювати власні функціональні інтерфейси з одним абстрактним методом
7. Реалізовувати методи по замовчуванню для додавання функціональності в інтерфейс без порушення умов функціонального інтерфейсу

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 31.1 Функціональні інтерфейси та лямбда-вирази	Лекція 31 2 год		Лекція-візуалізація
Заняття 31.2 Опрацювання технік комбінування лямбда-виразів для створення більш складних функціональних конструкцій	Практичне заняття 31 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 31. Застосування вбудованих функціональних інтерфейсів і їх методів для використання з лямбда-виразами	Самостійна робота		Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 32. Strings, StringBuffer, StringBuilder

Знати:

1. Використання методів для операцій з рядками, таких як `length()`, `charAt()`, `substring()`, `equals()`, `compareTo()`, та інші
2. Означення: `StringBuffer` і `StringBuilder`
3. Поняття мутабільності

Вміти:

1. Використовувати `StringBuilder` або `StringBuffer` для ефективного об'єднання багатьох рядків
2. Використовувати методи `replace()`, `substring()`, `delete()`, `insert()` для модифікації змінюваних рядків
3. Застосовувати `StringBuffer` в багатопоточних сценаріях для безпечної маніпуляції рядками

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 32.1 Strings, StringBuffer, StringBuilder	Лекція 32 2 год		Лекція-візуалізація
Заняття 32.2 Об'єднання рядків та інші функції класів Strings, StringBuffer, StringBuilder	Практичне заняття 32 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 32. Strings, StringBuffer, StringBuilder	Самостійна робота		Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів
Тема 33. Wrappers, autoboxing, autounboxing Знати: 1. Означення: Обгортки (Wrappers) 2. Класи обгортки: Integer, Double, Boolean, Character 3. Означення: Автоматичне упакування (Autoboxing) 4. Означення: Автоматичне розпакування (Autounboxing) Вміти: 1. Використовувати конструкції try, catch та finally для ефективною обробки винятків 2. Створення об'єктів обгортки та робота з їх методами 3. Створювати та використовувати власні винятки Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16 Результати навчання: ПР1, ПР9, ПР16 <i>Рекомендовані джерела: 1-7</i>			
Заняття 33.1 Wrappers, autoboxing, autounboxing	Лекція 33 2 год		Лекція-візуалізація
Заняття 33.2 Практичне використання автоматичного упакування в реальних сценаріях	Практичне заняття 33 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів

<p>Тема 33. Реалізація autoboxing, autounboxing. Розгляд вибору між примітивами та обгортками в залежності від вимог проекту для оптимальності та читабельності коду</p>	<p>Самостійна робота</p>		<p>Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів</p>
<p>Тема 34. Enums</p> <p>Знати:</p> <ol style="list-style-type: none"> 1. Означення Enums 2. Додавання методів до Enums для реалізації додаткової функціональності 3. Використання полів для зберігання додаткової інформації 4. Конструктори Enums 5. Обробка багатьох винятків 5. Використання блоку throws 6. Створення власних винятків 7. Використання блоку try-with-resources <p>Вміти:</p> <ol style="list-style-type: none"> 1. Створювати змінні типу Enums та їх використання в програмі 2. Використовувати цикли для перебору всіх значень Enums 3. Викликати методи та працювати з додатковою функціональністю Enums 4. Користуватися Enums для представлення додаткової інформації та зберігання значень 5. Використання конструкції switch для обробки різних значень Enums <p>Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16</p> <p>Результати навчання: ПР1, ПР9, ПР16</p> <p>Рекомендовані джерела: 1-7</p>			
<p>Заняття 34.1 Enums</p>	<p>Лекція 34 2 год</p>		<p>Лекція-візуалізація</p>
<p>Заняття 34.2 Використання Enums в контексті різних патернів проектування</p>	<p>Практичне заняття 34 2 год</p>	<p>1</p>	<p>Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів</p>
<p>Тема 34. Робота з Enums для збереження та обміну даним</p>	<p>Самостійна робота</p>		<p>Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів</p>

Тема 35. Var-args

Знати:

1. Означення var-args
2. Оголошення var-args
3. Використання var-args в методі
4. Можливість передачі окремих значень
5. Комбінування var-args і інших параметрів
6. Обробка нульових аргументів
7. Var-args і перевантаження методів
8. Обмеження var-args

Вміти:

1. Передавати var-args в параметри методів в якості аргументу
2. Передавати окремі значення разом із varargs

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16

Результати навчання: ПР1, ПР9, ПР16

Рекомендовані джерела: 1-7

Заняття 35.1 Var-args	Лекція 35 2 год		Лекція-візуалізація
Заняття 35.2 Дослідження можливості передавання інших типів даних через var-args для розширення функціональності методів	Практичне заняття 35 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 35. Розглядання різних способів обробки значень var-args, включаючи ітерацію та взаємодію з масивом аргументів	Самостійна робота		Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

Тема 36. Збирач сміття Garbage collector

Знати:

1. Розуміння принципів роботи Garbage collector, його цілей та взаємодії з програмою
2. Типи Збирачів сміття
3. Життєвий цикл об'єкта
4. Пам'ять Java Heap
5. Генерації об'єктів
6. Оптимізація пам'яті
7. Використання блоку try-with-resources

Вміти:

1. Ініціювати збирання сміття та налаштувати параметри Garbage collector в Java
2. Використовувати інструменти для профілювання та моніторингу роботи Garbage collector з метою виявлення можливих проблем
3. Опрацювати ручне видалення об'єктів (System.gc()), використовувати WeakReference та інше

Формування компетенцій: ЗК1, ЗК2, ЗК4, ЗК5, ЗК6, ЗК7, ЗК8, ЗК9, ЗК11, ЗК12, ЗК13, ЗК14, ЗК15, СК8, СК14, СК16
 Результати навчання: ПР1, ПР9, ПР16
Рекомендовані джерела: 1-7

Заняття 36.1 Збирач сміття Garbage collector	Лекція 36 2 год		Лекція-візуалізація
Заняття 36.2 Розгляд різних типів збирачів сміття в Java та їх відмінностей	Практичне заняття 36 2 год	1	Усне опитування, навчальна дискусія, розв'язання задач за допомогою програмного підходу, тренінгові технології, вирішення тестів
Тема 36. Використання інструментів для аналізу та відслідковування роботи Garbage collector в реальному часі	Самостійна робота		Самостійне поглиблення теоретичних знань згідно переліку рекомендованої літератури, самостійне написання програм згідно поточних завдань, вирішення тестів

МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

- Мультимедійний проектор;
- Комп'ютерний клас для проведення практичних занять.

ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ

1. Boyarsky Jeanne, Selikoff Scott. OCA: Oracle Certified Associate Java SE8 Programmer I Study Guide: Exam 1Z0-808. – John Wiley & Sons, Inc., Indianapolis, 2015.- 435 p.
2. Java: The Complete Reference, Tenth Edition 10th Edition by Herbert Schildt(Author), Hardcover: 1344 pages Publisher: McGraw-Hill Education; 10th Edition (October 6, 2017).
3. Васильєв О. Програмування мовою java. Л.-Ліра-К, 2022р.
4. Intro to Java Programming, Comprehensive Version (10th Edition) 10th Edition, by Daniel Liang, 1344 pages, Publisher: Pearson; 10 edition , 2014
5. Computer Science: An Interdisciplinary Approach 1st Edition by Robert Sedgewick (Author), Kevin Wayne (Author), Hardcover: 1168 pages, Publisher: Addison-Wesley Professional;
6. Core Java Volume I--Fundamentals (Core Series) 11th Edition by Cay Horstmann (Author), Hardcover: 928 pages Publisher: Pearson; 11th edition (May 15, 2018).
7. Java 8 API <https://docs.oracle.com/javase/8/docs/api/>

ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)

- Курс передбачає роботу в колективі.

- Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики.
- Освоєння дисципліни передбачає обов'язкове відвідування лекцій і практичних занять, а також самостійну роботу.
- Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем лекційних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблене опрацювання за рекомендованою літературою.
- Усі завдання, передбачені програмою, мають бути виконані у встановлений термін.
- Якщо студент відсутній з поважної причини, він презентує виконані завдання під час самостійної підготовки та консультації викладача.
- Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання. Виявлення ознак академічної недоброчесності в практичній (письмовій) роботі студента є підставою для її незарахування викладачем.
- Студент, який спізнився має право бути присутнім на занятті. Студенти мають інформувати старосту про неможливість відвідати заняття.
- Користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням є підставою для незарахування викладачем роботи студента.

КРИТЕРІЙ ТА МЕТОДИ ОЦІНЮВАННЯ

Умовою допуску до підсумкового контролю є виконання всіх практичних робіт і виконання самостійних завдань, які передбачені структурою освітньої компоненти Прикладне програмування-JAVA.

Якщо студента не допущено до складання іспиту, як такого, що не виконав індивідуальний план, йому надається час до перескладання для виконання всіх вимог допуску. Студент має право на два перескладання. При повторному перескладанні екзамену його у студента може приймати комісія, яка створюється директором ННІТ. Оцінка комісії є остаточною. У випадку отримання студентом 0 балів (неприйнятно), що тягне відрахування за невиконання навчального плану.

Оцінювання студентів здійснюється за накопичувальною 100-бальною системою і складається із двох основних оцінкових блоків і розподіляється в певних пропорціях 60 (бали напрацьовані під час вивчення дисципліни – Поточний контроль), 40 (підсумкове оцінювання - Іспит):

Форми контролю	Види навчальної роботи	Оцінювання
ПОТОЧНИЙ КОНТРОЛЬ	<i>Активність на заняттях, у т.ч.:</i>	
	Виконання практичних робіт	40 балів
	Самостійна робота	20 балів
Додаткова оцінка	Участь у наукових конференціях, підготовка наукових публікацій, участь у Всеукраїнських та Міжнародних конкурсах наукових студентських робіт за спеціальністю, створення кейсів тощо.	максимальна оцінка – 10 балів
	Отримання сертифікату про проходження online-курсів (або очних) із програмування на мові Java, згідно тематики курсу	максимальна оцінка – 20 балів
	Отримання сертифікату Oracle Java SE8 Programmer I (II)	максимальна оцінка – 40 балів
ПІДСУМКОВЕ ОЦІНЮВАННЯ	Метою заходу є контроль сформованості практичних навичок та професійних компетентностей, необхідних для виконання професійних обов'язків. Іспит проходить у письмовій формі.	максимальна оцінка – 40 балів

НЯ Екзамен сем.) (1)		
Залік (2 сем.)	Метою заходу є контроль сформованості практичних навичок та професійних компетентностей, необхідних для виконання професійних обов'язків. Залік проходить у письмовій формі.	максимальна оцінка – 40 балів
Курсова робота сем.) (2)	Курсова робота оцінюється згідно критеріїв якості укладання та функціонування програмного коду (макс. 40 балів), відповідності вимогам завдання (35 балів) та відповіді на запитання під час усного захисту (25 балів).	максимальна оцінка – 100 балів

ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ

бали	Критерії оцінювання	Рівень компетентності	Оцінка /запис в екзаменаційній відомості
90-100	Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях. Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та співставляти дані об'єктів діяльності фахівця на основі набутих з даної та суміжних дисциплін знань та умінь. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань проявив вміння самостійно вирішувати поставлені завдання, активно включатись в дискусію, може відстоювати власну позицію в питаннях та рішеннях, що розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується дисципліни, яка вивчається, але виходить за рамки об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень чи складних практичних завдань.	Високий Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і вирішенні практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, що відносяться до дисципліни, яка вивчається.	Відмінно / Зараховано (А)
82-89	Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності. Вміє самостійно виправляти допущені помилки, кількість яких є незначною. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.	Достатній Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни	Добре / Зараховано (В)

75-81	Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях/ рішеннях/ розрахунках не є системними. Знає характеристики основних положень, що мають визначальне значення при проведенні практичних занять, при виконанні індивідуальних / контрольних завдань та поясненні прийнятих рішень, в межах дисципліни, що вивчається.	Достатній Конкретний рівень, за вивченим матеріалом робочої програми дисципліни. Додаткові питання про можливість використання теоретичних положень для практичного використання викликають утруднення.	Добре / Зараховано (C)
64-74	Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядалися з викладачем, але допускає значну кількість неточностей і грубих помилок, які може усунути за допомогою викладача.	Середній Забезпечує достатньо надійний рівень відтворення основних положень дисципліни	Задовільно / Зараховано (D)
60-63	Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних/розрахункових завдань дисципліни. Виконання практичних / індивідуальних / контрольних завдань значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами.	Середній Є мінімально допустимим у всіх складових навчальної програми з дисципліни	Задовільно / Зараховано (E)
35-59	Студент може відтворити окремі фрагменти з курсу. Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими. Цілісність розуміння матеріалу з дисципліни у студента відсутня.	Низький Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	Незадовільно з можливістю повторного складання) / Не зараховано (FX) В залікову книжку не представляється
1-34	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до здачі заліку.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни	Незадовільно з обов'язковим повторним вивченням / Не допущений (F) В залікову книжку не представляється

