

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра моделювання і програмного забезпечення

ЗАТВЕРДЖУЮ

Перший проректор

_____ Владислав ЧУБАРОВ

“ _____ ” _____ 2023 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування на основі Java технологій

(шифр і назва навчальної дисципліни)

Спеціальність _____ **121 – «Інженерія програмного забезпечення»** _____
(шифр і назва напрямку підготовки)

Освітньо-професійна програма _____ **«Інженерія програмного забезпечення»** _____

Факультет _____ **Інформаційних технологій** _____
(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість національних кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольно-модульні роботи (шт)	Курсові роботи	Залік (сем.)	Іспит (сем.)
						Лекції	Лабораторні					
Денна	2	4	90	3	36	18	18	54	2	-	4	-
Всього			90	3	36	18	18	54				
Денна скорочена	1	2	90	3	36	18	18	54	2	-	2	-
Всього			90	3	36	18	18	54				
Заочна	2	4	90	3	12	6	6	78	1	-	4	-
Всього			90	3	12	6	6	78				
Разом			270	9	84	42	42	186				

Кривий Ріг – 2023 рік

Робочу програму навчальної дисципліни «Програмування на основі Java технологій» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 – «Інформаційні технології» зі спеціальності 121 – «Інженерія програмного забезпечення».

Розробники: доцент кафедри МПЗ Котов І.А.

Робоча програма затверджена на засіданні кафедри моделювання і програмного забезпечення

Протокол від «_____» _____ 2023р. №_____

Завідувач кафедри МПЗ доцент, к.п.н. _____ Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від «_____» _____ 2023р. №_____

Голова вченої ради _____ Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від «_____» _____ 2023р. №_____

Гарант ОПП _____ Андрій СТРЮК

ЗМІСТ

1 ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	4
2 МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	5
3 ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	8
4 СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ	10
5 ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ	11
6 ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ	11
7 САМОСТІЙНА РОБОТА	11
8 МЕТОДИ НАВЧАННЯ	13
9 МЕТОДИ КОНТРОЛЮ	14
10 РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ	14
11 ТЕХНІЧНІ ТА ІНФОРМАЦІЙНІ РЕСУРСИ	18
12 ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ	19
13 НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ	21
14 ТЕРМІНОЛОГІЧНИЙ СЛОВНИК	22
15 ЗМІНИ ТА ДОПОВНЕННЯ ДО РОБОЧОЇ ПРОГРАМИ	24
16 РОБОЧИЙ ПЛАН З ДИСЦИПЛІНИ «ПРОГРАМУВАННЯ НА ОСНОВІ JAVA ТЕХНОЛОГІЙ»	25

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, освітній рівень	Характеристика навчальної дисципліни		
		денна форма навчання	денна скорочена форма навчання	заочна форма навчання
Кількість кредитів-3	Галузь знань 12 <u>Інформаційні технології</u> (шифр і назва)	Нормативна		
Модулів-2	Спеціальність: <u>121 Інженерія програмного забезпечення</u> (код та найменування спеціальності)	Рік підготовки		
Змістових модулів-2		2-й	1-й	2-й
Загальна кількість годин-90		Семестр		
	4-й	2-й	4-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 3	Ступінь вищої освіти: <u>бакалавр</u>	Лекції		
		18 год.	18 год.	6 год.
		Практичні, семінарські		
		–	–	–
		Лабораторні		
		18 год.	18 год.	6 год.
		Самостійна робота		
		54 год.	54 год.	78 год.
Вид контролю: залік				

Примітка

Співвідношення кількості годин аудиторних занять до самостійної роботи становить:

для денної форми навчання – 0,67;

для заочної форми навчання – 0,15;

для скороченої форми навчання – 0,67.

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1 **Мета** дисципліни «Програмування на основі Java технологій» – це формування у студентів теоретичних знань і практичних навиків проектування, розробки та відлагодження додатків із застосуванням сучасних технологій об'єктно-орієнтованого програмування на мові програмування Java, що забезпечують вивчення нових концепцій консольного і візуального програмування, та найбільш ефективних технологій створення програмного забезпечення в процесі вирішення прикладних професійних задач.

Мова програмування Java сьогодні є потужним засобом, який дозволяє як моделювати об'єкти реального світу, так і генерувати віртуальні об'єкти в пам'яті комп'ютера. Використання мови програмування Java забезпечує розробку інтерфейсу додатків, орієнтованих на стандарт візуальних оболонок. Технології програмування Java дозволяють розробляти додатки від найпростіших розрахункових алгоритмів до багатокористувацьких програмних комплексів із застосуванням графічного інтерфейсу і засобів мережі Internet.

2.2 **Основні завдання** вивчення навчальної дисципліни «Програмування на основі Java технологій»:

- ознайомити студентів з методами структурного і об'єктно-орієнтованого підходу в мові програмування Java як найбільш поширеними і ефективними методами розробки програмних продуктів;

- знайомство з основними структурами даних і типовими методами обробки цих структур;

- навчити студентів розробці алгоритмів на основі структурного і об'єктно-орієнтованого підходу в мові програмування Java;

- навчити студентів критеріям перевірки правильності побудови класів в мові програмування Java, основним тенденціям в галузі розвитку технологій об'єктно-орієнтованого програмування;

- закріпити навички алгоритмізації та програмування на основі вивчення мови програмування Java;

- прищепити студентам знання способів використання основних візуальних компонентів розробки додатків на мові програмування Java;

- дати досвід розробки власних засобів контролю та управління елементами програмних систем;

- ознайомити студентів з принципами функціонування та управління спеціальними засобами програмування (реалізація багатозадачності, СОМ технології);

- створення практичної бази для вивчення інших навчальних дисциплін.

2.3. Відповідно до освітньої програми дисципліна забезпечує наступні **компетенції**:

Загальні:

- K01. Здатність до абстрактного мислення, аналізу та синтезу;

- K02. Здатність застосовувати знання у практичних ситуаціях;

- K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел;

- K15. Здатність розробляти архітектури, модулі та компоненти програмних систем;

- K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;

- K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення;

- K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
 - K26. Здатність до алгоритмічного та логічного мислення.
- Фахові:
- СК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;
 - СК02 Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
 - СК03 Здатність розробляти архітектури, модулі та компоненти програмних систем;
 - СК04 Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами;
 - СК05 Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу;
 - СК06 Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки);
 - СК10 Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;
 - СК11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення;
 - СК12 Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення;
 - СК13 Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
 - СК14 Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання освітньої програми, яким відповідає дисципліна:

- ПР01 Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- ПР02 Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності;
- ПР05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення;
- ПР07 Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- ПР11 Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;
- ПР13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;

- ПР18 Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних;
- ПР19 Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

2.4. У результаті вивчення навчальної дисципліни здобувачі повинні

Знати:

- прийоми та засоби алгоритмізації обчислювальних задач;
- теоретичні положення курсу, які передують практичному використанню ЕОМ для програмування та використання прикладного ПЗ;
- основні принципи об'єктно-орієнтованого програмування на мові Java;
- призначення та особливості використання засобів Інтегрованих Середовищ Розробки (IDE) для розробки професійних додатків;
- опис класів в мовах програмування Java;
- спадкування, поліморфізм, віртуальні і динамічні методи;
- обробку подій і повідомлень Windows;
- візуальні методи розробки Windows-додатків;
- обробку виняткових ситуацій.

Вміти:

- формалізувати поставлену задачу;
- розробити структури даних для представлення предметної області розв'язуваної задачі;
- розробляти додатки в сучасних середовищах - IDE;
- проектувати ієрархію класів для вирішення конкретних прикладних завдань;
- реалізовувати робочі проекти ієрархії класів на сучасних мовах програмування;
- виготовляти файли допомоги для додатків;
- тестувати і налагоджувати додатки з метою підвищення надійності і ефективності;
- організовувати управління іншими додатками, використовуючи механізми автоматизації, використовувати сучасні готові бібліотеки класів, технології та інструментальні засоби;
- використовувати сучасні інструментальні засоби для підготовки дистрибутивних інсталяційних дисків для програмного виробу;
- виготовляти програмну документацію на програмний виріб.

2.5. Міждисциплінарні зв'язки

При вивченні дисципліни використовуються знання здобувачів з дисциплін «Вища математика», «Дискретні структури», «Основи програмування», «Алгоритми та структури даних», «Архітектура та проектування програмного забезпечення», «Об'єктно-орієнтоване програмування», «Сучасні технології web-програмування», «Офісне програмне забезпечення», «Англійська мова (за професійним спрямуванням)».

Знання, одержані здобувачами при вивченні дисципліни, використовуються при вивченні дисципліни «Програмування на основі Java технологій».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Змістовий модуль 1. Основи мови програмування Java

Тема 1. Введення в мову Java. Історія Java. Призначення і застосування

Введення в мову Java. Історія Java. Призначення і застосування. Компіляція і виконання програми. Особливості. Забезпечення платформ. Області застосування. Основи ООП. Принципи ООП - інкапсуляція. Переваги ООП. Передача параметрів в функції. Спадкування, поліморфізм. Об'єктна модель Java. Клас і об'єкт. Введення в ООП. Базові принципи ООП. Множинне спадкування. Конструктори, методи і поля класів. модифікатори.

Тема 2. Об'єктна модель Java

Об'єктна модель Java. Оголошення класів, методи класу, конструктори, деструктори, вкладені класи, спадкування, поліморфізм (перевантаження методів). Створення екземплярів. Статичні класи і методи. Структура програми. Літерали, ідентифікатори, типи даних і змінні. Бібліотеки класів (пакети).

Тема 3. Технології і платформи Java

Вступ. Огляд технологій і платформи Java. SDK для Java. IDE для розробки на мові Java. Структура програмного проекту. Стандартні пакети Java. Робота в IDE Eclipse. Компіляція та збирання проекту. Введення в середу розробки додатків NetBeans. Установка. Створення і ведення проекту. Файли проекту. найпростіша програма.

Тема 4. Вступ до типів даних в мові Java

Типи даних в мові Java. Прості типи: цілі, речові, символні, логічні. Типи даних Java. Довідкові типи даних. Вирази і оператори. Перетворення типів. Метод main (). Змінні і константи, поля об'єктів і класів. Область видимості.

Тема 5. Структурні типи даних в мові Java

Агрегатні (структуровані) типи: масиви, багатовимірні масиви, рядки. Складні типи даних. Масиви одновимірні, багатовимірні. Рядки. Методи роботи з рядками. Оболонкові класи. Клас Math, його методи та константи.

Тема 6. Основні оператори в мові Java

Оператори: арифметичні, оператор ділення по модулю, інкремент і декремент, побітові оператори, оператори відносини. Керуючі конструкції. Оператори циклів. Оператори розгалуження. Оператори вибору. Оператори переривання, переходу і повернення.

Тема 7. Арифметичні і логічні оператори

Кодування. Аналіз програми: прогалини, коментарі, лексеми. Види лексем: ідентифікатори, ключові слова, літерали. Робота з операторами: оператори присвоювання і порівняння. Арифметичні операції. Логічні оператори. Бітові операції. Логічні оператори, оператор if-в-рядку, пріоритети операторів.

Тема 8. Управляючі структури мови Java

Оператори, які керують виконанням програми. Умовні оператори IF, оператор множинного вибору SWITCH. Логічні оператори циклу WHILE, DO-WHILE, арифметичний оператор циклу FOR, оператор «кома», оператор CONTINUE

Змістовий модуль 2. Практичне програмування засобами мови Java

Тема 9. Строкові типи

Робота з рядками. Строкові типи. Тип (клас) String. Оголошення та створення рядків. Витяг символів. Порівняння рядків. Пошук символу. Копіювання і модифікація строк. Клас StringBuffer. Читання даних з командного рядка. Пошук рядків. Парсинг рядків.

Тема 10. Базові пакети мови Java

Поняття пакету та інтерфейсу. Оператори package і import. Утилітарні пакети Java.Lang і Java.Util (математичні функції, обробка рядків, робота з датою / часом, обробка масивів, генерація випадкових чисел)

Тема 11. Конструювання візуального інтерфейсу

Конструювання візуального інтерфейсу: бібліотека AWT. Візуальні елементи. Методи управління інтерфейсом. Графічний інтерфейс користувача (GUI). Контейнери та компоненти. Бібліотеки AWT і SWING. Використання візуального редактора GUI. Принципи конструювання візуального інтерфейсу.

Тема 12. Методи управління інтерфейсом засобами мови Java

Конструювання візуального інтерфейсу: бібліотека Swing. Візуальні елементи. Методи управління інтерфейсом. Обробка подій. Модель слухача і джерела події. Графіка в Java. Контейнери та компоненти. Бібліотеки AWT і SWING.

Тема 13. Подієва модель функціонування додатків мови програмування Java

Обробка подій в візуальних компонентах на основі бібліотеки Swing (слухачі, обробники, анонімні класи). Обробка подій від клавіатури і миші. Створення і робота з меню. Стандартні діалогові вікна.

Тема 14. Поточна система введення-виведення мови Java

Система введення-виведення мови Java. Поточна модель організації I/O в Java. Класи InputStream і OutputStream. Консольне введення-виведення. Класи для управління консольним введенням-виведенням. Форматоване введення-виведення. Введення і виведення даних в Java. Поточна модель організації I/O в Java. Класи InputStream і OutputStream.

Тема 15. Файловий ввід-вивід

Система введення-виведення мови Java. Файловий ввід-висновок. Класи для управління файловим введенням-виведенням. Потoki введення-виведення при роботі з файлами. Прямий доступ до файлу randomAccessFile. Використання інтерфеса Path для роботи з файлами. Робота з класом Files для операцій над файлами. Канальний і потоковий ввід-висновок в файлах. Робота з атрибутами файлів. Пошук файлів з використанням класу PathMatcher.

Тема 16. Мова Java і Інтернет

Створення Інтернет-додатків на Java. Поняття, структура і функціонування аплету. Інтеграція аплетів і web-ресурсів. Особливості та завдання локалізації програм. Визначення та представлення локалізуємих даних. Читання і установка локалізуємих даних за допомогою об'єкта Locale. Побудова ресурсів. Виклик ресурсів з додатків. Форматування тексту і його локалізація з використанням NumberFormat DateFormat.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	Денна/денна скорочена форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекц.	лаб.	с.р.		лекц.	лаб.	с.р.
1	2	3	4	5	6	7	8	9
Модуль								
Змістовий модуль 1. Основи мови програмування Java								
Тема 1. Введення в мову Java. Історія Java. Призначення і застосування	6	1	1	4	6	-	-	4
Тема 2. Об'єктна модель Java	6	1	2	3	6	1	1	6
Тема 3. Технології і платформи Java	6	2	1	3	6	-	-	5
Тема 4. Вступ до типів даних в мові Java	6	1	1	4	6	-	-	4
Тема 5. Структурні типи даних в мові Java	7	2	2	3	7	-	-	5
Тема 6. Основні оператори в мові Java	6	1	1	4	6	1	1	6
Тема 7. Арифметичні і логічні оператори	5	1	1	3	5	-	-	4
Тема 8. Управляючі структури мови Java	4	1	1	2	4	1	1	6
Разом за змістовним модулем 1:	46	10	10	26	46	3	3	40
Змістовий модуль 2. Практичне програмування засобами мови Java								
Тема 9. Строкові типи	6	1	1	4	6	-	-	3
Тема 10. Базові пакети мови Java	6	1	1	4	6	-	-	6
Тема 11. Конструювання візуального інтерфейсу	5	1	1	3	5	1	1	5
Тема 12. Методи управління інтерфейсом засобами мови Java	6	1	1	4	6	-	-	3
Тема 13. Подієва модель функціонування додатків мови програмування Java	6	1	1	4	6	1	1	5
Тема 14. Поточкова система введення-виведення мови Java	6	1	1	4	6	1	1	6
Тема 15. Файловий ввід-вивід	5	1	1	3	5	-	-	4
Тема 16. Мова Java і Інтернет	4	1	1	2	4	-	-	6
Разом за змістовним модулем 2:	44	8	8	28	44	3	3	38
Разом за семестр:	90	18	18	54	90	6	6	78
Усього годин:	90	18	18	54	90	6	6	78

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин	
		Денна /скорочена ф. н.	Заочна ф. н.
Змістовий модуль 1			
1.	Дослідження основних компонентів інтегрованих середовищ розробки програм (IDE) на мові Java	1	2
2.	Управління програмним проектом на мові Java	2	
3.	Вивчення основних принципів ООП в мові Java	2	
4.	Прості і структурні типи даних в мові Java. Основні оператори мови Java	2	1
5.	Обробка масивів і рядків в мові Java	2	
Разом за модуль 1:		9	3
Змістовий модуль 2			
6.	Система введення-виведення в мові Java	2	1
7.	Використання утилітарних пакетів (Java.Lang і Java.Util) в мові Java	2	1
8.	Конструювання призначеного для користувача інтерфейсу засобами мови Java	2	
9.	Обробка подій в візуальних компонентах додатків на мові Java	2	
10.	Управління базами даних в додатках на мові Java	1	1
Разом за модуль 2:		9	3
РАЗОМ		18	6

7. САМОСТІЙНА РОБОТА

7.1 Загальні положення

На самостійну роботу студентам денної форми навчання відведено 54 години, заочної - 78 годин. Самостійна робота студентів при вивченні дисципліни «Програмування на основі Java технологій» залучає такі складові:

- опрацювання теоретичних основ прослуханого лекційного матеріалу;
- робота з навчальною літературою.
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- підготовка до виконання, а також до захисту лабораторних робіт;
- підготовка до проведення контрольних заходів.

При виконанні самостійної роботи студенти повинні:

- опрацювати матеріал, що винесено на самостійне вивчення;
- скласти блок-схеми алгоритмів і програми для рішення індивідуальних задач;
- оформити звіт з лабораторних робіт.

7.2. Розподіл годин самостійної роботи

№ з/п	Назва теми	Кількість годин		
		денна	денна скорочена	заочна
1	Історія розвитку Java: Браузери; мережеві комп'ютери; платформа Java.	1	1	2
2	Основні версії продукту Java.	1	1	4
3	Методологія об'єктно - орієнтованого програмування	2	2	5
4	Об'єкти: стан; поведінка; унікальність. Класи. Інкапсуляція. Спадкування.	3	3	4
5	Поліморфізм. Типи відносин між класами. Агрегація. Асоціація. Спадкування.	3	3	5
6	Метакласи	3	3	3
7	Переваги ООП. Недоліки ООП.	1	1	4
8	Кодування. Аналіз програми: прогалини, коментарі, лексеми.	2	2	4
9	Види лексем: ідентифікатори, ключові слова, літерали.	2	2	4
10	Робота з операторами: оператори присвоювання і порівняння. Арифметичні операції.	3	3	3
	Підготовка до контрольних заходів (модуль 1)	5	5	-
	Разом за змістовим модулем 1:	26	26	38
11	Логічні оператори. Бітові операції.	2	2	4
12	Змінні. Примітивні і посилальні типи даних: примітивні типи і цілочисельні типи, дробові типи	2	2	2
13	Логічний тип. Довідкові типи: об'єкти і правила роботи з ними. Клас Object. Клас String. Клас Class.	2	2	5
14	Масиви в Java	2	2	3
15	Робота з операторами: оператори присвоювання і порівняння.	2	2	1
16	Арифметичні операції.	1	1	2
17	Логічні оператори.	2	2	4
18	Примітивні і посилальні типи даних: примітивні типи і цілочисельні типи, дробові типи. Логічний тип.	2	2	4
19	Об'єкти і правила роботи з ними	3	3	4
20	Дерево компонентів Java	1	1	2
21	Модель повідомлень. особливості AWT	2	2	4
22	Робота з кольорами, шрифтами, отрисовка графічних, примітивів, менеджери компоновки	2	2	5
	Підготовка до контрольних заходів (модуль 2)	5	5	-
	Разом за змістовим модулем 2:	28	28	40
	Разом:	54	54	78

8. МЕТОДИ НАВЧАННЯ

Методи навчання це упорядковані способи діяльності викладача і студентів, спрямовані на ефективне вирішення навчально-виховних завдань. Методи навчання призначені для процесу навчання, а також як спосіб пізнавальної діяльності студентів з оволодіння знаннями, уміннями і навичками. Використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Навчальна лекція – це логічне, послідовне викладання змісту навчання, яке характеризується судженнями, висновками, підсумком. Вона охоплює основний теоретичний матеріал однієї або кількох тем навчальної дисципліни. Призначенням лекції є формування у здобувачів фундаментальних знань з дисципліни, а також визначає основний зміст і характер усіх інших навчальних занять та самостійної роботи здобувачів із цієї дисципліни.

Лабораторне заняття – форма організації навчання, яку проводять за завданням і під керівництвом НПП. Основні дидактичні цілі – експериментальне підтвердження вивчених теоретичних положень навчальної дисципліни та формування вмінь й навичок їх практичного застосування. Проведення лабораторного заняття ґрунтується на попередньо підготовлених наборах завдань різної складності для розв'язання на занятті. Лабораторне заняття проводиться у навчальних лабораторіях з використанням пристосованого до умов навчального процесу устаткування.

Самостійна робота здобувача є основним способом оволодіння навчальним матеріалом у час, вільний від обов'язкових аудиторних занять. Мета виконання самостійної роботи – поглиблення, узагальнення й закріплення теоретичних знань і практичних умінь здобувачів із дисципліни шляхом вироблення вміння самостійної роботи з навчальною і фаховою літературою та інформацією в мережі Інтернет.

Самостійна робота здобувачів здійснюється у формі: підготовки до лекцій і лабораторних занять, опрацюванні тем, винесених на самостійне опрацювання. Самостійну роботу здобувач може виконувати у бібліотеці, комп'ютерних класах, у домашніх умовах.

Підготовка до лекцій передбачає самостійне опрацювання теоретичного матеріалу. При цьому необхідно звернути увагу на необхідність чіткого засвоєння основних термінів та визначень, розуміння їх змісту, обов'язкового аналізу використання теоретичних положень для розв'язання наданих прикладів.

Самоперевірку засвоєння навчального матеріалу здобувач здійснює за контрольними запитаннями, що надано після кожної теми у конспекті лекцій та іншій літературі, та після кожного лабораторного заняття у відповідних методичних вказівках. Якщо на деякі запитання здобувач не може надати відповіді, то необхідно повторити вивчення навчального матеріалу, або визначити правильну відповідь з викладачем на консультації.

Під час вивчення даної дисципліни використовуються:

- мультимедійні освітні технології: інтерактивні лекції (презентації) із використанням програм MS Power Point, Google Slides у поєднанні з анімацією та звуковим супроводом; перегляд відеороликів за окремими пунктами тем, використання електронних посібників;
- діалогові технології: групові обговорення, використання «мозкового штурму».

Лекції проводяться з використанням технічних засобів навчання й супроводжуються демонстрацією презентацій за допомогою проектора.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист лабораторних робіт проводиться за допомогою Google Meet або Zoom.

9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до практичних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань;
- перевірка підготовки до лабораторних занять.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться під час усної співбесіди зі здобувачами по теоретичним матеріалам, за результатами захисту лабораторних робіт й виконання самостійної роботи. Підсумковим контролем є залік у 4 семестрі для денної форми навчання, у 2 семестрі для денної скороченої форми навчання і залік у 4 семестрі для заочної форми навчання.

10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Для оцінювання успішності студентів використовується модульно-рейтингова система, яка передбачає розподіл балів за виконання всіх запланованих видів робіт.

При цьому максимальна кількість балів, за кожний модуль при умові його бездоганного виконання, дорівнює 100. Ця сума складається з балів, що їх накопичив студент у ході *поточного контролю* та балів, отриманих за *контрольно-модульну роботу* (КМР).

Поточний контроль по модулю 1 проводиться на 8-ому тижні семестру. Поточний контроль по модулю 2 проводиться на 16-ому тижні семестра.

Модульний контроль здійснюється 1 раз за семестр.

Контрольно-модульна робота складається з контрольних питань та завдання.

- оцінювання **контрольних питань** розподіляється пропорційне їх кількості.
- оцінювання **контрольного завдання** виконується за принципами оцінювання лабораторних робіт (вірно виконана та оформлена блок-схема, працююча програма).

10.1. Денна форма навчання

Максимальна кількість балів за модулі при умові їх бездоганного виконання для студентів *денної* форми навчання:

№ з/п	Назва змістового модуля	Максимальна кількість балів
1	Основи мови програмування Java	
	Лабораторні роботи	25
	КМР за 1 змістовий модуль	25
2	Практичне програмування засобами мови Java	
	Лабораторні роботи	25
	КМР за 2 змістовий модуль	25
Всього за семестр		100

Сума складається з балів, що накопичив студент у ході поточного контролю.

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 50-ти відсоткам ваги у семестрі. Бали розподіляються поміж лабораторними роботами модуля у відповідності до їх відносної складності. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється.

Бали за контрольно-модульні роботи складають 50 відсотків відповідно.

Контрольно-модульна робота (КМР) може дати максимально 25 балів при найвищій якості виконання. При зниженні якості КМР знижується і сума балів відповідно до шкали, що наводиться у таблиці:

Шкала оцінювання контрольно-модульних робіт

Відсоток вірних компонентів КМР	0 – 30	31 – 60	61 – 75	76 – 85	86 – 94	95 – 100
Сума балів за КМР	0	5	10	15	20	25

10.2. Заочна форма навчання

Успішність здобувачів-заочників оцінюється аналогічним чином, за 100-бальною шкалою. При чому, лабораторні роботи, які проводяться в аудиторії, оцінюються відносно меншою кількістю балів, ніж роботи, виконані самостійно.

Максимальна кількість балів за модулі при умові їх бездоганного виконання для студентів *заочної* форми навчання:

№ з/п	Назва змістового модуля	Максимальна кількість балів
1	Основи мови програмування Java	
	Лабораторні роботи	10
2	Практичне програмування засобами мови Java	
	Лабораторні роботи	10
КР		80
Всього за семестр		100

Сума складається з балів, що накопичив студент у ході поточного контролю.

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 20-ти відсоткам ваги. Бали розподіляються поміж лабораторними роботами у відповідності до їх відносної складності.

Бали за контрольні роботи складають 80 відсотків. При зниженні якості виконання тієї чи іншої лабораторної роботи, знижується і кількість балів, якою вона оцінюється.

Контрольна робота (КР) може дати максимально 80 балів при найвищій якості виконання. При зниженні якості КР знижується і сума балів відповідно до шкали, що наводиться у таблиці:

Шкала оцінювання контрольних робіт

Відсоток вірних компонентів КР	0 – 30	31 – 60	61 – 75	76 – 85	86 – 94	95 – 100
Сума балів за КР	30	40	50	60	70	80

Шкала оцінювання лабораторних робіт

№ модуля	№ зан.	Вид роботи	Тема	Максимальна кількість балів		
				денна	денна скорочена	заочна
1	1	Лаб. робота 1	Дослідження основних компонентів інтегрованих середовищ розробки програм (IDE) на мові Java	5	5	-
	2,3	Лаб. робота 2	Управління програмним проектом на мові Java	5	5	-
	4,5	Лаб. робота 3	Вивчення основних принципів ООП в мові Java	5	5	5
	6,7	Лаб. робота 4	Прості і структурні типи даних в мові Java. Основні оператори мови Java	5	5	5
	8,9	Лаб. робота 5	Обробка масивів і рядків в мові Java	5	5	-
Контрольно-модульна робота				25	25	-
Разом по модулю				50	50	10
2	10,11	Лаб. робота 6	Система введення-виведення в мові Java	5	5	-
	12	Лаб. робота 7	Використання утилітарних пакетів (Java.Lang і Java.Util) в мові Java	5	5	-
	13,14	Лаб. робота 8	Конструювання призначеного для користувача інтерфейсу засобами мови Java	5	5	5
	15,16	Лаб. робота 9	Обробка подій в візуальних компонентах додатків на мові Java	5	5	-
	17,18	Лаб. робота 10	Управління базами даних в додатках на мові Java	5	5	5
Контрольно-модульна робота / Контрольна робота				25	25	80
Разом по модулю				50	50	90
Разом за семестр				100	100	100

Оцінювання кожної лабораторної роботи ведеться за показниками, наведеними в таблиці:

№ з/п /Критерій оцінювання	Максимальна кількість балів (денна форма/заочна форма)			
	Своєчасність виконання	Правильність виконання	Захист роботи	Всього за роботу
Лабораторна робота № 1	1/-	1/-	3/-	5/-
Лабораторна робота № 2	1/-	1/-	3/-	5/-
Лабораторна робота № 3	1/-	1/2	3/3	5/5
Лабораторна робота № 4	1/-	1/2	3/3	5/5
Лабораторна робота № 5	1/-	1/-	3/-	5/-
Лабораторна робота № 6	1/-	1/-	3/-	5/-
Лабораторна робота № 7	1/-	1/-	3/-	5/-
Лабораторна робота № 8	1/-	1/2	3/3	5/5
Лабораторна робота № 9	1/-	1/-	3/-	5/-
Лабораторна робота № 10	1/-	1/2	3/3	5/5
Всього за семестр	10/-	10/8	30/12	50/20

Під своєчасністю практичного виконання та своєчасністю захисту лабораторної роботи розуміється виконання та захист у тиждень згідно із графіком робіт.

Правильність виконання роботи оцінюється наступним чином:

- робота виконана без зауважень - максимальний бал;
- робота виконана достатньо повно з деякими зауваженнями – дві третини від максимального балу;
- робота виконана не повністю – одна третина від максимального балу;
- при перевірці роботи виявлені грубі помилки – 0 балів.

Захист лабораторної роботи передбачає відповіді на контрольні питання, які представлені у методичних вказівках до виконання лабораторних робіт, а також поставлені викладачем усно. Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів.

Семестровий контроль здійснюється у формі заліку в шостому семестрі для денної та заочної форм навчання, у четвертому семестрі для денної скороченої та заочної скороченої форми навчання; екзамену – відповідно у сьомому і п'ятому семестрах.

У разі виконання студентом усіх видів поточних контрольних заходів залік виставляється студенту на підставі зарахованих балів протягом семестру. Результати заліку оцінюються за 100-бальною шкалою. У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS.

У семестрі, в якому семестровий контроль здійснюється у формі заліку, 100 балів, набрані студентом за результатами поточного контролю, можуть бути оцінені як 100 відсотків семестрового завдання. Результати заліку оцінюються за 100-бальною шкалою.

У семестрі, в якому семестровий контроль здійснюється у формі екзамену, 100 балів, набрані студентом за результатами поточного контролю, складають 60 відсотків, а екзаменаційне завдання – 40 відсотків. Результати екзамену оцінюються за 100-бальною шкалою.

У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS:

Шкала оцінювання

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	A	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	B	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89
	C	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	D	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	E	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	FX	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	F	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29

При наявності у здобувачів результатів неформального навчання за освітнім компонентом «Програмування на основі Java технологій» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Програмування на основі Java технологій Програмування на основі Java технологій».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів:

Тема	Посилання на рекомендовані курси
Основи об'єктно - орієнтованого програмування в Java	https://www.coursera.org/specializations/object-oriented-programming
Основи програмування в Java	https://www.coursera.org/specializations/java-programming
Структури даних в Java	https://www.coursera.org/specializations/java-object-oriented
Структури даних і алгоритми в Java	https://www.coursera.org/specializations/data-structures-algorithms
Проектування програмного забезпечення в Java	https://www.coursera.org/learn/java-programming-design-principles
Масиви і списки в Java	https://www.coursera.org/learn/java-programming-arrays-lists-data
Бази даних в Java	https://www.coursera.org/specializations/java-database-connectivity

11. ТЕХНІЧНІ ТА ІНФОРМАЦІЙНІ РЕСУРСИ

11.1. Технічні ресурси

- аудиторія персональних комп'ютерів класу Celeron, AMD, Pentium, Core 2 Duo, Core i5, i7 (або вище);
- забезпеченість комп'ютерами – не менше 12 шт. на 25 студентів;
- мультимедійне обладнання.

11.2. Програмні ресурси

- системне програмне забезпечення: операційна система Windows (Windows 7, 8 або 10;) або Linux;
- інтегровані системи програмування (IDE) – NetBeans, IntelliJ IDEA, Eclipse, JDeveloper, jCreator або їх аналоги;
- офісні додатки - Microsoft Office Word, Microsoft Office Excel;

10.3 Інформаційні ресурси

- УМК з дисципліни в електронному вигляді, який знаходиться на кафедрі;
- Бібліотека Криворізького національного університету (м. Кривий Ріг, вул. Пушкіна, 37) – Режим доступу: <http://lib.knu.edu.ua/>
- література в електронних ресурсах Інтернет.
- NetBeans IDE. – <http://www.netbeans.org/>
- Eclipse IDE. - <http://www.eclipse.org/>
- IntelliJ IDEA IDE. - <https://www.jetbrains.com/idea/download>
- The Java Tutorials. – <http://docs.oracle.com/javase/tutorial/>

12. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ

12.1 Модуль 1

1. Що таке Java? Історія створення.
2. Особливості мови і платформи Java.
3. Засоби розробки Java -приложений.
4. Інтегровані середовища розробки.
5. Основи об'єктно-орієнтованого програмування.
6. Лексика мови.
7. Типи даних. Вбудовані типи даних.
8. Способи завдання літералів різних типів.
9. Призначення і використання інтерфейсів при програмуванні на мові Java.
10. Абстрактні класи.
11. Статичні члени класу.
12. Псевдовипадкові числа.
13. Оператор привласнення.
14. Порядок дій (пріоритет операторів).
15. Оператори порівняння.
16. Логічні оператори.
17. Вбудований клас String. Строкові операції.
18. Спадкоємство методів в мові Java.

19. Імена. Пакети. Оголошення класів.
20. Перетворення типів.
21. Об'єктна модель в Java
22. Масиви. Алгоритми сортування.
23. Масиви. Багатовимірні масиви.
24. Арифметичні оператори. Оператори інкремента і декремента.
25. Оператори і структура коду.
26. Успадкування полів класів в мові Java.
27. Потоки виконання.
28. Типи-перерахування.
29. Вкладеність класів. Внутрішні локальні класи, анонімні класи.
30. Статичні вкладені і внутрішні класи.
31. Пакет java.awt. Пакет java.lang. Пакет java.util. Пакет java.io.

12.2 Модуль 2

1. Цикл розробки програмних засобів з використанням Java.
2. Мінімальний комплект для розробки програм на Java.
3. Засоби розробки і системні вимоги.
4. Основні технічні можливості NetBeans.
5. Змінні, константи, зона видимості Java.
6. Примітивні типи даних.
7. Посилальні типи даних.
8. Логічні оператори.
9. Арифметичні оператори.
10. Перетворення типів. Оболонкові класи.
11. Клас Math.
12. Складні типи даних. Масиви.
13. Складні типи даних. Рядки.
14. Конструкції, що управляють. Оператор If, Switch.
15. Оператори циклу.
16. Оператори переходу.
17. Основні принципи ООП. Інкапсуляція.
18. Основні принципи ООП. Успадкування.
19. Основні принципи ООП. Поліморфізм.
20. Поняття об'єкту і робота з ним.
21. Типи стосунків між класами. Агрегація, асоціація.
22. Достоїнства і недоліки ООП.
23. Клас. Опис полів класу. New.
24. Опис методів класу. Модифікатори доступу.
25. Конструктори.
26. Посилання This. Перевантаження методів.
27. Спадкоємство. Super. Заміщення методів (override).
28. Динамічне призначення методів.
29. Введення вивід в Java.
30. Елементи графічного інтерфейсу

13. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

13.1. Навчальна та довідкова література

1. Ken Kousen Modern Java Recipes: Simple Solutions to Difficult Problems in Java 8 and 9 1st Edition. Wiley. 2017. 322 p.
2. Christian Ullenboom Java: The Comprehensive Guide to Java Programming for Professionals. Rheinwerk Computing. 2022. 1128 p.
3. Cay S. Horstmann Core Java, Volume I: Fundamentals, 12th Edition. Addison-Wesley Professional. 2021.
4. Herbert Schildt Java: The Complete Reference, Twelfth Edition, 12th Edition. McGraw-Hill. 2021. 1573p.
5. Kathy Sierra, Bert Bates, Trisha Gee Head First Java, 3rd Edition. O'Reilly Media, Inc. 2022.
6. Cay S. Horstmann Core Java 11 Fundamentals, Second Edition. Pearson. 2018.
7. Cay S. Horstmann Core Java for the Impatient 3rd Edition. Addison-Wesley Professional. 2022. 576p.
8. Shildt Herbert. Java Complete guide. Eleventh edition. — McGraw-Hill Education, 2018. — 1248 p.
9. Joshua Bloch Effective Java 3rd Edition. Addison-Wesley Professional; 3rd edition. 2017. 416p.
10. Haris Tsetsekas Object-Oriented Programming Exercises with Java. Kindle Edition. 2023. 154p.
11. Sar Maroof Improve Java Coding: Best Practices for Effective Coding. Kindle Edition. 2023. 3505p.
12. Scott Brandt Java From Zero: Learn Java Programming Fast for Beginners to Professionals: The Complete Guide With Code Examples and Exercises to Become a Professional. Kindle Edition. 2023. 269p.
13. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley The Java® Language Specification. Oracle America, Inc. 2015. 788p.

13.2. Додаткова література

14. Java Platform, Standard Edition & Java Development Kit Version 11 API Specification: <https://docs.oracle.com/en/java/javase/11/docs/api/>
15. Savitch W. Java: an introduction to problem solving & programming. Pearson Prentice Hall, 2005. – 1060p.
16. Алгоритми і структури даних: навчальний посібник / Н. Б. Шаховська; Р.О. Голощук; за заг. ред. Пасічника В.В. - Львів: Магнолія 2006, 2011. - 215 с.
17. Cheng F. Exploring Java 9. Build Modularized Applications in Java. - Berkeley: Apress, 2018. - 174 p.
18. Кадомський К.К., Ніколюк П.К. Java. Теорія і практика: Навч. посіб. - Вінниця: Донну, 2019. 197 с.
19. Васильєв О.М. Програмування мовою Java. — Тернопіль: Богдан, 2019. — 696 с.
20. Simon Kendal. Object oriented programming using Java. Ventus Publishing ApS, 2009. – 209 с.
21. Raoul-gabriel U. Modern Java in Action. Lambdas, streams, functional and reactive programming / Raoul-gabriel Urma, Mario Fusco, Alan Mycroft – Manning Publications Co, 2019. – 592

14 ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

abstract method. A method that has no implementation.

Abstract Window Toolkit (AWT). A collection of graphical user interface (GUI) components that were implemented using native-platform versions of the components. These components provide that subset of functionality which is common to all native platforms. Largely supplanted by the Project Swing component set. See also Swing .

API. Application Programming Interface. The specification of how a programmer writing an application accesses the behavior and state of classes and objects.

applet. A component that typically executes in a Web browser, but can execute in a variety of other applications or devices that support the applet programming model.

Argument. A data item specified in a method call. An argument can be a literal value, a variable, or an expression.

binary operator. An operator that has two arguments.

bitwise operator. An operator that manipulates the bits of one or more of its operands individually and in parallel. Examples include the binary logical operators (&, |, ^), the binary shift operators (<<, >>, >>>) and the unary one's complement operator (~).

Break. A Java keyword used to resume program execution at the statement immediately following the current statement. If followed by a label, the program resumes execution at the labeled statement.

Bytecode. Machine-independent code generated by the Java compiler and executed by the Java interpreter.

Class. In the Java programming language, a type that defines the implementation of a particular kind of object. A class definition defines instance and class variables and methods, as well as specifying the interfaces the class implements and the immediate superclass of the class. If the superclass is not explicitly specified, the superclass will implicitly be Object.

class method. A method that is invoked without reference to a particular object. Class methods affect the class as a whole, not a particular instance of the class. Also called a static method . See also instance method .

compilation unit. The smallest unit of source code that can be compiled. In the current implementation of the Java platform, the compilation unit is a file.

Compiler. A program to translate source code into code to be executed by a computer. The Java compiler translates source code written in the Java programming language into bytecode for the **Java virtual machine** 1 . See also interpreter .

Constructor. A pseudo-method that creates an object. In the Java programming language, constructors are instance methods with the same name as their class. Constructors are invoked using the new keyword.

critical section. A segment of code in which a thread uses resources (such as certain instance variables) that can be used by other threads, but that must not be used by them at the same time.

Default. A Java keyword optionally used after all case conditions in a switch statement. If all case conditions are not matched by the value of the switch variable, the default keyword will be executed.

derived from. Class X is "derived from" class Y if class X extends class Y. See also subclass , superclass.

DOM. Document Object Model. A tree of objects with interfaces for traversing the tree and writing an XML version of it, as defined by the W3C specification.

Else. A Java keyword used to execute a block of statements in the case that the test condition with the if keyword evaluates to false.

Encapsulation. The localization of knowledge within a module. Because objects encapsulate data and implementation, the user of an object can view the object as a black box that provides services. Instance variables and methods can be added, deleted, or changed, but as long as the services provided by the object remain the same, code that uses the object can continue to use it without being rewritten. See also instance variable , instance method.

GUI. Graphical User Interface. Refers to the techniques involved in using graphics, along with a keyboard and a mouse, to provide an easy-to-use interface to some program.

Identifier. The name of an item in a program written in the Java programming language.

Implements. A Java keyword included in the class declaration to specify any interfaces that are implemented by the current class.

Interface. A Java keyword used to define a collection of method definitions and constant values. It can later be implemented by classes that define this interface with the "implements" keyword.

Internet. An enormous network consisting of literally millions of hosts from many organizations and countries around the world. It is physically put together from many smaller networks and data travels by a common set of protocols.

JAR. JAR (Java Archive) is a platform-independent file format that aggregates many files into one. Multiple applets written in the Java programming language, and their requisite components (.class files, images, sounds and other resource files) can be bundled in a JAR file and subsequently downloaded to a browser in a single HTTP transaction. It also supports file compression and digital signatures.

Java. Sun's trademark for a set of technologies for creating and safely running software programs in both stand-alone and networked environments.

Java 2 Platform. The second generation of the Java platform. (The first generation was the JDK.) Also see "Java Platform" and "Java Platform Editions".

Java Database Connectivity (JDBC). An industry standard for database-independent connectivity between the Java platform and a wide range of databases. The JDBC provides a call-level API for SQL-based database access.

Java Development Kit (JDK). A software development environment for writing applets and applications in the Java programming language. Technically, the JDK is the correct name for all versions of the Java platform from 1.0 to 1.1.x.

Java Foundation Classes (JFC). An extension that adds graphical user interface class libraries to the Abstract Windowing Toolkit (AWT).

Object. The principal building blocks of object-oriented programs. Each object is a programming unit consisting of data (instance variables) and functionality (instance methods). See also class.

Package. A group of types . Packages are declared with the package keyword.

Pixel. The picture element on a display area, such as a monitor screen or printed page. Each pixel is individually accessible.

Return. A Java keyword used to finish the execution of a method. It can be followed by a value required by the method definition.

Swing. A collection of graphical user interface (GUI) components that runs uniformly on any native platform which supports the Java virtual machine *. Because they are written entirely in the Java programming language, these components may provide functionality above and beyond that provided by native-platform equivalents. (Contrast with AWT.)

While. A Java keyword used to declare a loop that iterates a block of statements. The loop's exit condition is specified as part of the while statement.

15. ЗМІНИ ТА ДОПОВНЕННЯ ДО РОБОЧОЇ ПРОГРАМИ

[illegible]

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри _____

Схвалено на засіданні кафедри

Протокол № _____ від “_____” _____ 20__ р.

Завідувач кафедри

Схвалено на засіданні кафедри

Протокол № _____ від “ _____ ” _____ 20 ____ р.

Завідувач кафедри

16. РОБОЧИЙ ПЛАН З ДИСЦИПЛІНИ «ПРОГРАМУВАННЯ НА ОСНОВІ JAVA ТЕХНОЛОГІЙ»

Семестр 4

Вид навчальної роботи	Годин у семестрі /кредити	Тиждень																	Вид підсумков ого контролю	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		18
		2		2		2		2		2		2		2		2		2		
Лекційні заняття	18	ПК		ПК		ПК		ПК		КМР1		ПК		ПК		ПК		КМР2		
Лабораторні	18		2		2		2		2		ПК		2		ПК		2		ПК	
Самостійна робота	54	2	3	3	1	3	4	3	3	3	1	3	5	3	3	3	5	3	3	
Всього годин/кредитів	90/3	4	5	5	3	5	6	5	5	5	3	5	7	5	5	5	7	5	5	
																			залік	

Позначки: ПК – поточний контроль, КМР – контрольна модульна робота
Викладач: І.А. Котов