

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
 ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
 ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ  
 Декан факультету комп'ютерних  
 інформаційних технологій  
 Ігор ЯКИМЕНКО  
 "29" 08 2025 р.

ЗАТВЕРДЖУЮ  
 Проректор з науково-педагогічної роботи  
 Віктор ОСТРОВЕРХОВ  
 "29" 08 2025 р.

ЗАТВЕРДЖУЮ  
 Директор навчально наукового  
 інституту новітніх освітніх  
 технологій  
 Святослав ПИТЕЛЬ  
 "29" 08 2025 р.



**РОБОЧА ПРОГРАМА**  
**з дисципліни «Технологія Java»**

Ступінь вищої освіти: бакалавр  
 Галузь знань – 12 «Інформаційні технології»  
 Спеціальність – 126 «Інформаційні системи та технології»  
 Освітньо-професійна програма – «Інформаційні системи та технології»

Кафедра комп'ютерних наук

Форма навчання	Курс	Семестр	Лекції (год.)	Лаб. (год.)	ІРС (год.)	Тренінг (год.)	СРС (год.)	Разом (год.)	Залік (сем.)
денна	3	6	30	14	3	6	97	150	6
заочна	3	6	8	4	-	-	138	150	6

29.08.2025

Тернопіль – ЗУНУ  
 2025

Робоча програма розроблена старшим викладачем кафедри комп'ютерних наук, доктором філософії Василем ТИМЧИШИНОМ.

Робочу програму затверджено на засіданні кафедри комп'ютерних наук, протокол № 1 від 26 серпня 2025 р.

Завідувач кафедри  
д.т.н., професор

  
\_\_\_\_\_ Андрій ПУКАС

Гарант ОП  
д.т.н., професор

  
\_\_\_\_\_ Андрій МЕЛЬНИК

# СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «Технологія Java»

## 1. Опис дисципліни «Технологія Java»

Дисципліна – «Технологія Java»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань - 12 Інформаційні технології	Статус дисципліни: вибіркова Мова навчання: українська
Кількість залікових модулів - 3	Спеціальність :126 Інформаційні системи та технології	Рік підготовки: <i>Денна – 3</i> <i>Заочна – 3</i> Семестр: <i>Денна – 6</i> <i>Заочна – 6</i>
Кількість змістових модулів - 3	Ступінь вищої освіти : бакалавр	Лекції: <i>Денна–30 год.</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна – 14 год.</i> <i>Заочна – 4 год.</i>
Загальна кількість годин - 150		Самостійна робота: <i>Денна – 97 год.</i> тренінг – 6 год. <i>Заочна - 138год.</i> Індивідуальна робота: 3 год.
Тижневих годин – 10 год., з них аудиторних – 3 год.		Вид підсумкового контролю – залік

## 2. Мета й завдання вивчення дисципліни «Технологія Java»

### 2.1. Мета вивчення дисципліни

Мета дисципліни «Технологія Java» полягає у набутті знань та навиків щодо розробки власних front-end та back-end застосунків, що виконуються в середовищі віртуальних машин Java. Курс спрямований на опанування прикладних принципів програмування мовою Java та формування навичок розробки програмних застосунків. У процесі навчання розглядаються ключові поняття синтаксису та структури програм, робота з даними, створення власних програмних структур та використання можливостей мови і бібліотек.

### 2.2. Завдання вивчення дисципліни

У результаті вивчення курсу „Технологія Java” студенти повинні:

- оволодіти синтаксисом, ідіомами та стилем програмування на мові програмування Java;
- розуміти теорію типів, принципи успадкування, контролю доступу та іменування;
- грамотно використовувати параметризовані типи;

- розуміти принципи організації багатониткового (багатопотокового) виконання методів;
- розуміти схеми синхронізації на основіJava-моніторів;
- ефективно користуватися javadoc-коментарями і анотаціями.

### **2.3. Результати навчання**

У результаті вивчення курсу „Технологія Java” студенти повинні вміти:

- використовувати найважливіші поняття, методів і засобів технології Java;
- використовувати основи технології: типи, значення, змінні, перетворення між ними, поняття пакетів;
- користуватись компонентами front-end та back-end розробки;
- ефективно застосовувати засоби вводу-виводу та обробки виключних ситуацій;
- розробляти програми багатозадачної обробки з неатомарними операторами;
- розробляти мережевий back-end на основі серверних компонентів Java.

## **3. Програма навчальної дисципліни «Технологія Java»**

### ***Змістовий модуль 1. Основи мови Java та базові конструкції***

#### **Тема 1. Типи даних. Літерали. Ключові слова. Змінні.**

Типи даних Java. Примітивні типи. Типи-оболонки. Літерали. Ідентифікатори. Ключові слова. Коментарі.Змінні. Оголошення змінних. Ініціалізація змінних. Перетворення та приведення типів. Автоматичне просування типів у виразах. Література: 1–10.

#### **Тема 2. Операції**

Арифметичні операції в Java. Поняття виразів. Пріоритет арифметичних операцій. Порозрядні логічні операції.

Операція присвоювання та комбіновані операції присвоювання. Тернарні операції та їх застосування. Пріоритет операцій у виразах та правила обчислення.

Література: 1–10.

#### **Тема 3. Керуючі оператори**

Умовні оператори. Вкладені конструкції. Оператор вибору switch, кейси, default, break. Приклади застосування циклів для обробки числових послідовностей, масивів, задач перебору.Розширений цикл for (foreach). Використання для обробки масивів, колекцій ArrayList, LinkedList. Оператори break, continue, метки циклів.

Література: 1–10.

#### **Тема 4. Масиви**

Одновимірні масиви. Оголошення, створення, ініціалізація. Доступ до елементів. Властивість `length`. Двовимірні та багатовимірні масиви. Таблична структура даних. Доступ до елементів. Нерегулярні (рвані) масиви. Ініціалізація масивів різної довжини. Приклади використання.

Література: 1–10.

#### **Тема 5. Рядки символів**

Клас `String`. Загальні відомості. Створення рядків. Ініціалізація. Незмінність рядків (`immutable`). Методи класу `String`: довжина, доступ до символів, порівняння, пошук, підрядки, заміна. Масиви рядків. Одновимірні та двовимірні масиви рядків. Ініціалізація та обробка рядкових даних.

Література: 1–10.

### ***Змістовий модуль 2. Об'єктно-орієнтоване програмування в Java***

#### **Тема 6. Класи. Методи в класах**

Поняття класу та об'єкта. Створення класів та об'єктів. Оператор `new`. Посилання на об'єкти. Зберігання даних у пам'яті. Конструктори. Параметризовані та за замовчуванням. Ключове слово `this`. Виклик конструкторів.

Література: 1–10.

#### **Тема 7. Вкладені класи**

Вкладені та внутрішні класи. Статичні вкладені класи. Приклади оголошення та використання. Створення об'єктів вкладених і внутрішніх класів. Конструкції `.this` та `.new`. Використання внутрішніх класів у поєднанні з інтерфейсами. Висхідне та низхідне перетворення. Переваги висхідного перетворення.

Література: 1–10.

#### **Тема 8. Інкапсуляція. Управління доступом. Пакети**

Інкапсуляція. Приховування даних. Модифікатори доступу `private`, `protected`, `public`. Пакети. Структура проекту. Використання `package` та `import`. Файли `.java` та `.class`. Стандартні бібліотеки Java. Управління доступом між класами та пакетами. Пакетний рівень доступу та `public` доступ.

Література: 1–10.

#### **Тема 9. Спадковість у класах. Поліморфізм**

Повторне використання коду. Композиція, спадковість, делегування. `extends`. Суперклас та підклас. Приховування даних у спадкових структурах. Виклик конструктора суперкласу. Посилання на об'єкти підкласу.

Перевизначення та перевантаження методів. Динамічна диспетчеризація. Поліморфізм.

Література: 1–10.

### **Тема 10. Абстрактні класи. Інтерфейси**

Абстрактні класи та методи. `abstract`. Використання на практиці. Інтерфейси. `interface`, `implements`. Особливості застосування. Переваги інтерфейсів.

Література: 1–10.

## *Змістовий модуль 3. Розширені можливості Java та робота з даними*

### **Тема 11. Виключення**

Поняття виключних ситуацій. Обробка виключень `try`, `catch`, `finally`. `throw`, `throws`. Створення та генерація виключень. Класи Java для обробки виключень. `Throwable`, `Exception`. Створення власних виключень. Приклади використання.

Література: 1–10.

### **Тема 12. Зчислення (перерахунки). Автоупакування. Анотації**

Перерахування `enum`. Оголошення та використання. Перерахування як класи: конструктори, поля, методи. Автоупакування та авторозпакування. Використання у виразах, присвоєннях та `switch`.

Література: 1–10.

### **Тема 13. Система вводу/виводу. Робота з файлами**

Потоки вводу/виводу. Байтові та символні потоки. Стандартні потоки. Робота з консоллю: `InputStreamReader`, `PrintStream`. Перенаправлення потоків. Файлова система. Клас `File`. Основні методи.

Література: 1–10.

### **Тема 14. Лямбда-вирази. Функціональні інтерфейси. Посилання на методи**

Основні поняття. Функціональний інтерфейс. Лямбда-вирази для узагальнених функціональних інтерфейсів. Передача лямбда-виразу в метод в якості параметру. Генерування виключень у лямбда-виразах. Доступ до елементів класу з лямбда-виразу. Захоплення змінних у лямбда-виразах.

Література: 1–10.

#### 4. Структура залікового кредиту дисципліни «Технологія Java»

денна форма навчання	Кількість годин					
	Лекції	Лабораторна робота	СРС	ІРС	Тренінг	Контрольні заходи
Змістовий модуль 1. Основи мови Java та базові конструкції						
Тема 1. Типи даних. Літерали. Ключові слова. Змінні	2	1	6	1	2	Усне опитування/тестування
Тема 2. Операції	2	1	7			Усне опитування/тестування
Тема 3. Керуючі оператори	2	1	7			Усне опитування/тестування
Тема 4. Масиви	2	1	7			Усне опитування/тестування
Тема 5. Рядки символів	2	1	7			Усне опитування/тестування
Змістовий модуль 2. Об'єктно-орієнтоване програмування в Java						
Тема 6. Класи. Методи в класах	2	1	7	1	2	Усне опитування/тестування
Тема 7. Вкладені класи	2	1	7			Усне опитування/тестування
Тема 8. Інкапсуляція. Управління доступом. Пакети	2	1	7			Усне опитування/тестування
Тема 9. Спадковість у класах. Поліморфізм	2	1	7			Усне опитування/тестування
Тема 10. Абстрактні класи. Інтерфейси	2	1	7			Усне опитування/тестування
Змістовий модуль 3. Розширені можливості Java та робота з даними						
Тема 11. Виключення	2	1	7	1	2	Усне опитування/тестування
Тема 12. Зчислення (перерахунки). Автоупакування. Анотації	2	1	7			Усне опитування/тестування
Тема 13. Система вводу/виводу. Робота з файлами	3	1	7			Усне опитування/тестування
Тема 14. Лямбда-вирази. Функціональні інтерфейси. Посилання на методи	3	1	7			Усне опитування/тестування
Разом	30	14	97	3	6	

заочна форма навчання	Кількість годин		
	Лекції	Лабораторна робота	СРС
Тема 1. Типи даних. Літерали. Ключові слова. Змінні			8
Тема 2. Операції		1	10
Тема 3. Керуючі оператори			10
Тема 4. Масиви			10
Тема 5. Рядки символів	2	1	10
Тема 6. Класи. Методи в класах			10
Тема 7. Вкладені класи		1	10
Тема 8. Інкапсуляція. Управління доступом. Пакети	2		10
Тема 9. Спадковість у класах. Поліморфізм	2		10
Тема 10. Абстрактні класи. Інтерфейси			10
Тема 11. Виключення		1	10
Тема 12. Зчислення (перерахунки). Автоупакування. Анотації			10
Тема 13. Система вводу/виводу. Робота з файлами			10
Тема 14. Лямбда-вирази. Функціональні інтерфейси. Посилання на методи	2		10
Разом	8	4	138

## 5. Тематика лабораторних робіт

### Лабораторна робота № 1

**Тема:** Програмування лінійних алгоритмів. Стандартні класи і їхні методи у мові Java.

**Мета:** Отримати практичні навички створення програм з лінійною структурою, де команди виконуються послідовно одна за одною. Навчитися застосовувати методи стандартних класів для реалізації математичних обчислень та формул.

**Питання для обговорення:**

1. Яка структура найпростішої Java-програми та яку роль відіграє метод main?
2. Які стандартні класи Java використовуються для виконання математичних операцій та роботи з рядками?
3. Як організувати введення даних з клавіатури та форматований вивід у консоль?
4. Які примітивні типи даних існують у Java та як відбувається перетворення типів у арифметичних виразах?
5. Який порядок виконання операцій (пріоритет) діє при обчисленні складних лінійних виразів?

## Лабораторна робота №2

**Тема:** Основні алгоритмічні структури мови Java.

**Мета:** Навчитися використовувати умовні оператори та цикли для керування ходом виконання програми.

**Питання для обговорення:**

1. Як працюють умовні конструкції if-else та вкладені умови?
2. У яких випадках доцільно використовувати оператор вибору switch?
3. Чим відрізняється цикл з передумовою (while) від циклу з післяумовою (do-while)?
4. Як організувати цикл з лічильником за допомогою оператора for?
5. Для чого використовуються оператори переходу break та continue?

## Лабораторна робота №3

**Тема:** Посилальні типи даних мови Java.

**Мета:** Отримати практичні навички роботи з масивами та рядками.

**Питання для обговорення:**

1. Чим принципово відрізняються примітивні та посилальні типи даних?
2. Як оголошувати, ініціалізувати та перебирати елементи масивів?
3. Що таке багатовимірні масиви та як вони реалізовані в Java?
4. Які особливості класу String та що означає незмінність (immutability) рядків?
5. У чому різниця між порівнянням через == та використанням методу equals() для об'єктів?

## Лабораторна робота №4

**Тема:** Застосування стандартної бібліотеки мови Java.

**Мета:** Опанувати роботу з класами пакетів java.lang та java.util. Навчитися працювати з класами-обгортками, сучасним API для дати та часу, а також форматуванням даних.

**Питання для обговорення:**

1. Що таке класи-обгортки і як працює автоупаковка?
2. Які основні класи використовуються для роботи з датою та часом?
3. Як працює клас Math та його статичні методи?
4. Що таке регулярні вирази та класи Pattern і Matcher?
5. Як здійснюється форматування рядків та чисел у Java?

## Лабораторна робота №5

**Тема:** Робота з файлами. Застосування колекцій.

**Мета:** Навчитися використовувати потоки введення-виведення для читання та запису файлів

**Питання для обговорення:**

1. Чим відрізняються байтові та символні потоки введення-виведення?
2. Як працювати з файловою системою за допомогою класів File та Path?
3. Яка структура Java Collections Framework та основні відмінності між List, Set та Map?
4. У чому різниця між ArrayList та LinkedList?

5. Як працює ітератор і для чого він потрібен при обході колекцій?

### **Лабораторна робота №6**

**Тема:** Потоки виконання (threads) і синхронізація.

**Мета:** Отримати навички створення багатопотокових програм. Навчитися використовувати принципи паралельного виконання коду та механізми синхронізації доступу до спільних ресурсів.

**Питання для обговорення:**

1. Які існують способи створення та запуску нового потоку (Thread vs Runnable)?
2. Які стани має життєвий цикл потоку в Java?
3. Що таке стан гонитви (race condition) і як його уникнути?
4. Як працює ключове слово synchronized та монітори об'єктів?
5. Для чого використовуються методи wait, notify та notifyAll?

### **Лабораторна робота №7**

**Тема:** Обробка виключних ситуацій у Java. Власні виключення.

**Мета:** Опанувати механізми обробки виключень у Java. Навчитися забезпечувати надійну обробку помилок у програмах та запобігати їх некоректному завершенню.

**Питання для обговорення:**

1. Що таке виключна ситуація (exception) та як вона впливає на виконання програми?
2. Як працюють конструкції try, catch та finally?
3. У чому різниця між throw та throws?
4. Як створити власний клас виключення та у яких випадках це доцільно?
5. Що таке ієрархія виключень у Java та чим відрізняються checked і unchecked exceptions?

## **6. Тренінг з дисципліни**

Тренінг (training) – це запланований процес модифікації (зміни) відношення, знання чи поведінкових навичок того, хто навчається, через набуття навчального досвіду з тим, щоб досягти ефективного виконання поставлених завдань в одному із видів діяльності.

Порядок проведення тренінгу:

1. Вступна частина проводиться з метою ознайомлення студентів з темою тренінгового заняття.

2. Організаційна частина полягає у створенні робочого настрою у колективі студентів, визначенні правил.

3. Практична частина реалізується шляхом виконання завдань у групах студентів у кількості 3–5 осіб з певних проблемних питань теми тренінгового заняття (розробити стратегію оплати праці персоналу в ІТ-компанії). Практична частина, зокрема, включає: - ознайомитись з існуючими підходами до організації менеджменту людських ресурсів в ІТ; - розробити схему матеріальної мотивації для працівників, включаючи фіксовану частину і можливі премії (відсотки прибутку, бонуси за певні завдання і т.п.); - розробити схему нематеріальної мотивації для працівників (мінімум 2 критерії та

винагороди); - розробити критерії визначення кращих працівників компанії та схему мотивації за 1-3 місяці.

4. Підведення підсумків – обговорення результатів виконаних завдань у групах (представити результати у вигляді презентації); обмін думками з питань, які виносили на тренінг.

## **7. Самостійна робота**

Самостійна робота є основним засобом засвоєння студентами навчального матеріалу в позааудиторний час без участі викладача. Обсяг і зміст самостійної роботи визначається робочою програмою та робочим планом в межах встановленого обсягу годин із навчальної дисципліни, методичними вказівками викладача.

Самостійна робота студента забезпечується системою навчально-методичних засобів: конспектами лекцій викладача, підручниками, навчальними та методичними посібниками, монографічною літературою і періодикою, а також засобами самоконтролю (тестами, типовим пакетом контрольних завдань).

Навчальний матеріал, передбачений навчальним планом для засвоєння студентом в процесі самостійної роботи, виноситься на підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався при проведенні навчальних занять.

Викладач проводить діагностику якості самостійної роботи студента на індивідуальних заняттях. Вони проводяться в позааудиторний час за окремим графіком, складеним кафедрою.

Самостійна робота студента проводиться за такими напрямками:

1. Підготовка до практичних занять:

- вивчення лекційного матеріалу;
- вивчення теми за підручниками і посібниками;
- опрацювання завдань до практичних занять.

2. Опрацювання програмних питань, що не розглядаються на навчальних заняттях і виносяться на самостійне опрацювання.

3. Ведення конспекту.

4. Підготовка до написання модульних контрольних робіт.

5. Підготовка до екзамену.

### **Перелік тем для самостійної роботи**

1 Обробка параметрів командного рядка засобами мови програмування Java.

2 Зчитування і запис даних у файли формату CSV на мові програмування Java.

3 Організація обміну даними з REST сервісом на мові програмування Java.

4 Класи та об'єкти

5. Застосування класів у програмах на Java. Визначення класу та об'єкту класу. Приклади

6. Об'єкти. Створення та збереження об'єктів класів. Оператор new. Області зберігання даних в пам'яті. Використання масивів посилань на об'єкти

7. Область дії об'єктів (scope). Видалення об'єктів. Переваги використання підходу “збору сміття” (garbagecollection) для об'єктів
8. Конструктори
9. Конструктори. Параметризовані конструктори. Ключове слово this. Збір “сміття”. Метод finalize(). Приклади
10. Конструктори за замовчуванням (defaultconstructors). Виклик конструкторів класу з інших конструкторів
11. Ініціалізація в класах
12. Ініціалізація змінних у методах класу. Ініціалізація членів даних класу. Способи ініціалізації членів даних класу
13. Ініціалізація статичних членів даних. Статичний блок. Ініціалізація одновимірних та багатовимірних масивів
14. Методи у класах
15. Поняття методу. Загальна форма. Параметри методу. Повернення з методу. Оператор return. Повернення об'єкту з методу. Приклади методів
16. Методи. Передача параметрів у методах класу. Передача змінних простих типів та об'єктів у метод в якості параметру
17. Перевантаження методів у класах. Перевантаження конструкторів
18. Аргументи змінної довжини у методах. Перевантаження методів з аргументами змінної довжини. Властивість length
19. Рекурсія
20. Рекурсія. Приклади розв'язку задач. Переваги та недоліки рекурсії
21. Приклади розв'язку задач на рекурсію в Java
22. Статичні члени даних класу. Статичні методи. Ключове слово static
23. Специфікатор final. Незмінні дані, методи, класи. Пусті константи. Статичні незмінні дані
24. Вкладені класи
25. Вкладені та внутрішні класи. Статичні вкладені класи. Приклади
26. Створення об'єктів вкладених статичних та нестатичних класів. Конструкції .this та .new
27. Приклади застосування внутрішніх класів в поєднанні з інтерфейсами. Низхідне та висхідне перетворення. Переваги застосування висхідного перетворення
28. Інкапсуляція. Управління доступом. Пакети
29. Інкапсуляція. Управління доступом в класі. Модифікатори private, protected, public
30. Пакети. Використання пакетів у Java. Директиви import та package. Компільовані модулі (.java). Проміжні .class файли. Використання стандартних бібліотек Java
31. Управління доступом до класів у пакетах. Пакетний рівень доступу до класу. Загальнодоступний(public) рівень доступу до класу
32. Спадковість у класах. Поліморфізм
33. Повторне використання коду в класах. Поняття композиції, спадковості, делегування. Ключове слово extends. Приклади
34. Спадковість. Основні поняття. Суперклас та підклас. Ключове слово extends. Приховування даних в успадкованих класах. Модифікатори доступу private, protected, public

35. Спадковість. Посилання на об'єкт підкласу. Виклик конструктора суперкласу. Ключове слово `super`
36. Перевизначення та перевантаження успадкованих методів. Приклади. Динамічна диспетчеризація методів. Поліморфізм
37. Абстрактні класи. Інтерфейси
38. Абстрактні класи. Абстрактні методи. Приклади. Ключове слово `abstract`
39. Інтерфейси. Особливості застосування в поєднанні з класами. Переваги застосування інтерфейсів. Ключові слова `interface`, `implements`
40. Виключення
41. Виключення (exceptions). Виключна ситуація. Ключові слова `try`, `catch`, `finally`
42. Оператори `throw`, `throws`. Приклади
43. Класи Java для обробки виключних ситуацій з пакету `java.lang`. Методи класу `Throwable`. Приклади
44. Клас `Exception`. Створення власних класів виключень. Приклади
45. Зчислення (перерахунки). Автоупакування. Анотації
46. Зчислення (enumeration). Загальні поняття. Ключове слово `enum`.  
Приклади
47. Використання зчислень як класів. Приклади оголошення зчислень, що містять конструктори, внутрішні поля, методи
48. Автоупакування та авторозпакування. Загальні поняття. Використання при присвоєнні (`=`) та в методах. Приклади
49. Автоупакування та авторозпакування у виразах та операторі `switch`. Використання автоупакування та авторозпакування значень для типів `boolean` та `char`
50. Система вводу/виводу. Робота з файлами
51. Система вводу/виводу Java. Потік. Байтові потоки. Символьні потоки. Стандартні потоки
52. Робота з консоллю в Java. Класи `InputStreamReader`, `PrintStream`. Створення потоку вводу/виводу зв'язаного з консоллю. Перенаправлення потоків вводу виводу
53. Робота з файлами в Java. Клас `File`. Основні методи роботи
54. Байтові потоки. Класи `DataInputStream`, `DataOutputStream`, `FileInputStream`, `FileOutputStream`. Приклади використання
55. Приклади реалізації операцій, які модифікують текстові файли. Класи `FileReader`, `FileOutputStream`, `PrintStream`
56. Автоматичне закриття файлу. Оператор `try` з ресурсами. Приклади
57. Серіалізація об'єктів. Ключове слово `transient`. Приклади
58. Узагальнення
59. Узагальнення. Параметризовані типи. Узагальнені класи, інтерфейси, методи
60. Забезпечення типової безпеки з допомогою узагальнень. Узагальнені інтерфейси. Приклади

## 8. Методи навчання

У навчальному процесі застосовуються: лекції, в тому числі з використання мультимедійного проектора та інших ТЗН; лабораторні роботи, індивідуальні заняття; робота в Інтернет.

## 9. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Технологія Java» використовуються наступні методи оцінювання навчальної роботи студентів:

- поточне тестування та опитування;
- презентації результатів виконаних завдань і досліджень;
- оцінювання результатів модульних контрольних робіт;
- інші види індивідуальних і групових завдань;
- підсумковий залік.

## 10. Політика оцінювання

– *Політика щодо дедлайнів і перескладання.* Для виконання усіх видів завдань здобувачами освіти і проведення контрольних заходів встановлюються конкретні терміни. Перескладання модулів проводиться в установленому порядку.

– *Політика щодо академічної доброчесності.* Списування під час проведення контрольних заходів заборонені. Під час контрольного заходу учасник може користуватися лише дозволеними допоміжними матеріалами або засобами, йому забороняється в будь-якій формі обмінюватися інформацією з іншими учасниками, використовувати, розповсюджувати, збирати варіанти контрольних завдань.

– *Політика щодо відвідування.* За об'єктивних причин (наприклад, карантин, воєнний стан, хвороба, закордонне стажування) навчання може відбуватись в дистанційній формі за погодженням із керівником курсу з дозволу дирекції факультету.

## 11. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Технологія Java» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2	Модуль 3
40 %	40 %	5 %	15%
Поточне оцінювання	Модульний контроль	Тренінги	Самостійна робота
Оцінка за поточне оцінювання визначається, як середнє арифметичне з оцінок, отриманих під час занять	Середнє арифметичне із оцінок, отриманих за виконання лабораторних робіт 1-4	Визначається як єдина оцінка, отримана під час презентації проекту, розробленого під час тренінгу	Визначається як єдина оцінка по 100-бальній шкалі за результат самостійної роботи за обраною тематикою

### **Виконання лабораторних робіт:**

90-100 балів (Відмінно) - здобувач освіти самостійно, без помилок, виконав усі кроки в рамках лабораторної роботи, правильно задокументував етапи та вільно оперує поняттями та принципами, що відносяться до теми дисципліни.

75-89 балів (Добре) - здобувач освіти виконав завдання лабораторної роботи, проте в процесі виконання допустив кілька дрібних помилок, які не вплинули на кінцевий результат (наприклад, неточна послідовність дій), в процесі роботи виникали додаткові запитання.

60-74 балів (Задовільно) - здобувач освіти виконав завдання лабораторної роботи, але з суттєвими помилками, наприклад, не з першого разу чи не до кінця. Розуміння поставлених у лабораторній роботі завдань є поверхневим та неповним.

1-59 балів (Незадовільно) - здобувач освіти не зміг виконати завдання або результати були повністю невірними. Не продемонстрував базових навичок роботи з програмним забезпеченням.

### **Тренінг:**

90-100 балів (Відмінно) - здобувач освіти самостійно, без помилок, виконав усі етапи завдання, правильно задокументував усі етапи роботи, та вільно оперує поняттями та принципами дисципліни.

75-89 балів (Добре) - здобувач освіти виконав завдання, але з кількома дрібними помилками, які не вплинули на кінцевий результат, в процесі роботи виникали додаткові запитання.

60-74 балів (Задовільно) - здобувач освіти виконав завдання, але з суттєвими помилками, наприклад, не з першого разу. Розуміння поставлених у тренінгу завдань є поверхневим.

1-59 балів (Незадовільно) - здобувач освіти не зміг виконати завдання або результати були повністю невірними. Не продемонстрував достатній рівень навичок роботи з апаратним та програмним забезпеченням.

**Самостійна робота:** оцінюється у вигляді єдиної підсумкової оцінки за 100-бальною шкалою на основі результатів самостійної роботи студента з обраної теми.

### **Шкала оцінювання:**

За шкалою ЗУНУ	За національною шкалою	За шкалою ECTS
90–100	відмінно	A (відмінно)
85–89	добре	B (дуже добре)
75-84		C (добре)
65-74		D (задовільно)
60-64	задовільно	E (достатньо)
35-59		FX (незадовільно з можливістю повторного складання)
1-34	незадовільно	F (незадовільно з обов'язковим повторним курсом)

## **12. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна**

№	Найменування	Номер теми
1.	Мультимедійний проектор	1-14

2.	Проекційний екран	1-14
3.	Комунікаційне програмне забезпечення (Internet Explorer, GoogleChrome, Firefox)	1-14
4.	Операційна система Windows, наявність доступу до мережі Internet	1-14
5.	Персональні комп'ютери	1-14
6.	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-14
7.	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-14
8.	Базове програмне забезпечення Microsoft Office	1-14
9	Спеціалізоване програмне забезпечення IntelliJ IDEA Ultimate, Java.	1-14

### **РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ**

1. Schildt H. Java: The Complete Reference, 12th Edition / Herbert Schildt. – New York: McGraw-Hill Education, 2022. – 1248 p.
2. Walls C. Spring in Action, 6th Edition / Craig Walls. – Shelter Island, NY: Manning Publications Co., 2022. – 824 p.
3. Ullenboom C. Spring Boot 3 and Spring Framework 6: Build Scalable Java Applications / Christian Ullenboom. – Berkeley, CA: Apress Media LLC, 2023. – 712 p.
4. Pivotal Team. Java Persistence with Spring Data and Hibernate / Pivotal Software Inc. – Shelter Island, NY: Manning Publications Co., 2023. – 640 p.
5. Logan A. Java for Beginners 2022: The Best Guide to Start Coding in Java Immediately / Alex Logan. – Independently Published, 2022. – 295 p.
6. Bloch J. Effective Java / Joshua Bloch. – Addison-Wesley Professional, 2018. – 416 p.
7. Schildt H.; Coward D. Java: The Complete Reference / Herbert Schildt, Danny Coward. – McGraw-Hill Education, 2024. – 1408 p.
8. Sierra K.; Bates B.; Gee T. Head First Java / Kathy Sierra, Bert Bates, Trisha Gee. – O'Reilly Media, 2022. – 754 p.
9. Horstmann C. S. Core Java, Volume I: Fundamentals / Cay S. Horstmann. – Oracle Press, 2024. – 848 p.
10. Eckel B. Thinking in Java / Bruce Eckel. – Prentice Hall, 2022. – 1152 p.