

0. 목차

1. Java 언어 개요 및 개발 환경

- Java란 무엇인가? (역사, 특징, 플랫폼 독립성)
- JDK, JRE, JVM 구조
- Java 버전별 주요 변화 (8, 11, 17, 21 등)
- 개발 도구 설치 (Eclipse, IntelliJ, VSCode)
- 첫 번째 프로그램 (`Hello, world`)

2. 기초 문법

- 식별자와 키워드
- 데이터 타입 (기본형과 참조형)
- 변수 선언 및 초기화
- 연산자 (산술, 비교, 논리, 비트, instanceof 등)
- 형변환 (암시적, 명시적)
- 문자열 처리 (`String`, `StringBuilder`, `StringBuffer`)
- 배열 (1차원, 다차원)

3. 제어문

- 조건문 (`if`, `if-else`, `switch`)
- 반복문 (`for`, `while`, `do-while`)
- `break`, `continue`, `label` 사용법

4. 메서드와 스택 프레임

- 메서드 정의 및 호출
- 매개변수 전달 (값 전달, 참조 전달)
- 반환값 처리
- 메서드 오버로딩

5. 객체지향 프로그래밍 (OOP)

- 클래스와 객체
- 생성자와 초기화 블록
- 접근제한자 (`public`, `private`, `protected`, default)
- `this` 키워드
- 객체 간 협력 및 관계 (has-a, is-a)
- 캡슐화, 추상화, 상속, 다형성

6. 상속과 다형성

- `extends` 키워드
- 메서드 오버라이딩
- 타입 캐스팅 (업캐스팅, 다운캐스팅)
- `instanceof` 연산자
- `final` 키워드

7. 추상 클래스와 인터페이스

- 추상 클래스 (`abstract`)
- 인터페이스의 정의와 구현 (`implements`)
- 다중 구현 및 다형성
- Java 8 이후 인터페이스의 `default`, `static` 메서드

8. 예외 처리

- 예외 계층 구조 (`Exception`, `Error`, `RuntimeException`)
- `try-catch-finally` 구조
- `throw`, `throws` 키워드
- 사용자 정의 예외

9. Java의 주요 API

- `java.lang`, `java.util`, `java.io` 등
- 날짜/시간 API (`java.time`)
- 난수 생성 (`Random`, `SecureRandom`)
- 래퍼 클래스 (`Integer`, `Double` 등)

10. 컬렉션 프레임워크

- List: `ArrayList`, `LinkedList`
- Set: `HashSet`, `TreeSet`, `LinkedHashSet`
- Map: `HashMap`, `TreeMap`, `LinkedHashMap`
- Queue: `PriorityQueue`, `Deque`
- 반복자 (`Iterator`, `ListIterator`)
- 컬렉션 정렬 (`Comparable`, `Comparator`)

11. 제네릭(Generics)

- 제네릭 클래스, 메서드 정의
- 와일드카드 (`<?>`, `<? extends T>`, `<? super T>`)

- 제네릭 제한 (`T extends`, `T super`)
- 타입 소거(Type Erasure)

12. 내부 클래스

- 인스턴스 내부 클래스
- 정적 내부 클래스 (`static class`)
- 지역 내부 클래스
- 익명 클래스 (Anonymous class)
- 람다식과 함수형 인터페이스

13. 열거형 (Enum)

- enum 정의 및 사용
- 메서드, 생성자 포함 enum
- `values()`, `ordinal()`, `valueOf()`

14. 자바 입출력 (I/O)

- `InputStream`, `OutputStream`
- `Reader`, `Writer`
- `BufferedReader`, `BufferedWriter`
- 파일 읽기/쓰기 (`File`, `FileReader`, `Files`)
- 직렬화 (`Serializable`, `transient`)

15. 멀티스레딩과 동기화

- Thread 생성 방법 (`extends Thread`, `implements Runnable`)
- Thread 상태와 생명주기
- `synchronized` 키워드
- `wait()`, `notify()`, `notifyAll()`
- `Thread.sleep()`, `yield()`, `join()`
- `ExecutorService` 와 스레드풀

16. 람다와 함수형 프로그래밍

- 람다식 기본 문법
- 함수형 인터페이스 (`@FunctionalInterface`)
- 주요 함수형 인터페이스 (`Function`, `Consumer`, `Supplier`, `Predicate`)
- 메서드 참조 (`::`)
- Stream API와 함께 사용

17. Stream API

- `Stream` 생성 및 가공
- 중간 연산: `filter`, `map`, `distinct`, `sorted`
- 최종 연산: `forEach`, `collect`, `reduce`, `count`
- 병렬 스트림 (`parallelStream()`)

18. 애노테이션 (Annotation)

- 기본 애노테이션 (`@Override`, `@Deprecated`, `@SuppressWarnings`)
- 사용자 정의 애노테이션
- 리플렉션을 통한 애노테이션 처리

19. 리플렉션 (Reflection)

- 클래스 정보 탐색 (`Class`, `Field`, `Method`, `Constructor`)
- 객체 생성 및 메서드 호출
- `private` 멤버 접근

20. 모듈 시스템 (Java 9+)

- `module-info.java`
- `exports`, `requires` 키워드
- 모듈 간 의존성

21. JDBC (Java Database Connectivity)

- 드라이버 로딩과 연결
- SQL 실행 (`Statement`, `PreparedStatement`)
- 트랜잭션 처리
- `ResultSet` 처리
- DB 커넥션 풀 (HikariCP 등)

22. 네트워크 프로그래밍

- 소켓 통신 (`Socket`, `ServerSocket`)
- UDP 통신 (`DatagramSocket`, `DatagramPacket`)
- HTTP 클라이언트 (`URLConnection`, `HttpClient`)

23. GUI 프로그래밍

- AWT와 Swing 개요
- 주요 컴포넌트 (`JFrame`, `JButton`, `TextField`)
- 이벤트 처리 모델

- 레이아웃 관리자

24. JavaFX (현대적 GUI 프레임워크)

- Stage, Scene, Node 개념
- FXML 기반 설계
- 이벤트 처리, CSS 적용

25. 빌드 및 배포

- `javac`, `java`, `jar` 명령어
- Maven / Gradle 빌드 도구
- JAR 생성 및 실행
- 모듈 기반 배포 (JLink, JPackage)

26. 테스트 및 디버깅

- 단위 테스트 (`JUnit5`, `Mockito`)
- 애플리케이션 로깅 (`Log4j`, `SLF4J`, `java.util.logging`)
- 디버깅 기법 (`breakpoint`, `watch`, `stack trace`)

27. 고급 주제

- 클래스 로더 구조
- Garbage Collection 알고리즘
- 메모리 관리 구조 (Heap, Stack, Metaspace)
- Unsafe API
- JVM 튜닝 및 프로파일링 (`jvisualvm`, `jconsole`, `Flight Recorder`)

28. 실무 프레임워크 연동

- Spring Core, Spring Boot 기초
- Hibernate / JPA 연동
- RESTful API 구현
- Jackson을 이용한 JSON 처리
- Java + Docker 배포