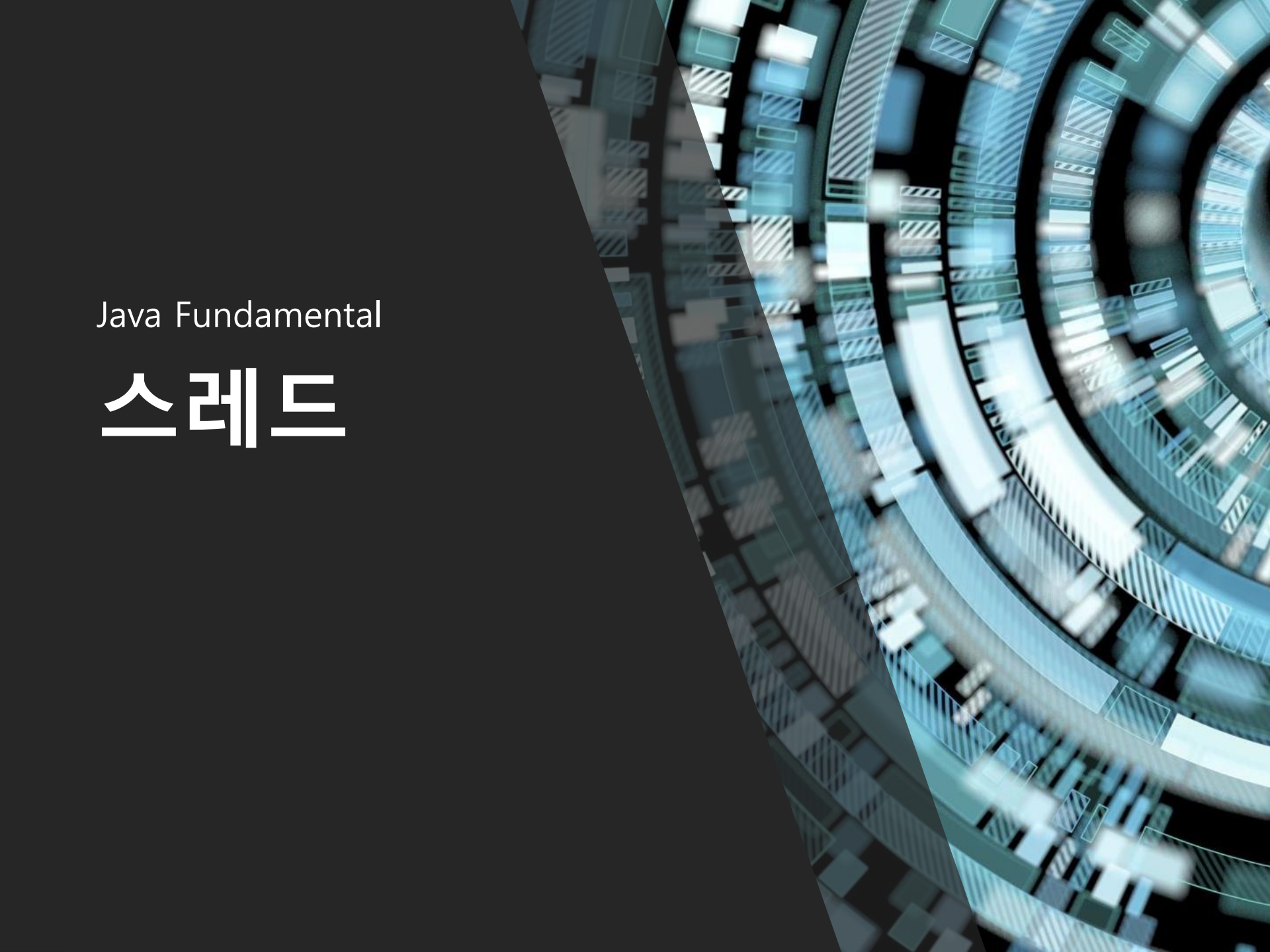
Java Fundamental

스레드



스레드

- **프로세스 내**에서 **실행되는 흐름**의 단위
- 프로세스의 **데이터를 공유**
- 둘 이상의 스레드를 동시에 진행하는 경우를 **멀티스레드**
- 멀티스레드를 통해 **동시에 여러 작업을 처리**할 수 있음
- 처리 성능 향상, 효율적인 데이터 공유 가능
- 복잡한 구현, 동기화 문제 발생

스레드의 생성

- **Thread 클래스를 상속**한 후 인스턴스 생성
- **Runnable 인터페이스를 구현**한 후 스레드 생성자에 전달
- **run()** 메서드에 **동작시킬 코드**를 **오버라이드**

요소	설명
Thread()	새 스레드 객체를 할당
Thread(Runnable target)	새 스레드 객체를 할당

스레드의 실행과 종료

- **start()** 메서드로 스레드를 실행
- 여러 개의 스레드가 실행 중이면 각각 **빠르게 번갈아가며 동작**
- run() 메서드의 **코드를 모두 수행한 후 종료**

요소	설명
static int activeCount()	활성화된 스레드의 추산치를 반환
static Thread currentThread()	현재 실행 중인 스레드의 참조를 반환
void start()	스레드를 실행 *JVM에 의해 run() 메서드 실행

스레드 스케줄링

- **어떤 스레드의 동작을 먼저 수행할 것인지** 정하는 것
- 스레드 별로 **우선순위**를 판별하여 결정

요소	설명
static int MAX_PRIORITY	최대 우선순위(10)
static int MIN_PRIORITY	최소 우선순위(1)
static int NORM_PRIORITY	기본 우선순위(5)
int getPriority()	이 스레드의 우선순위를 반환
void setPriority(int newPriority)	이 스레드의 우선순위를 변경

스레드의 중단

- **sleep()** 메서드를 통해 작동 중인 스레드를 **일시 중단** 가능
- **인터럽트**를 발생시켜 **완전 중단** 가능

요소	설명
static void sleep(long millis)	현재 실행 중인 스레드를 명시된 숫자의 밀리 초만큼 일시 중단
void interrupt()	해당 스레드에 인터럽트를 발생시킴
static boolean interrupted()	현재 스레드에 인터럽트가 발생되었는지 검사
boolean isInterrupted()	해당 스레드에 인터럽트가 발생되었는지 검사

스레드 간섭

- **다수의 스레드**가 **공유된 데이터**에 접근할 때 발생하는 문제
- **메서드** 또는 **블록**에 **synchronized** 키워드를 통해 동기화

```
public synchronized void hello()
{
    // 동기화할 내용
}
```

```
synchronized (공유할 객체)
{
    // 동기화할 내용
}
```

스레드의 대기와 알림

- **wait()** 메서드로 다른 스레드의 처리 결과를 **기다림**
- **notify()** 메서드로 다른 스레드에게 처리 완료를 **알림**
- **공유 데이터의 동기화** 필요

요소	설명
void wait()	다른 스레드의 알림을 기다림
void wait(long timeout)	다른 스레드의 알림을 timeout 밀리초까지 기다림
void notify()	대기 중인 단일 스레드를 깨움
void notifyAll()	대기 중인 모든 스레드를 깨움