

# Chapter 16 Multithreading

## 16.1 Multitasking

멀티 태스킹이란?

- Multi-tasking : 여러 개의 작업을 동시에 실행하는 기법
- 단일 코어 CPU에서도 멀티태스킹이 가능하다. 운영체계가 CPU의 시간을 쪼개서 작업들에 기산을 할당하기 때문에 작업들이 동시에 수행되는 것처럼 보인다.
- Multi-threading : 하나의 애플리케이션 안에서 여러 가지 작업(Thread : 실행 흐름(실))을 동시에 하는 기법.

프로세스와 스레드

- 컴퓨터의 기본적인 실행 단위 2가지 : process, thread

Process	Thread
자신만의 데이터를 갖는다.	스레드들은 모두 동일한 데이터를 공유한다.

Multi-threading을 사용하는 이유

- 빠른 실행
- 최근의 CPU는 속도가 매우 빠르며 여러 개의 코어가 포함되어 있기 때문에 하나의 스레드로는 모든 코어를 이용할 수 없다.
- 멀티스레딩을 사용하면 여러 코어를 최대한 활용할 수 있다.
- 마우스와 키보드에서 입력을 받고 네트워크에 무언가 업로드하는 동시에 화면도 그려야 하는 게임에서도 사용된다.

멀티스레딩의 문제점

- 동기화 문제 : 여러 스레드들이 같은 데이터를 공유하게 되면서 발생한다.

## 16.2 Threading 생성과 실행

- 자바에서 스레드를 생성하여 작업을 실행하는 2가지 방법
- Thread Class 상속
- Runnable Interface 구현

Thread Class

- Thread Class : Thread를 나타내는 클래스

Method of Thread Class	Description
Thread()	매개 변수가 없는 기본 생성자
Thread(String name)	이름이 name인 Thread 객체를 생성한다.
static int activeCount()	현재 활동 중인 스레드의 개수를 반환한다.
String getName()	스레드의 이름을 반환한다.

Method of Thread Class	Description
int getPriority()	스레드의 우선순위를 반환한다.
void	interrupt()
boolean isInterrupted()	현재의 스레드가 중단될 수 있는지를 검사한다.
void setPriority(int priority)	스레드의 우선순위를 지정한다.
void setName(String name)	스레드의 이름을 지정한다.
static void sleep(int milliseconds)	현재의 스레드를 지정된 시간만큼 재운다.
void run()	스레드가 해야 하는 작업을 이 메소드 안에 위치시킨다. 스레드가 시작될 때 호출된다.
void start()	스레드를 시작한다.
static void yield()	현재 스레드를 다른 스레드에 양보하게 만든다.

### Thread Class를 상속하여 스레드 생성하기

- 첫 번째 방법은 Thread Class를 상속받아 자식 클래스를 만들고 run() 메소드를 재정의하는 방법이다.
- run() 메소드에는 스레드가 수행하여야 할 작업 내용이 들어간다.
- 자식 클래스의 인스턴스를 생성하고 start() 메소드를 호출하면 스레드가 실행된다.

```
class MyThread extends Thread {
    public void run() {
        for (int i = 0; i <= 10; i++) {
            System.out.print(i + " ");
        }
    }
}

public class MyThreadTest {
    public static void main(String [] args) {
        Thread t = new MyThread();
        t.start();
    }
}

// 0 1 2 3 4 5 6 7 8 9 10
```

## 16.3 Thread Scheduling

## 16.4 동기화

## 16.5 Thread 간의 조정