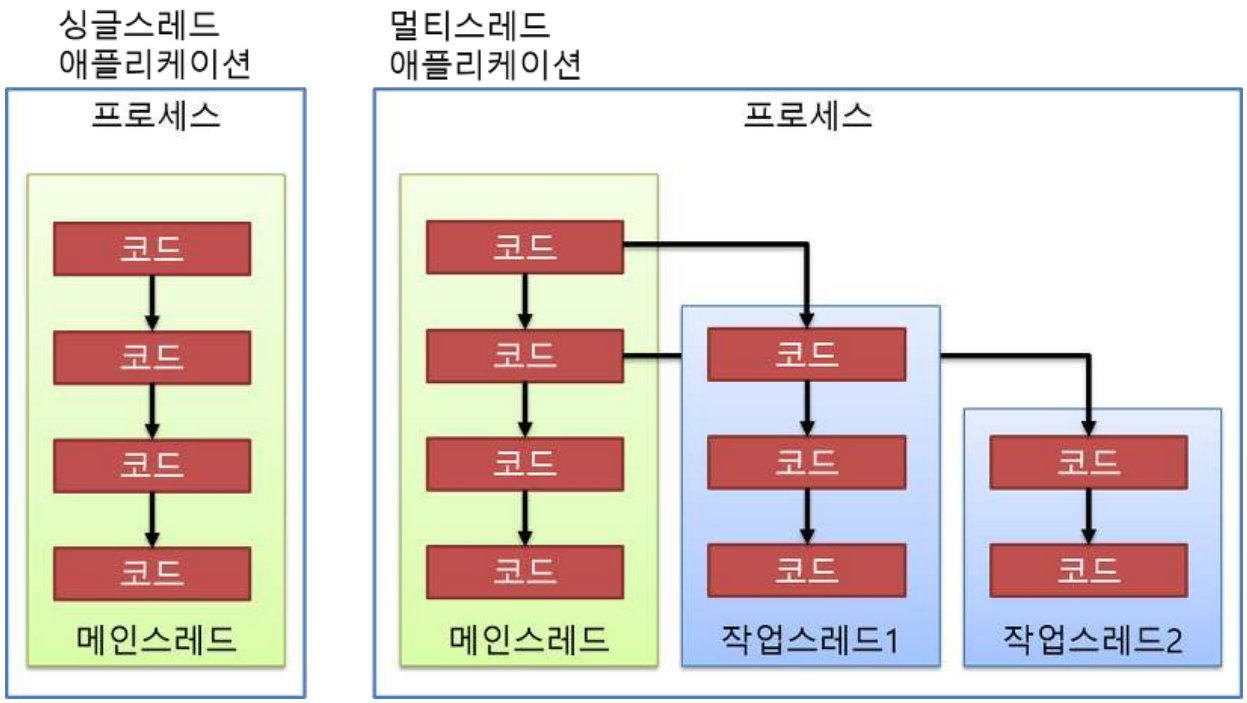
Chapter 23

멀티 스레드

멀티 스레드란?

멀티 태스킹이란 두가지 작업을 동시에 처리하는 것을 뜻합니다.
한 프로그램 안에서 멀티 태스킹을 동시에 하는 애플리케이션들도 있습니다.
ex) 메신저
스레드는 프로그램의 실행흐름 입니다.
프로그램 내에 스레드가 2개라면 두 개의 코드 흐름이 생긴다는 뜻입니다.



스레드 사용방법

방법1

1. Runnable인터페이스를 상속받아 run() 추상메서드를 재정의 하고
2. Thread객체의 생성자의 매개값으로 전달합니다.
3. 그 다음 start()메서드를 통해 실행시킵니다

방법2

1. Thread클래스를 상속받아 run() 추상메서드를 재정의 하고
2. Thread객체의 생성자의 매개값으로 전달합니다.
3. 그 다음 start()메서드를 통해 실행시킵니다.

두 개의 차이는 무엇인가요?

- Runnable은 인터페이스 이므로, 스레드 실행의 기능만 정의되어 있으므로 클래스 내부에서 사용할 수 있는메서드가 한정적 입니다. (스레드의 static 메서드는 전부 사용 가능하겠죠?)
- Thread클래스의 기능을 전부 받아서 사용 할 수 있습니다.

스레드 기본 메서드

start() – 스레드 클래스를 실행
(static)currentThread() – 현재 스레드를 반환
getName() – 스레드 이름 반환
(static)sleep() – 스레드를 잠시 멈춤
yield() – 스레드 실행을 다른 스레드에게 양보
join() – 해당 스레드를 우선 실행

구현 클래스 스레드의 실행

메인 스레드

```
public static void main(String[] args) {

	ThreadTest threadTest = new ThreadTest();

	Thread thread = new Thread(threadTest, "A");
	thread.start();

	System.out.println("MainClass종료");

}
```

Runable을 구현한 스레드 객체

```
public class ThreadTest implements Runnable {

@Override
public void run() {

	System.out.println(Thread.currentThread().getName());
	System.out.println("ThreadTest시작");

	for (int i = 0; i <= 10; i++) {
		System.out.println(i);

		try {
			Thread.sleep(500);
		} catch (Exception e) {}
	}
}

}
```

1

구현했다면, static메서드의 기능만 사용할 수 있다

Thread상속 클래스의 실행

```
public static void main(String[] args) {

	ThreadTest threadTest = new ThreadTest();
	threadTest.setName("A");
	threadTest.start();


	System.out.println("MainClass종료");
}
```

1

```
public class ThreadTest extends Thread {

@Override
public void run() {

	System.out.println(getName());
	System.out.println("ThreadTest시작");

	for (int i = 0; i <= 10; i++) {
		System.out.println(i);
		try {
			sleep(500);

		} catch (Exception e) {}
	}
}
}
```

2

상속받았다면 스레드 기능을 전부 사용할 수 있다 이해하자.

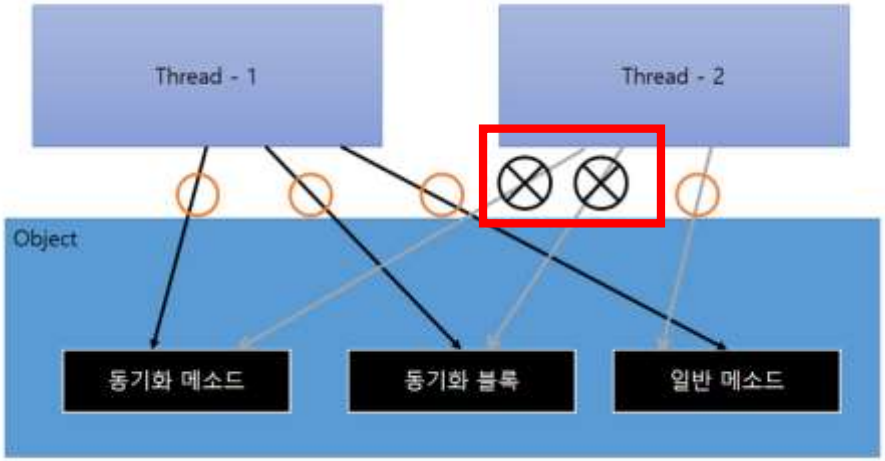
동기화 메소드 synchronized

스레드가 사용중인 객체를 다른 스레드가 변경할 수 없도록 할 때
객체에 lock을 걸어 다른 스레드가 건들 수 없도록 할 수 있습니다

메서드의 선언에 synchronized를 붙이면 됩니다. (일반메서드, static메서드 모두 적용 가능)

```
@Override
public synchronized void run() {
        …내용…
}
```

synchronized가 선언되면 해당 객체가 사용될 때, 다른 곳에서 호출 할 수 없습니다



Chapter 23

수고하셨습니다