24회차 복습 *** 헷갈리는 부분 확실히 정리하고 넘어갈것

지역변수, 전역변수

지역변수와 전역변수는 선언되는 위치에 따라 그 유형이 결정된다. **지역변수는 중괄호({}) 안에서 선언된 변수**를 말한다.

이렇게 선언된 지역변수는 선언된 지역 내에서만 유효하다. **전역변수는 특정 지역(중괄호)밖에서 선언된 변수**를 말한다.

말그대로 전체에서 어디서든 호출하면 사용할수 있는 변수는 전역변수

특정한 구역({ })안에 생성되어 그 지역에만 사용할수 있는 변수는 지역변수라고 생각하면 된다.

내가 헷갈렸던 것 = 지역변수 혹은 전역변수 선언할때 기준

어떤건 지역변수로 선언하고 어떤건 전역변수로 선언해야겠다 나누는거

전역변수도 종류가 있다.
1.객체변수 == 인스턴스 변수
객체변수는 클래스영역에서 선언되며 클래스의 객체를 생성할 때 만들어진다. 즉 객체화를 시켜서 호출해야지만 사용이 가능

2.클래스변수 == static 변수
객체화를 시키지 않고도 사용이 가능하며 객체변수가 객체화 시킬때마다 서로 다른 저장공간을 가지는 반면
클래스변수는 여러번 객체화 시켜도 공통적인 저장공간을 가지게 된다.

게임 캐릭터의 예를 들면, 캐릭터별로 키를 다르게 가질 수는 없고 모든 캐릭터가 2m의 키를 갖도록 고정한다고 하면,
이 키는 클래스 내 모든 객체가 똑같이 가지는 속성이므로 클래스변수가 된다. 일반적으로 static이 붙어있으면 클래스변수라 할 수 있고,
이 클래스 변수는 클래스 내 모든 영역에 영향을 끼치는 전역함수의 성질을 가짐. ㅇㅋ??

★★

전역변수가 장점이 많아 보이는 전역변수를 더 많이 사용하는 게 좋을 것 같지만, 그렇지 않다. 코드가 길어지고 프로그램이 복잡해질수록 많은 함수를 사용하게 되는데, 전역함수를 사용하면 프로그램에 대해 정확히 파악할 수 없기 때문에 나중에 프로그램을 유지/보수하기 힘들어진다. 또한, 지역변수와 전역변수 중에 이름이 같은 변수가 있을 경우 지역변수를 우선적으로 접근하다. 코드가 길어질수록 겹치는 이름의 변수가 많아지므로 의도하지 않은 결과가 나올 가능성이 높아진다.

Thread의 동기화

멀티스레드 프로그래밍을 하다보면 스레드를 동기화 해주어야 한다. 스레드를 동기화 하는 이유는 여러 개의 스레드가 같은 프로세스 내의 자원을 공유하면서 작업하는 경우에 서로의 작업이 다른 작업에 영향을 주기 때문이다.
스레드의 동기화를 위해서 임계영역(critical section)과 잠금(lock)을 사용한다.
임계영역을 지정하고, 임계영역이 가지고 있는 lock을 단 하나의 스레드에게 빌려주는 개념이다.
lock은 단 하나의 스레드만이 가질 수 있고, 임계영역에 내에서 수행할 코드를 수행한 이후에는 lock을 반납하여야 한다.

스레드를 동기화하는 방법들

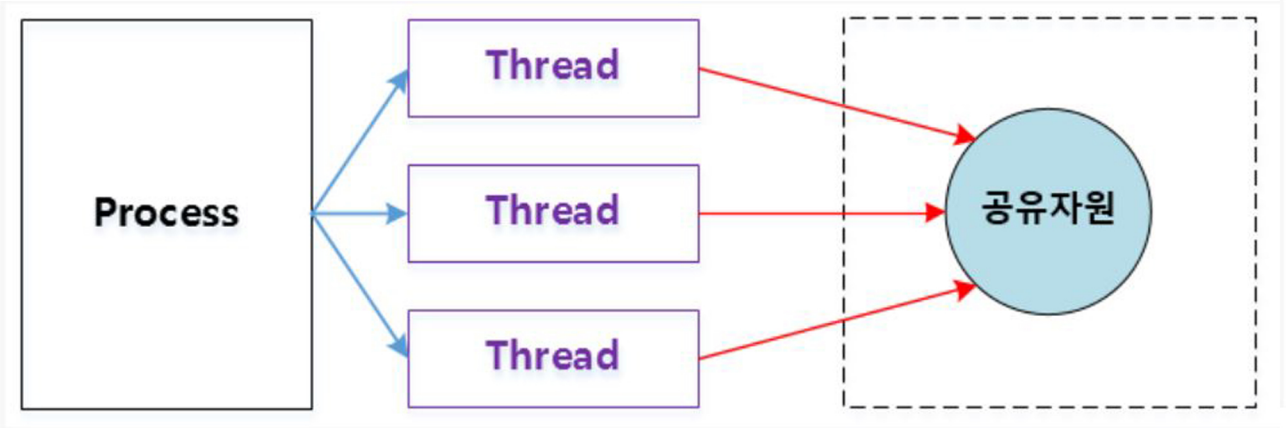
방법	설명
임계영역(critical section)	공유 자원에 대해 단 하나의 스레드만 접근하도록 한다. (하나의 프로세스에 속한 스레드만 가능하다)
뮤텍스 (mutex)	공유 자원에 대해 단 하나의 스레드만 접근하도록 한다. (서로 다른 프로세스에 속한 스레드도 가능)
이벤트 (event)	특정한 사건의 발생을 다른 스레드에게 알린다.
세마포어 (Semaphore)	한정된 개수의 자원을 여러 스레드가 사용하려고 할 때 접근을 제한한다.
대기 가능 타이머 (waitable timer)	특정 시간이 되면 대기중이던 스레드를 깨운다.

synchronized를 이용한 동기화

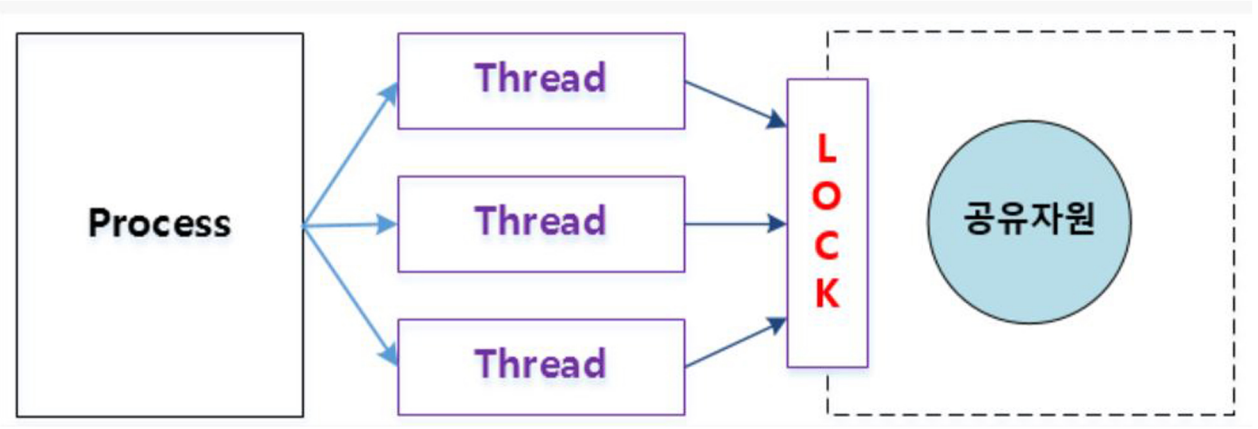
스레드를 동기화하는 방법 중 가장 간단한 방법은 synchronized를 이용해서 임계영역을 설정하는 것이다.

1. 메서드 전체를 임계 영역으로 설정한다.
2. 특정한 영역을 임계 영역으로 설정한다.(객체의 참조변수)

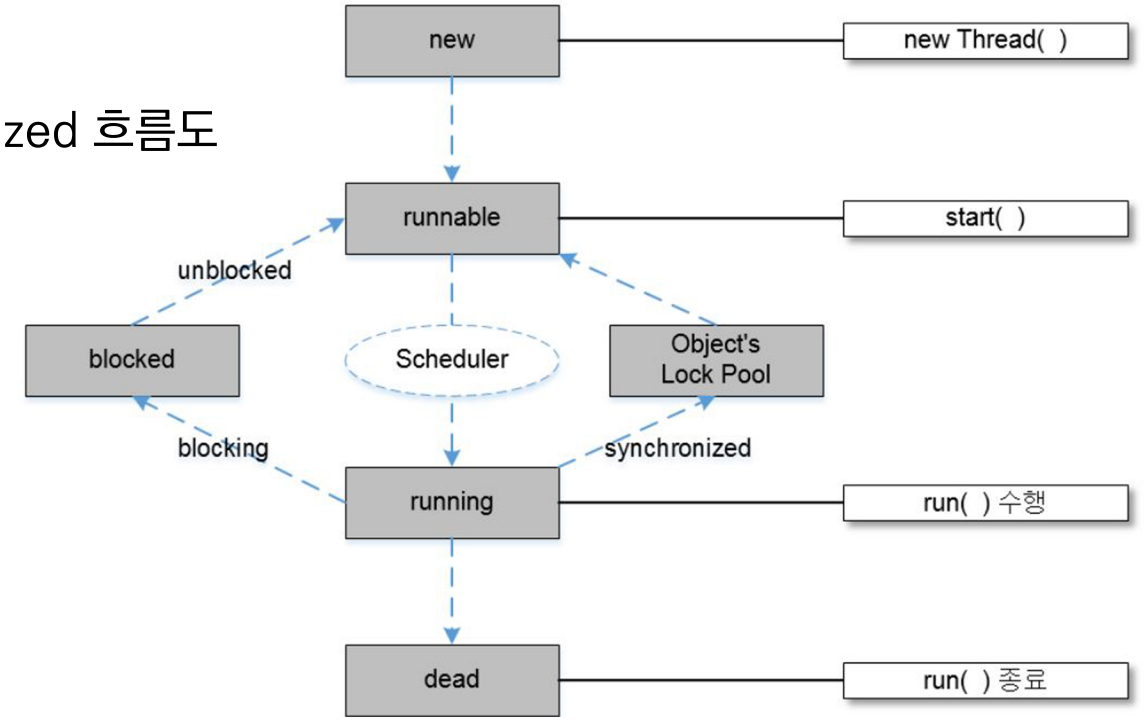
동기화 문제 발생



동기화 문제 처리



synchronized 흐름도



* 관련용어 *
공정 : 여러개의 스레드가 하나의 자원을 사용하기 위해 동시에 접근하는 프로그램을 작성할 경우 모든 스레드는 공정하게 그 자원을 사용할 수 있도록 해줘야됨.
기아상태 : 하나 or 그 이상 스레드가 자원 얻으려고 Blocked 상태에 있고 그 자원을 못얻으면 다른작업 못하는거
교착상태 : 두개 이상 스레드가 만족 못한상태로 계속 기다릴때 발생하는거
경쟁상태 : 여러 스레드가 lock얻으려고 서로 경쟁하는거

★ ㄱㄴㄲ Thread 사용할때 1.가급적 공유자원 ㄴ 2.만약 만들면 반드시 동기화 ㄱ ㅇㅋ? 넹.....