

Object Pool Pattern

결론 및 정리

정의

Object Pool Pattern은 자주 사용되는 객체들을 풀(pool)에 저장하고 필요할 때 풀에서 가져다 사용한 후 다시 풀에 반환하는 패턴입니다. 이 패턴은 객체 생성과 소멸에 대한 비용이 크거나, 초기화에 시간이 많이 걸리는 객체들에 대해 성능 향상을 도모하는 데 유용합니다.

1. **Object Pool**: 객체들을 관리하는 풀입니다. 사용 가능한 객체들의 목록과 사용 중인 객체들의 목록을 관리합니다.
2. **Reusable Object**: 풀에 저장되는 객체입니다. 이 객체는 생성 비용이 크거나 초기화에 시간이 많이 걸리는 특성을 가질 수 있습니다.

주요 동작:

- **"사용할 객체 type" acquire()**: 풀에서 사용 가능한 객체를 가져옵니다. 사용 가능한 객체가 없으면 새로운 객체를 생성하거나 요청을 대기시킬 수 있습니다.
- **void release("사용이 끝난 객체")**: 사용이 끝난 객체를 풀에 반환합니다.

구현

1. 자주 사용하는 객체 (Reusable Object) 에 대한 Collection 을 가지고 있으며, acquire / release 에 관한 함수를 가지고 있는 object pool class 를 만듭니다.
2. 사용하는 class 는 해당 object pool 을 생성하여 혹은, 외부의 object pool 을 이용하여 object 를 이용합니다.

Context & Problem & Merit

- 객체의 생성과 소멸에 비용이 크거나, 초기화에 시간이 오래 걸리는 경우에 Object Pool Pattern을 고려합니다.
 - 동시에 많은 수의 클라이언트가 동일한 객체를 요청할 때, 객체를 재사용함으로써 시스템의 성능을 향상시키고자 할 때 사용됩니다.
1. **성능 향상**: 객체를 재사용함으로써 객체 생성과 소멸에 대한 비용을 줄일 수 있습니다.
 2. **자원 관리**: 한정된 수의 객체를 효과적으로 관리하고 재사용할 수 있습니다.
 3. **스케일링**: 동시에 많은 요청이 발생할 때 시스템의 안정성을 유지할 수 있습니다.

