**Object Pool Design Pattern:**

The Object Pool Design Pattern is a [creational design pattern](https://www.geeksforgeeks.org/creational-design-pattern/) that manages a pool of reusable objects to minimize the overhead of creating and destroying objects. It maintains a collection of initialized objects and provides mechanisms for clients to efficiently borrow and return objects from the pool.

**What is an Object Pool Design Pattern?**

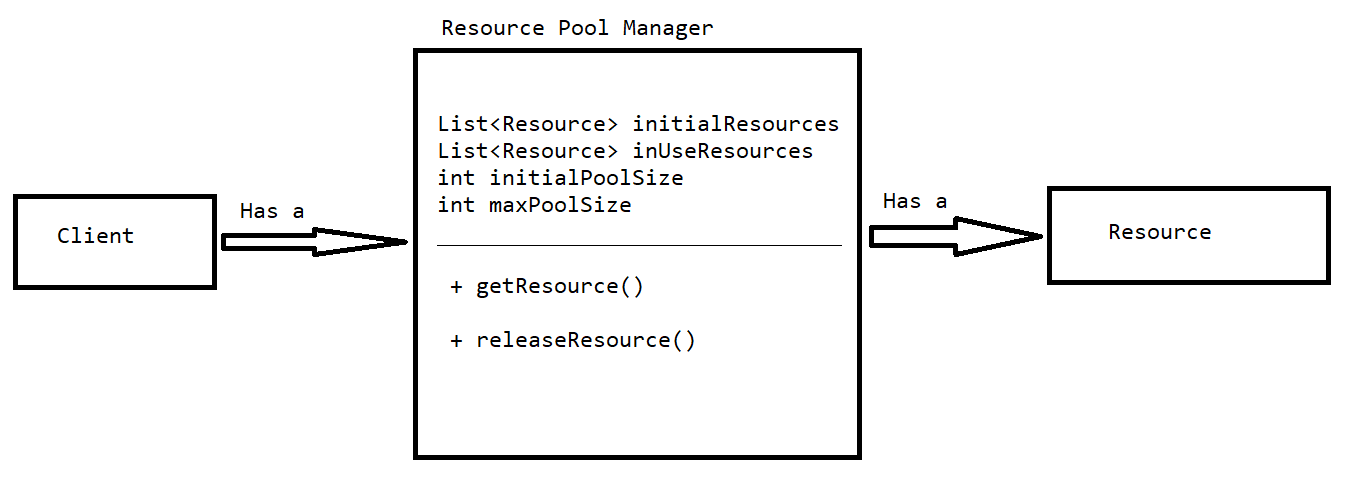
The Object Pool design pattern is a creational pattern that manages a pool of reusable objects, thus reducing the overhead of object creation and destruction. In this pattern, a pool of objects is created and maintained, and clients can borrow objects from the pool when they need them and return them to the pool when they’re done

* An Object pool is a container that contains some amount of objects.
* So, when an object is taken from the pool, it is not available in the pool until it is put back.

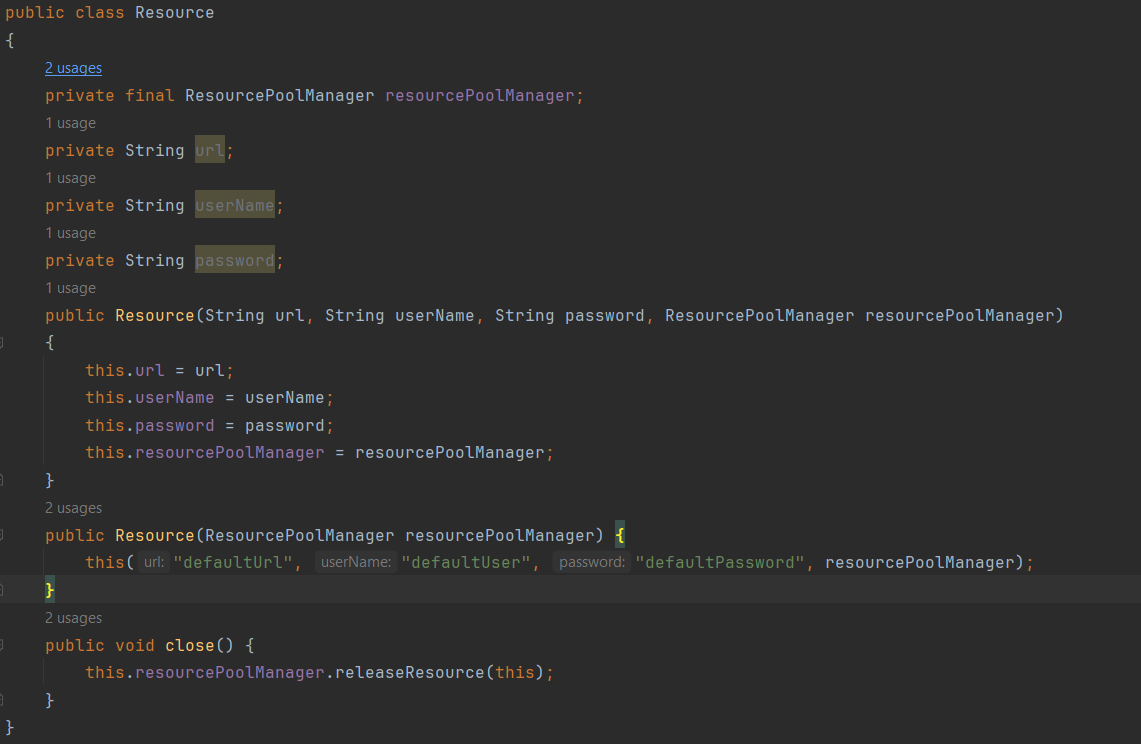
**What is the Object Pool Life Cycle?**

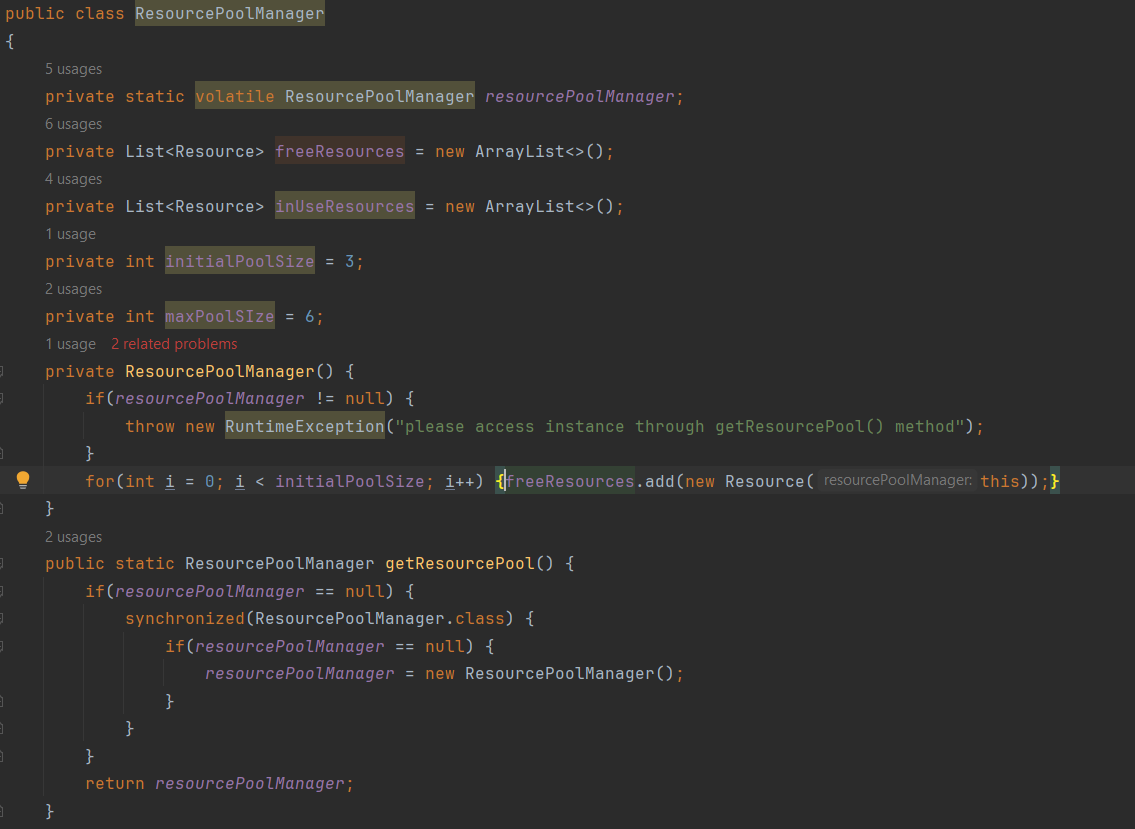
The lifecycle of objects in an object pool involves the following stages:

* **Stage 1: Creation**: Objects are initially created and added to the pool.
* **Stage 2: Borrowing**: Clients request and borrow objects from the pool.
* **Stage 3: Usage**: Clients use the borrowed objects for their tasks.
* **Stage 4: Returning**: After usage, clients return the objects to the pool for reuse.
* **Stage 5: Rejection or Destruction**: If the pool is full or objects are not used, they may be rejected or removed from the pool.
* Manages the pool of reusable objects like pool of DB connection object.
* Borrow from the pool -> use it -> then return it.

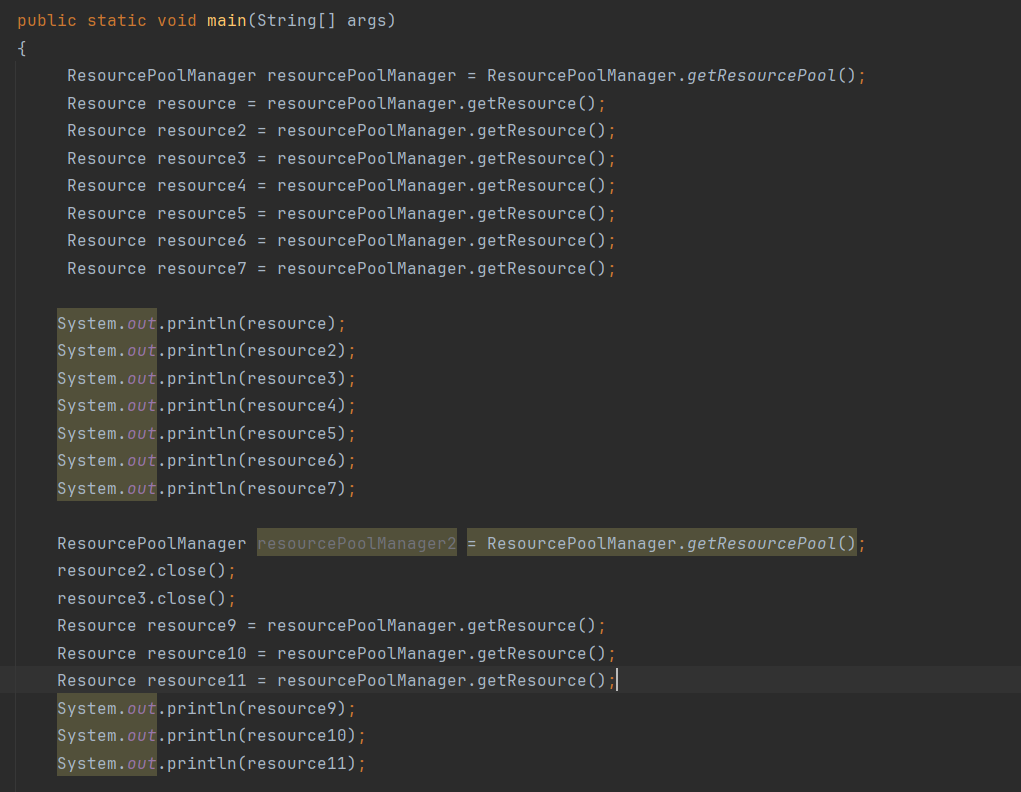


* In Object pool Design pattern Resource pool manger should be a singleton, if it not singleton then we can create n number of pool manager objects & create n\*maxPoolSize number of connections.
* To get connection & release connection should maintain thread safety, if we not make the methods are synchronized then multiple threads can access the same resource it can concurrency.
* Object pool design pattern is used with singleton design pattern and required thread safety while acquiring and releasing the resources.









**Advantages:**

* Reduces the overhead of creating the objects & destroying the frequently required objects(generally resource intensive objects).
* Reduce the latency, as it uses the pre initialized object.
* Prevent resource exhaustion by managing the number of resource intensive object creation.

**Disadvantages:**

* Resource leakage happen, if object is not handled properly & not being returned to the pool.
* Required more memory because of managing the pool.
* Pool management required thread safety which is additional overhead.
* Adds application complexity because of managing the pool.