

Pro Java №28

В Java есть несколько классов синхронизаторов, которые помогают управлять синхронизацией потоков и обеспечивать правильную координацию доступа к общим ресурсам.

Класс `Semaphore` из пакета `java.util.concurrent` предоставляет механизмы для управления доступом к общим ресурсам с помощью набора разрешений. Каждый вызов метода `acquire()` уменьшает число разрешений на единицу, а каждый вызов метода `release()` увеличивает число разрешений, освобождая их для других потоков.

Класс `CountDownLatch` позволяет потокам находиться в стадии ожидания до тех пор, пока не будут выполнены определенные действия.

`CountDownLatch` синхронизирует потоки, гарантируя, что все они будут ожидать завершения последнего потока. Метод `await()` позволяет потокам ожидать, а метод `countDown()` используется для уменьшения счетчика. В случае необходимости принудительного сброса можно использовать метод `reset()`.

Класс `CyclicBarrier` используется для синхронизации группы потоков, позволяя им все вместе достигать определенной точки (барьера) и продолжать выполнение только после того, как все потоки достигнут этого барьера. Когда определенное количество потоков достигает барьера, выполняется заданная операция (например, печать сообщения), и потоки продолжают выполнение.

Класс `Exchanger` из пакета `java.util.concurrent` используется для обмена данными между двумя потоками. Метод `exchange()` позволяет каждому потоку передать объект другому потоку и получить взамен объект, переданный вторым потоком.