Rewrite the transferFunds() method below to use j.u.c. locks with calls to tryLock (see slide 8) instead of synchronized. Your goal is to write a correct implementation that never deadlocks, unlike the buggy version below (which can deadlock). Assume that each Account object already contains a reference to a ReentrantLock object dedicated to that object e.g., from.lock() returns the lock for the from object. Sketch your answer below using pseudocode.

```java
public void transferFunds(Account from, Account to, int amount) {
    synchronized (from) {
        synchronized (to) {
            from.subtractFromBalance(amount);
            to.addToBalance(amount);
        }
    }
}
```