```
public class BankingExample {

  public static final int MAX_BALANCE = 1000;
  private /*@ spec_public @*/ int balance;
  private /*@ spec_public @*/ boolean isLocked = false;

  //@ public invariant balance >= 0 && balance <= MAX_BALANCE;

  //@ assignable balance;
  //@ ensures balance == 0;
  public BankingExample() { balance = 0; }

  //@ requires 0 < amount && amount + balance < MAX_BALANCE;
  //@ assignable balance;
  //@ ensures balance == \old(balance + amount);
  public void credit(int amount) { balance += amount; }

  //@ requires 0 < amount && amount <= balance;
  //@ assignable balance;
  //@ ensures balance == \old(balance) - amount;
  public void debit(int amount) { balance -= amount; }

  //@ ensures isLocked == true;
  public void lockAccount() { isLocked = true; }

  //@   requires !isLocked;
  //@   ensures \result == balance;
  //@ also
  //@   requires isLocked;
  //@   signals_only BankingException;
  public /*@ pure @*/ int getBalance() throws BankingException {
    if (!isLocked) { return balance; }
    else { throw new BankingException(); }
  }
}
```