Bank account module

Implement and expand IBankAccountHandler to create a Bank Account Handler module (not an application). Add methods and classes as needed. IBankAccountHandler is more of a beginning to a possible interface, make any changes you feel appropriate.

- Keep data in memory (no database or other storage)
- Add support for retrieving account information
- Add support for retrieving account history - 10 latest transactions shall be available
- Add support for retrieving accounts, in account balance order, for those accounts with balance exceeding specified balance limit
- Negative account balance is allowed
- Add support for registering `IBankAccountEventListener's` that will be notified when an account balance becomes negative, calls to registered listeners should be done in separate thread
- Error handling shall exist
- Log relevant information to file (using Log4J or other log framework)
- Simultaneous requests shall be supported (thread safe)
- Unit tests shall exist

```
public interface IBankAccountHandler {
    /**
     * Create new account.
     * @return Bank account id for created acconut
     */
    int createBankAccount();

    /**
     * Withdrawal from account.
     * @param bankAccountId Account id
     * @param amount The amount to withdrawal
     * @return Account balance after withdrawal
     */
    float withdrawal(int bankAccountId, float amount);

    /**
     * Deposit to account.
     * @param bankAccountId Account id
     * @param amount Amount to deposit
     * @return Account balance after deposit
     */
    float deopsit(int bankAccountId, float amount);

    // TODO: Add methods for additional functionality
}

public interface IBankAccountEventListener {
    void onNegativeAccountBalance(int bankAccountId, float balance);
}
```