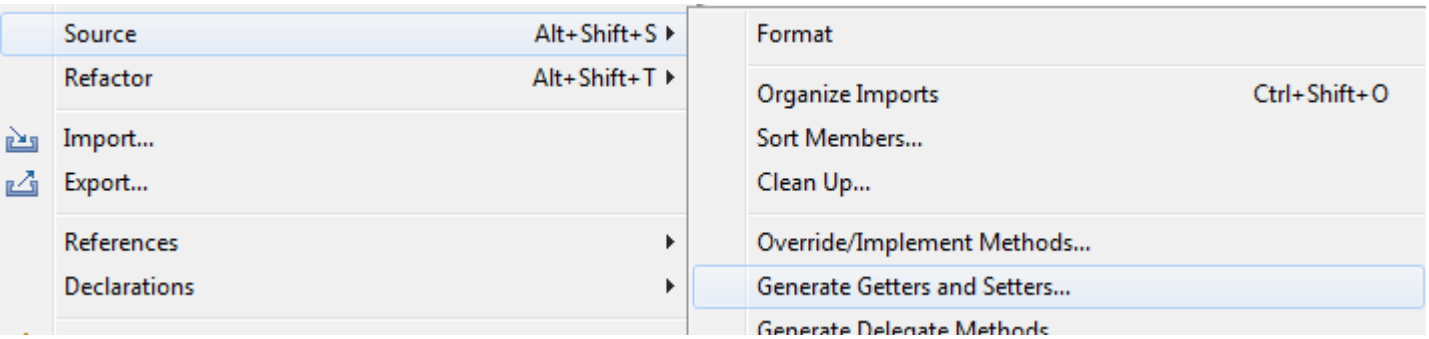
Eclipse and Java for Total Beginners

CTRL-I	Format Indentation
CTRL-O	Outline
CTRL-SPACE	Code Assist
CTRL-SHIFT-P	Find Matching Curly
CTRL-D	Delete Line

Lesson 11

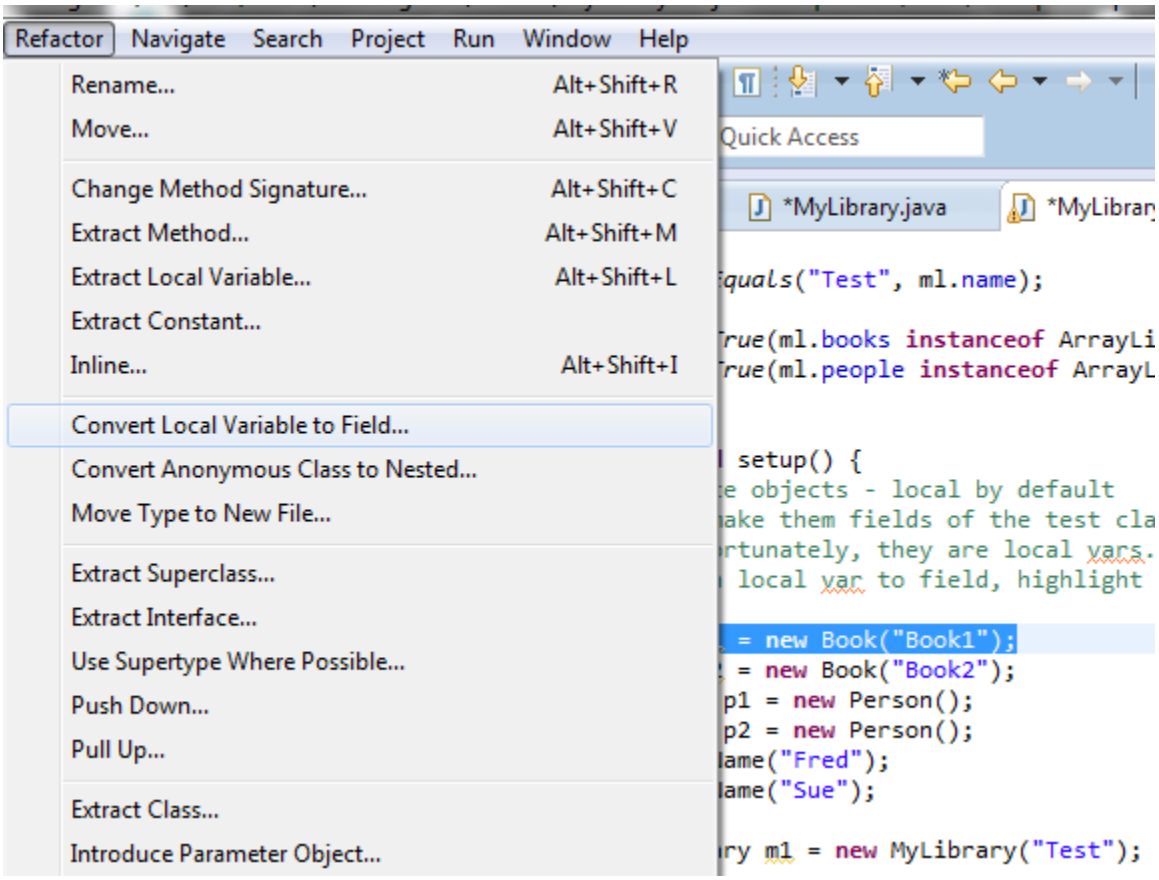
MyLibrary Methods

- "get" methods for fields (don't need setters)
- addBook, addPerson
- removeBook, removePerson
- checkOutBook, checkInBook
- getAvailableBooks
- getCheckedOutBooks
- getBooksForPerson



To test the class, need test objects, instead of instantating an object

```
Book b1 = new Book("Book1");
Book b2 = new Book("Book2");
Person p1 = new Person();
Person p2 = new Person();
p1.setName("Fred");
p2.setName("Sue");
```



Lesson 12

checkOut Method Design

- Enter Person object into person field of Book
- Use setPerson(Person) method
- What if book is already checked out?
  - Will not allow this
- Need to test for this in checkOut method

checkOut Method Pseudo Code

1. Check that this book is not already checked out.
2. If it is not checked out, set the person field in the book to this person and let calling method know it worked.
3. Otherwise, let calling method know it didn't work.

Create Test Methods, then use correctiong to add methods, fields, classes, etc

```
    public void testCheckout() {
        // set up objects
        setep();

        ml.addBook(b1);
        ml.addBook(b2);
        ml.addPerson(p1);
        ml.addPerson(p2);

        // doing two things..
        // if false, println
        // with method call in assertTrue, we can see it's return value
        assertTrue("Book did not check out correctlly",
                ml.checkOut(b1,p1));
        assertEquals("Fred",
                b1.getPerson().getName());

        assertFalse("Book was already checked out",
                ml.checkOut(b1,p2));
        assertTrue("Book check in failed",
                ml.checkIn(b1));

    }
```

Create Test first

```
public void testCheckout() {
        // set up objects
        setup();
```

```
        ml.addBook(b1);
        ml.addBook(b2);
        ml.addPerson(p1);
        ml.addPerson(p2);

        // doing two things..
        // if false, println
        // with method call in assertTrue, we can see it's return value
        assertTrue("Book did not check out correctlly",
                        ml.checkOut(b1,p1));

        assertEquals("Fred", b1.getPerson().getName());

        assertFalse("Book was already checked out",          ml.checkOut(b1,p2));
        assertTrue("Book check in failed", ml.checkIn(b1));

        assertFalse("Book was already checked in", ml.checkIn(b1));

        assertFalse("Book was never checked in", ml.checkIn(b2));


    }
```

Use Eclipse Helpers to create Class Methods

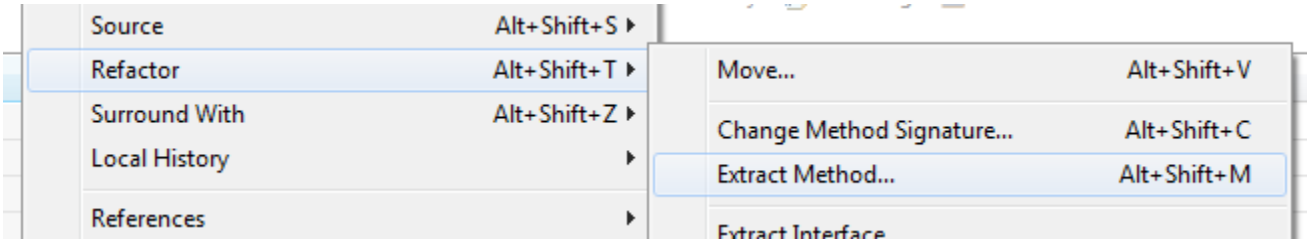
```
    public boolean checkOut(Book b1, Person p1) {
        if (b1.getPerson() == null ) {
            b1.setPerson(p1);
            return true;
        } else {
            return false;
        }
    }

    public boolean checkIn(Book b1) {
        if (b1.getPerson() != null ){
            b1.setPerson(null);
            return true;
        } else {
            return false;
        }
    }
```

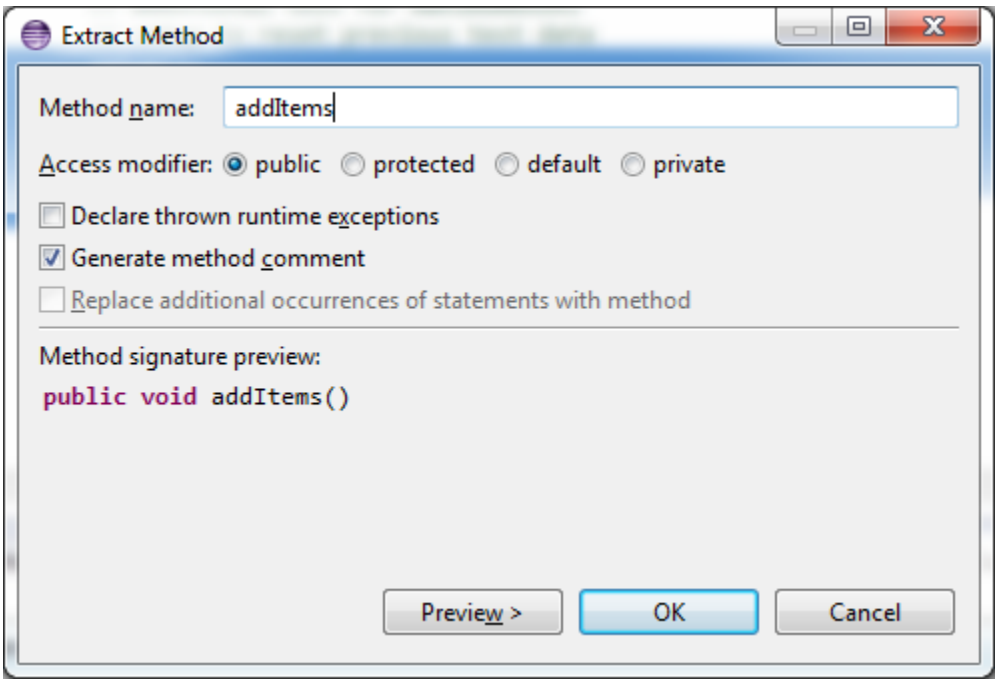
Note:

CTRL-SHIFT-P to find matching {}'s

Always using a set of statements so , extract method:

```
setup();
p1.setMaximumBooks(1);
ml.addBook(b1);
ml.addBook(b2);
ml.addPerson(p1);
ml.addPerson(p2);
```






```
addItems();
```