Carl Seiber
Check In 3

Method and Tests (also in code)

Method

```
//Carl Seiber: author of method.
/**
 * deletes an auction
 * 
 * @param NPO
 * @return True if the auction is deleted.
 */
public int deleteAuction(String NPO) {
   for (Auction a : myAuctionList) {
      if (a.getNPO().getMyName() == NPO) {
         Period days = Period.between(LocalDateTime.now().toLocalDate(), a.getAuctionDate().toLocalDate());
         int numOfDays = days.getDays();
         if (numOfDays >= 2) {
            a.getNPO().setAuction(false);
            myAuctionList.remove(a);
            return 1; //auction found and deleted successfully
         }
         return 2; //auction found but is past the deadline for auction deletion
      }
   }
   return 0; //no auction was found
}
```

Tests

```
//Carl Seiber: author of tests for deleteAuction
/**
 * Test method for {@link model.Calendar#deleteAuction(java.lang.String)}.
 */
@Test
public void testdeleteAuctionOnNoAuctionFoundWithNPO() {
   //Note: the addAuctionNoDateRestriction method was strictly created to allow scheduling of auctions earlier
   //than the one week for purposes of testing the deleteAuction. This was an exact copy of the addAuction
   //with only the valid date restriction removed and will not be used in implementation.
   myCalendar.addAuctionNoDateRestriction(a, LocalDateTime.now().plusDays(5), 10, "");
   myCalendar.addAuctionNoDateRestriction(b, LocalDateTime.now().plusDays(5), 10, "");
   assertEquals(myCalendar.deleteAuction(c.getMyName()), 0);
}

/**
 * Test method for {@link model.Calendar#deleteAuction(java.lang.String)}.
 */
@Test
public void testdeleteAuctionOnPastAuction() {
   //Note: the addAuctionNoDateRestriction method was strictly created to allow scheduling of auctions earlier
   //than the one week for purposes of testing the deleteAuction. This was an exact copy of the addAuction
   //with only the valid date restriction removed and will not be used in implementation.
   myCalendar.addAuctionNoDateRestriction(a, LocalDateTime.now().minusDays(5), 10, "");
   myCalendar.addAuctionNoDateRestriction(b, LocalDateTime.now().minusDays(5), 10, "");
   assertEquals(myCalendar.deleteAuction(c.getMyName()), 0);
}

/**
 * Test method for {@link model.Calendar#deleteAuction(java.lang.String)}.
 */
@Test
public void testdeleteAuctionOnAuctionScheduledInExactlyTwoDays() {
   myCalendar.addAuctionNoDateRestriction(a, LocalDateTime.now().plusDays(2), 10, "");
   myCalendar.addAuctionNoDateRestriction(b, LocalDateTime.now().plusDays(2), 10, "");
   assertEquals(myCalendar.deleteAuction(a.getMyName()), 1);
}
```

```java
/**
 * Test method for {@link model.Calendar#deleteAuction(java.lang.String)}.
 */
@Test
public void testdeleteAuctionOnAuctionScheduledInLessThanTwoDays() {
   myCalendar.addAuctionNoDateRestriction(a, LocalDateTime.now().plusDays(1), 10, "");
   myCalendar.addAuctionNoDateRestriction(b, LocalDateTime.now().plusDays(1), 10, "");
   assertEquals(myCalendar.deleteAuction(a.getMyName()), 2);
}

/**
 * Test method for {@link model.Calendar#deleteAuction(java.lang.String)}.
 */
@Test
public void testdeleteAuctionOnAuctionScheduledInMoreThanTwoDays() {
   myCalendar.addAuctionNoDateRestriction(a, LocalDateTime.now().plusDays(3), 10, "");
   myCalendar.addAuctionNoDateRestriction(b, LocalDateTime.now().plusDays(1), 10, "");
   assertEquals(myCalendar.deleteAuction(a.getMyName()), 1);
}
```