CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap TDA547

Lösningar till tentamen i Objektorienterad programmering

Fredagen 13 januari 2012, 14.00 – 18.00.

1. (a)
```
public class Place {
   private int row, seat;

   public Place(int row, int seat) {
      this.row = row;
      this.seat = seat;
   }

   public int getRow() {
      return row;
   }

   public int getSeat() {
      return seat;
   }
}
```

   (b) Testet `p1 == p2` testar om `p1` och `p2` refererar till *samma objekt*. Om vi har skapat två objekt med samma rad och säte, så kommer testet alltså att ge resultatet `false` i stället för det avsedda `true`

   (c) Den bästa lösningen är en metod som omdefinierar `equals` i `Object` och där alltså parametern har typ `Object`:

```
public boolean equals(Object other) {
   boolean res = false;
   if (other instanceof Place) {
       Place p = (Place)other;
       res = row == p.getRow()
             && seat == p.getSeat();
   }
   return res;
}
```

   Man får dock här också poäng för en enklare version:

```
public boolean equals(Place other) {
   return row == other.getRow()
          && seat == other.getSeat();
}
```

2. (a)
```
Selection sel = new Selection(new Place(11,7),3);
```
(b)
```
public static void printTickets(ShowData d, Selection sel) {
      int row = sel.getFirstPlace().getRow();
      int seat = sel.getFirstPlace().getSeat();
      for (int s = seat; s < seat + sel.getNrPlaces(); s++)
         printTicket(d, new Place(row,s));
   }
```

3. Både rader och säten numreras i salongen från 1 och uppåt, medan matriser i Java indiceras från 0 och uppåt. Vi låter därför platsen på rad r, säte s i salongen representeras av matriselementet `places[r-1][s-1]`:

```
public class Bookings {

   private State[][] places;
   private ShowData data;

   public Bookings(ShowData data, int nrRows, int nrSeats) {
      this.data = data;
      places = new State[nrRows][nrSeats];
      for (int r = 0; r < nrRows; r++)
         for (int s = 0; s < nrSeats; s++)
            places[r][s] = State.FREE;
   }

   private void changeState(State st, Selection sel) {
     int row = sel.getFirstPlace().getRow()-1;
     int seat = sel.getFirstPlace().getSeat()-1;
     for (int s = seat; s < seat + sel.getNrPlaces(); s++)
       state[row][s] = st;
   }

   public void book(Selection sel) {
     changeState(State.BOOKED,sel);
   }

   public void sell(Selection sel) {
     changeState(State.SOLD,sel);
   }

   public int getNrRows() {
      return places.length;
   }

   public int getNrSeats() {
      return places[0].length;
   }

   public State getState(int row, int seat) {
      return state[row-1][seat-1];
   }
}
```

4. En lösning som utnyttjar en lokal klass `Pair` kan se ut så här:

```
public class BookingList implements Reservations {

  private static class Pair {
    public final String passwd;
    public final Selection sel;

    public Pair (String passwd, Selection sel) {
      this.passwd = passwd;
      this.sel = sel;
    }
  }

  private Pair[] bookings;
  private int count;

  public BookingList(int size) {
    bookings = new Pair[size];
    count = 0;
  }

  public void reserve(String passwd, Selection sel) {
    bookings[count] = new Pair(passwd, sel);
    count++;
  }

  public Selection getReservation(String passwd) {
    for (int i = 0; i<count; i++) {
       if (passwd.equals(bookings[i].passwd)) {
           Selection res = bookings[i].sel;
           count--;
           for (int k=i; k<count; k++)
               bookings[k] = bookings[k+1];
           return res;
       }
    }
    return null;
  }
}
```

5. Endast `mouseClicked` måste definieras:

```
 public void mouseClicked(MouseEvent e) {
    int seat = e.getX()/SIZE;
    int row = e.getY()/SIZE;
    if (model.getState[row+1][seat+1] == State.FREE)
      sel = new Selection(new Place(row,seat),1);
    else
      sel = null;
    repaint();
 }
```

6.
```
public void actionPerformed(ActionEvent e) {
  Selection sel = bv.getSelection();
  if (bv != null) {
     model.sell(sel);
     TicketPrinter.printTickets(s,sel);
     bv.setSelection(null);
     bv.repaint();
  }
}
```