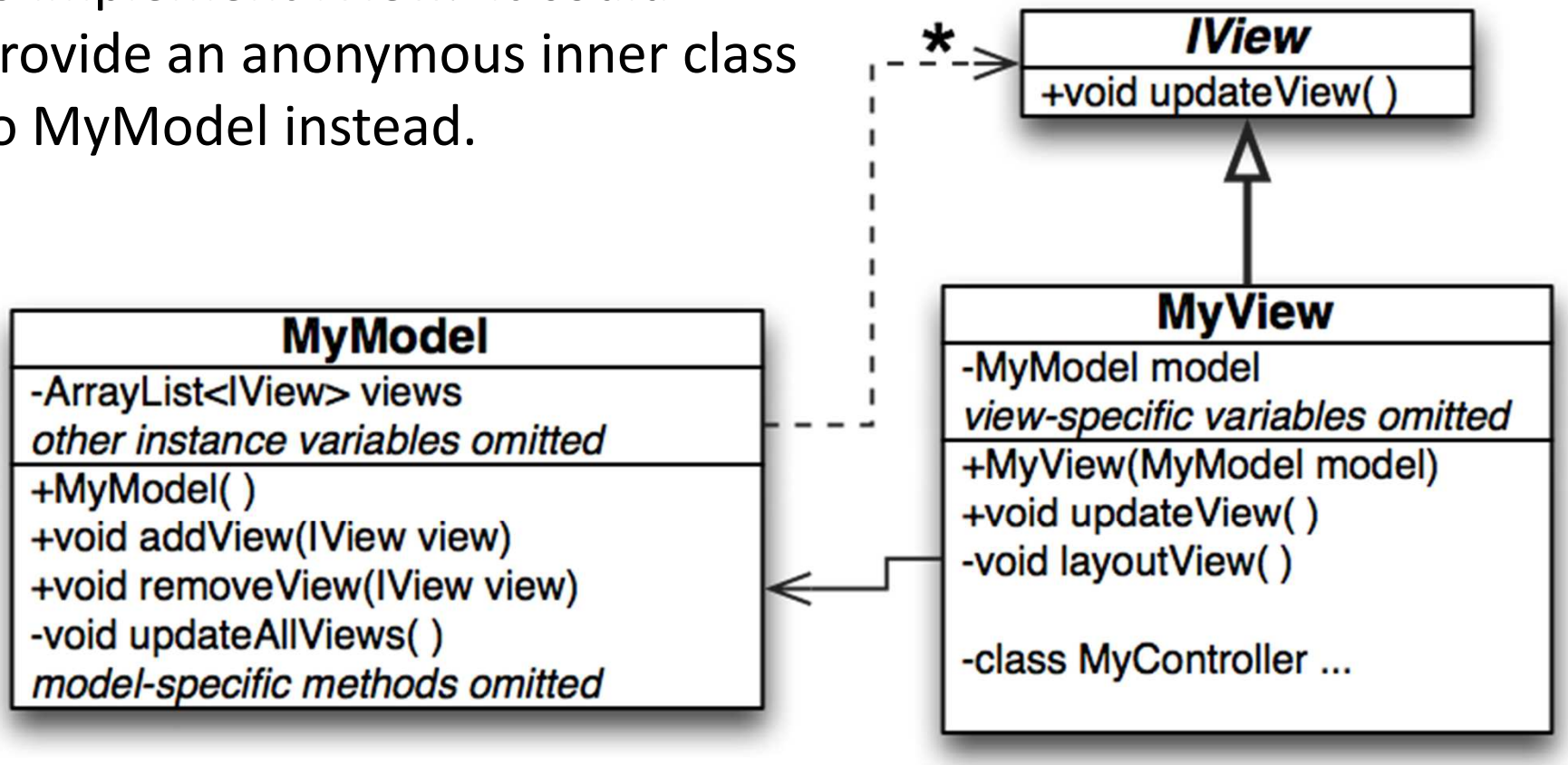
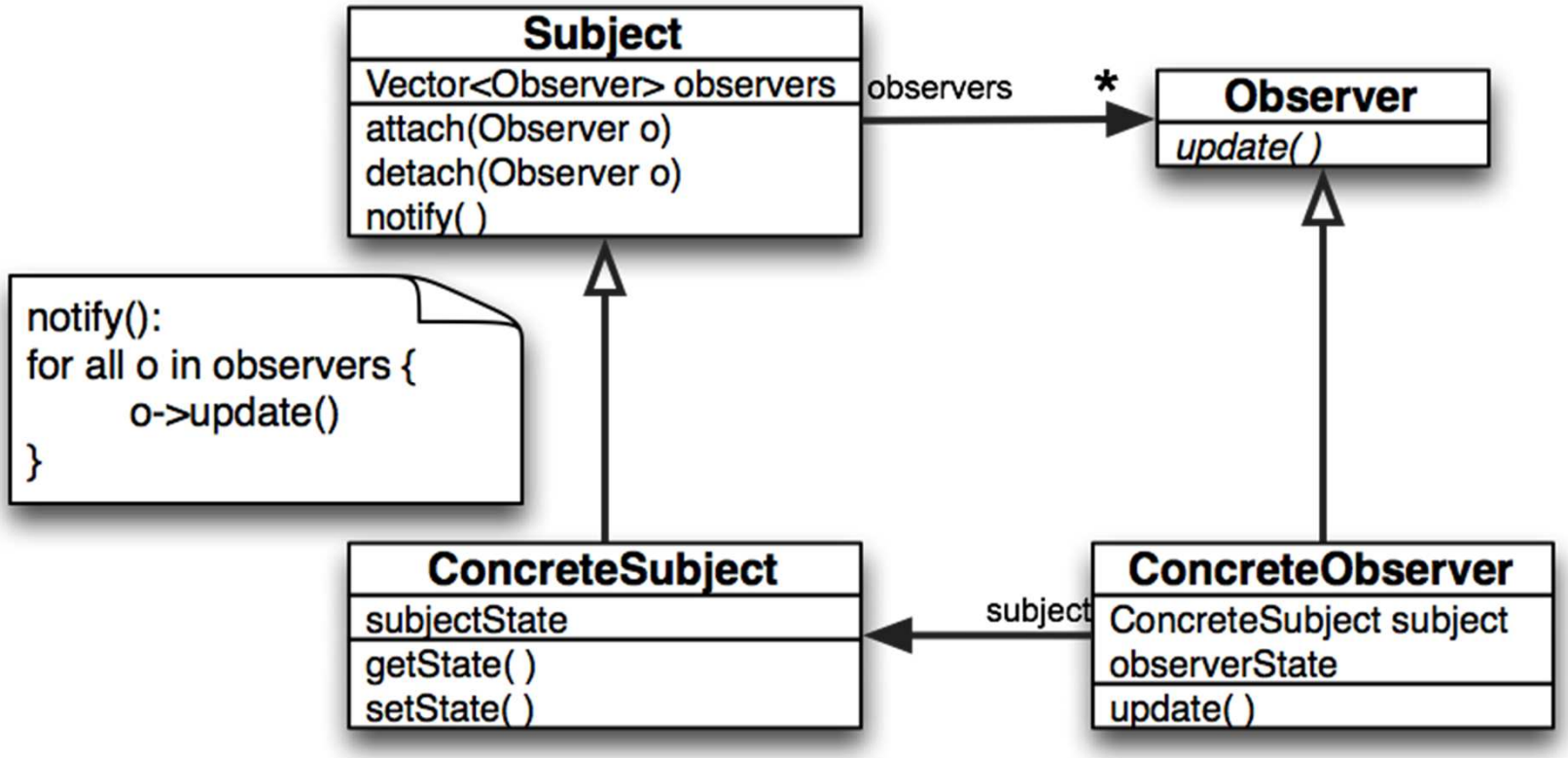
Design Patterns/MVC Reprise

MVC as UML (Java Version)

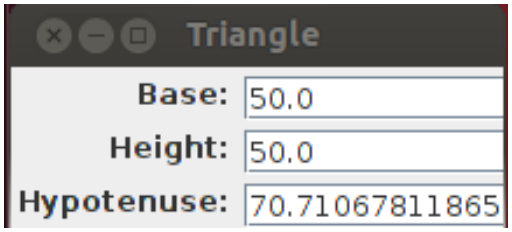
Note that MyView does not need to implement IView. It could provide an anonymous inner class to MyModel instead.

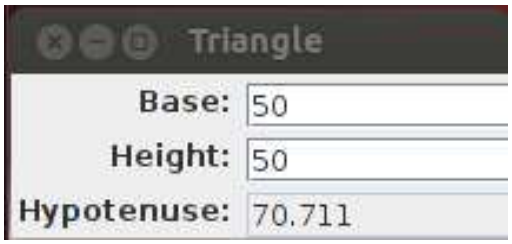


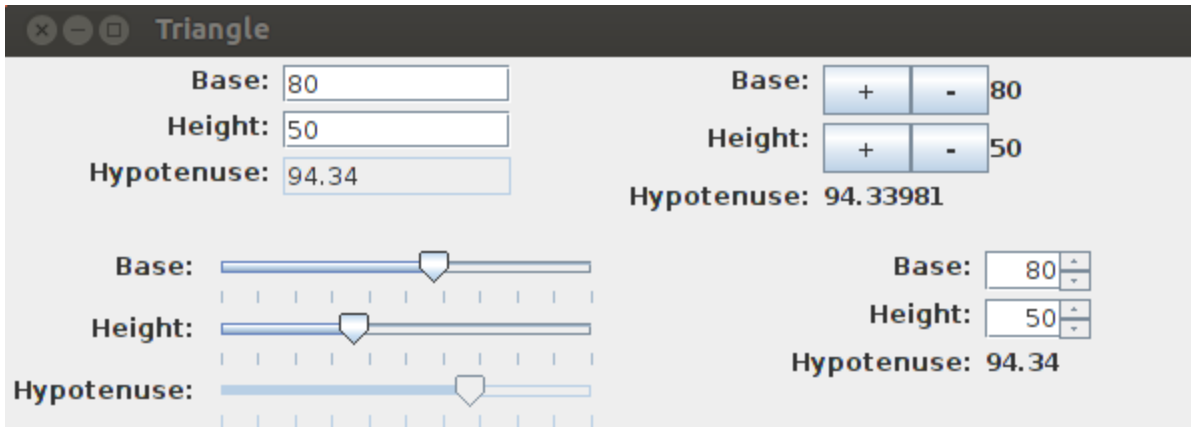
Observer Design Pattern

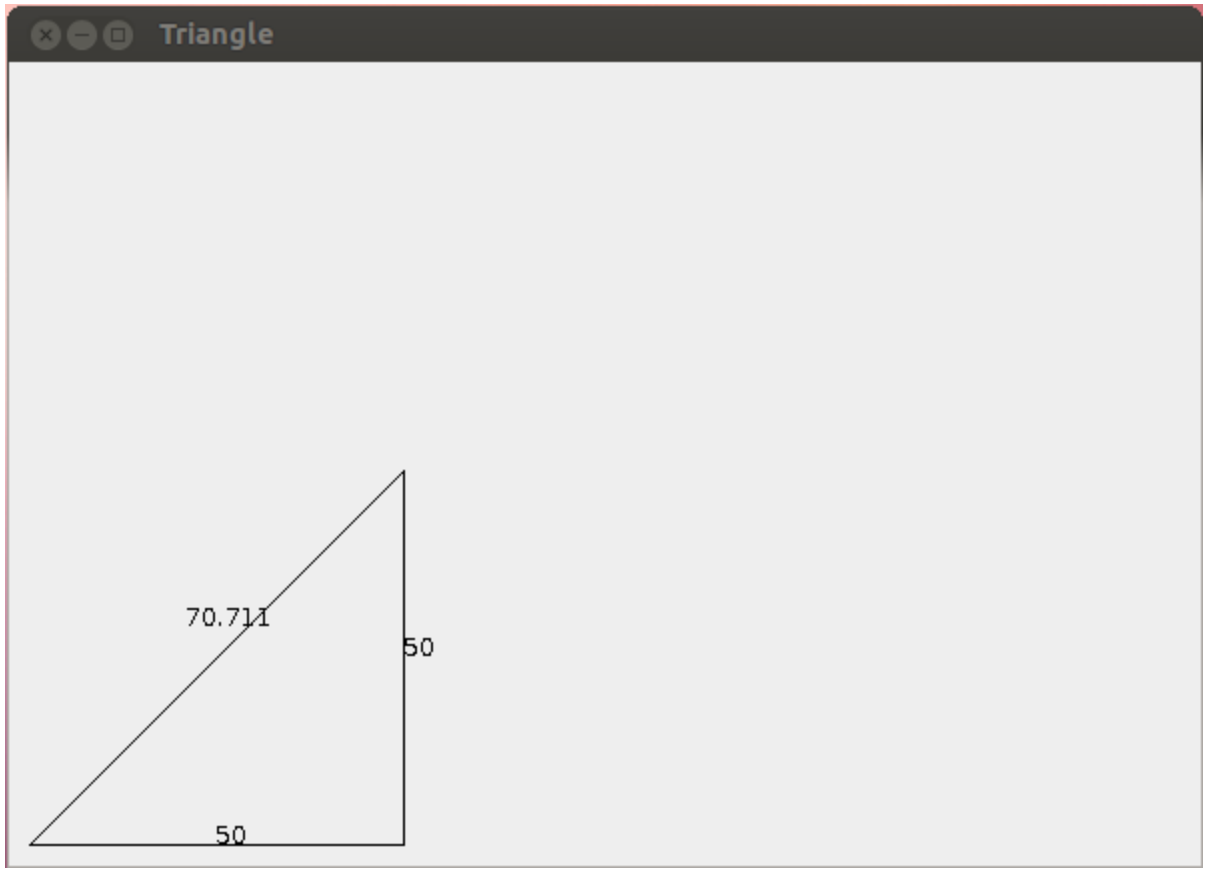


Triangle Example









Creating a Custom Table Using JTable

- Workhorse class is usually AbstractTableModel
- Need to implement 3 methods
  - public int getColumnCount();
  - public int getRowCount() ;
  - public Object getValueAt(int row, int col);
- Sets up a table with generic names for columns, and no editable columns
- To change this default behaviour, override:
  - public String getColumnName(int col);
  - public Class getColumnClass(int c);
  - public boolean isCellEditable(int row, int col);
  - public void setValueAt(Object value, int row, int col);

Benefits to AbstractTableModel

- Fires events for you in relatively easy fashion
  - void fireTableCellUpdated(int row, int column);
  - void fireTableChanged(TableModelEvent e);
  - void fireTableDataChanged();
  - void fireTableRowsDeleted(int firstRow, int lastRow);
  - void fireTableRowsInserted(int firstRow, int lastRow);
  - void fireTableRowsUpdated(int firstRow, int lastRow);
  - void fireTableStructureChanged();