

CS 349

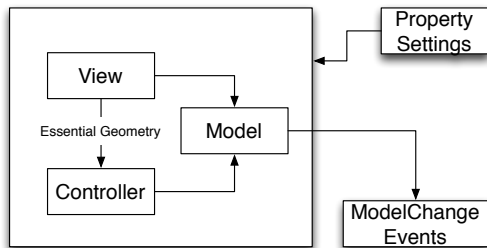
08 Abstract Models

Byron Weber Becker
Winter 2012



Simple Widgets

- Modern widget toolkits use MVC throughout
- Simple widgets usually contain a default model within themselves
- Examples: buttons, checkboxes, scrollbars, ...



Complex Widgets

- More complex widgets expect the application to implement model interface or extend an abstract class
- Examples: table and tree widgets

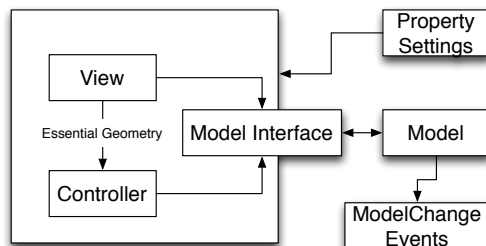


Table Model

```
public interface TableModel {
    int getColumnCount();
    String getColumnName(int columnIndex);
    Class<?> getColumnClass(int columnIndex);

    int getRowCount();

    Object getValueAt(int rowIndex, int columnIndex);
    void setValueAt(Object aValue, int rowIndex, int columnIndex);
    boolean isCellEditable(int rowIndex, int columnIndex);

    void addTableModelListener(TableModelListener l);
    void removeTableModelListener(TableModelListener l);
}
```



Table Model Listener

```
public interface TableModelListener {
    void tableChanged(TableModelEvent e);
}

public class TableModelEvent {
    TableModelEvent(TableModel source) { ... }
    TableModelEvent(TableModel source, int row) { ... }
    TableModelEvent(TableModel source, int firstRow, int lastRow) { ... }
    TableModelEvent(TableModel source, int firstRow, int lastRow, int column) { ... }
    TableModelEvent(TableModel source, int firstRow, int lastRow,
        int column, int type) { ... }

    int getColumn() { ... }
    int getFirstRow() { ... }
    int getLastRow() { ... }
    int getType() { ... } // one of INSERT, UPDATE, DELETE
}
```



Customizing JTable

- JTable has a `TableColumnModel` with information about each column in the table.
- The `TableColumnModel` has methods like:
 - `void addColumn(TableColumn aColumn)`
 - `TableColumn getColumn(int columnIndex)`
 - `int getColumnCount()`
 - `int[] getSelectedColumns()`
 - `void moveColumn(int columnIndex, int newIndex)`
- `TableColumnModel` is actually an interface; there is a `DefaultTableColumnModel` that implements the standard column-handler. It chooses appropriate `TableColumn` instances based on the type of data returned by `getColumnClass()` in the `TableModel`.



Customizing JTable

- The TableColumn gives information about a single column.
- It has methods like:
 - void setCellRenderer(TableCellRenderer cellRenderer)
 - void setCellEditor(TableCellEditor cellEditor)
 - void setPreferredWidth(int preferredWidth)
 - void setMinWidth(int minWidth)
 - void setResizable(boolean isResizable)
 - void setHeaderRenderer(TableCellRenderer headerRenderer)

Demos

- Changing table structure (adding rows)
 - JTableAddRowDemo
- Renderers and Editors
 - JTableEditorsDemo