

Модель MVC на примере JTable

MVC-схема в компонентах Swing

- *Модель* – один или несколько классов, в которых хранится и вырабатывается вся информация, связанная с компонентом, состояние объектов, созданных этим компонентом.

Для ввода и изменения информации
используются методы

`setXXX()`

MVC-схема в компонентах Swing

- *Вид* – один или несколько классов, определяющих способ представления на экране или другом устройстве результатов, полученных моделью, в определенном виде

Получают информацию из модели методами `getXXX()` и `isXXX()`

Модель сообщает видам об изменении состояния объекта методами `fireXXX()`

MVC-схема в компонентах Swing

- *Контроллер* – классы, создающие интерфейс для ввода информации и изменения состояния объекта. Они реагируют на события ввода клавиатуры, действия мыши и пр. и обращаются к методам `setXXX()` модели

MVC-схема в компонентах Swing

Вид регистрируется (подписывается) у модели. *Контроллер*, реагируя на события меняет модель. *Модель* сообщает (рассылает) видам об изменении состояния. *Вид* забирает измененную информацию у модели.

Контроллер и *вид* не взаимодействуют.

MVC-схема в компонентах Swing

Модели описаны интерфейсами, в которых перечислены необходимые методы
(ButtonModel)

У каждого интерфейса есть хотя бы одна стандартная реализация
(DefaultButtonModel)

Возможны реализации в виде абстрактных классов (AbstractListModel)

MVC-схема в компонентах Swing

Для реализации модели используется делегирование полномочий.

В качестве модели данных используется представитель – экземпляр класса с именем вида `xxxModel`. Обычно это защищенное или закрытое поле.

Доступ к модели осуществляется через метод `getModel()`. Если необходимо заменить модель используется метод `setModel(xxxModel)`.

Компонент JTable

- В качестве модели используются классы-делегаты
 - TableModel
 - TableColumnModel
 - JTableHeader

Компонент JTable

- Для отображения элементов таблицы используется класс-делегат
 - TableCellRenderer

Компонент JTable

- За выделение ячеек и строк отвечает
 - ListSelectionModel
- Редактирование ячеек передается класссу
 - TableCellEditor

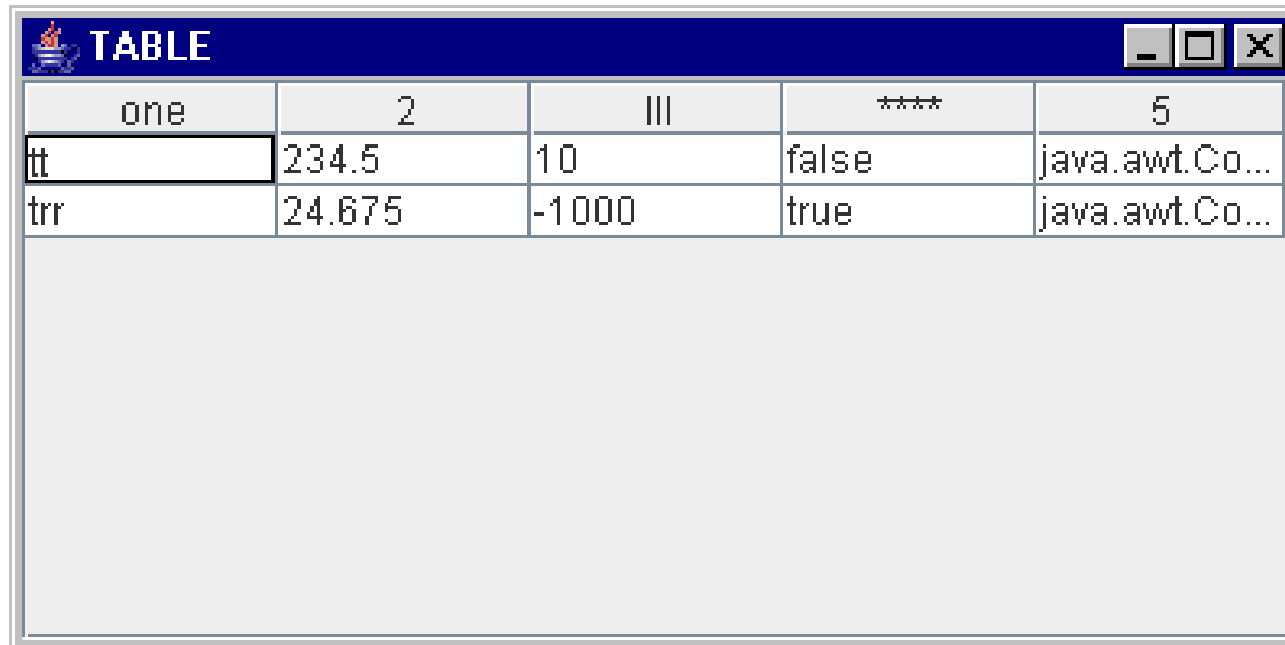
Пример таблицы (модель по умолчанию)

```
private Object[ ][ ] cells = {
    {"tt",234.5,10,false,Color.red},
    {"trr",24.675,-1000,true,Color.yellow}
};

private String[ ] columnNames = {"one","2","III","****","5"};

JTable tabl = new JTable (cells, columnNames);
add( new JScrollPane(tabl), BorderLayout.CENTER);
```

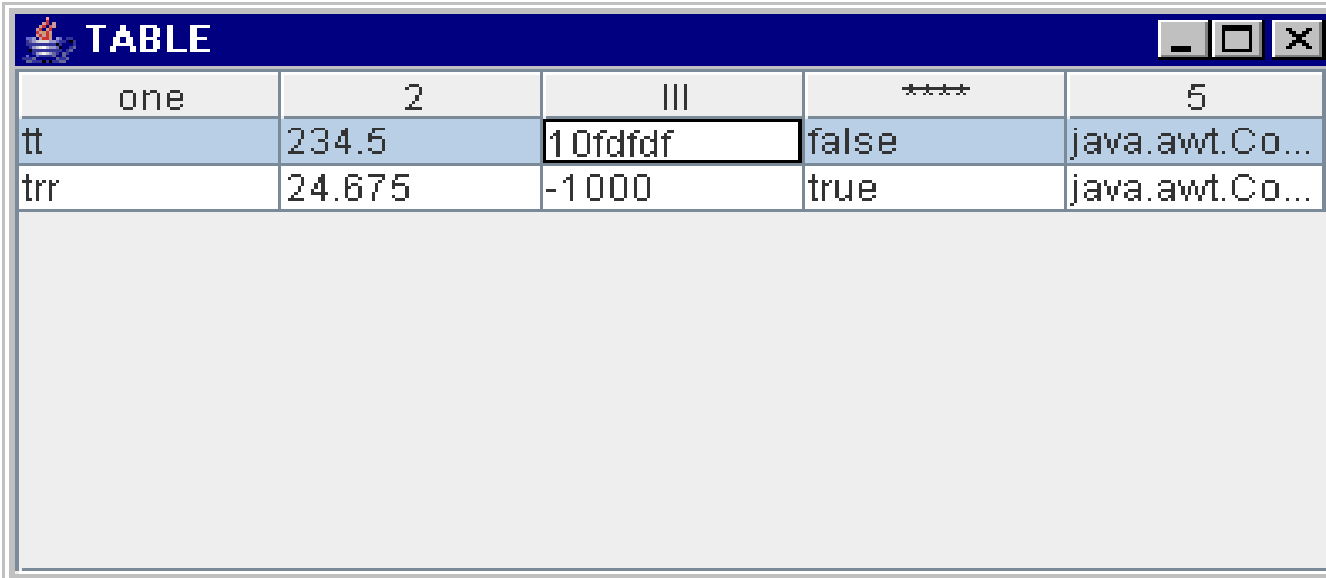
Пример



The image shows a screenshot of a Java Swing window titled "TABLE". The window contains a table with 5 columns and 2 rows of data. The first row contains the headers: "one", "2", "III", "****", and "5". The second row contains the data: "tt", "234.5", "10", "false", and "java.awt.Co...". The third row contains the data: "trr", "24.675", "-1000", "true", and "java.awt.Co...". The table is displayed in a standard Java Swing window with a blue title bar and standard window controls.

one	2	III	****	5
tt	234.5	10	false	java.awt.Co...
trr	24.675	-1000	true	java.awt.Co...

Пример



The image shows a screenshot of a Java Swing window titled "TABLE". The window contains a table with 5 columns and 2 rows of data. The first row has headers: "one", "2", "III", "****", and "5". The second row has data: "tt", "234.5", "10fdfdf", "false", and "java.awt.Co...". The third row has data: "trr", "24.675", "-1000", "true", and "java.awt.Co...".

one	2	III	****	5
tt	234.5	10fdfdf	false	java.awt.Co...
trr	24.675	-1000	true	java.awt.Co...

Собственная табличная модель

```
class FirstModel extends AbstractTableModel {  
    private Object[][] cell = { {"tt",234.5,10,false,Color.red},  
        {"trr",24.675,-1000,true,Color.yellow} };  
    private String[] columnNames = {"one","2","III","****","5"};  
    public FirstModel() { super();}  
    public int getRowCount() {return cell.length;}  
    public int getColumnCount() {return cell[0].length;}  
    public Object getValueAt (int row, int col) {return cell[row][col];}  
    public void setValueAt (Object value, int row, int col) {  
        if (col!=0) cell[row][col] = value;  
    }  
}
```

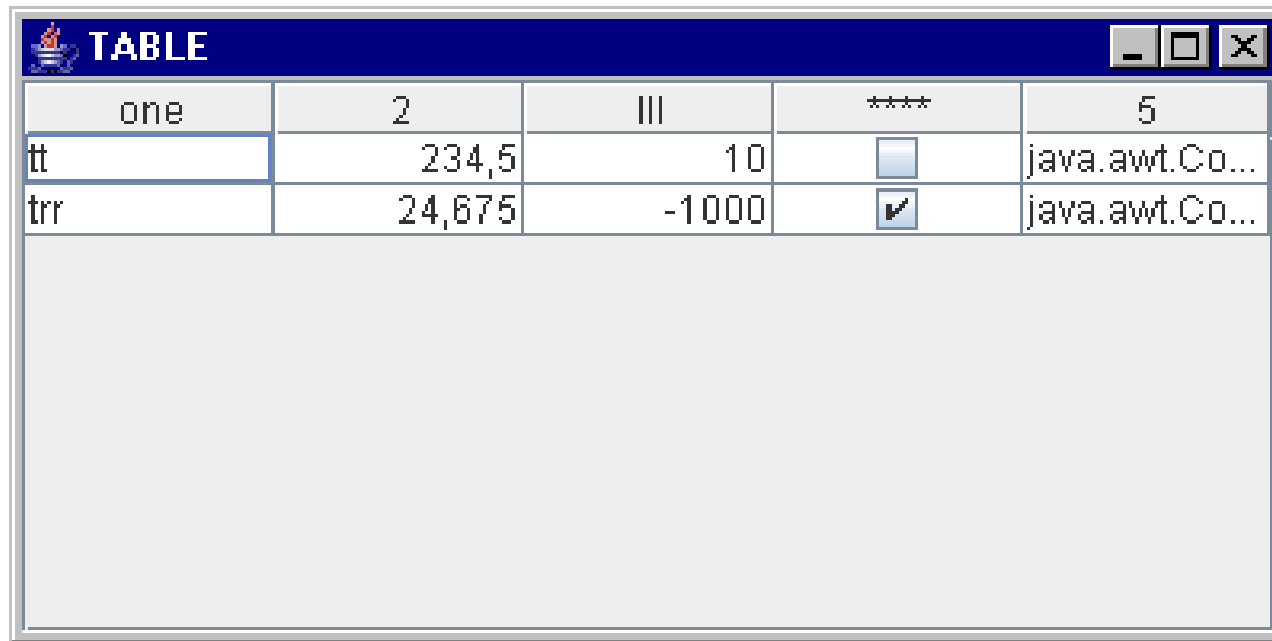
Собственная табличная модель(продолжение)

```
public boolean isCellEditable (int row, int col) {  
    return col !=0;  
}  
  
public Class getColumnClass(int col) {  
    return cell[0][col].getClass();  
}  
  
public String getColumnName(int col) {  
    return columnNames[col];  
}  
}
```

Собственная табличная модель(продолжение)

```
JTable tabl = new JTable (new FirstModel());
```

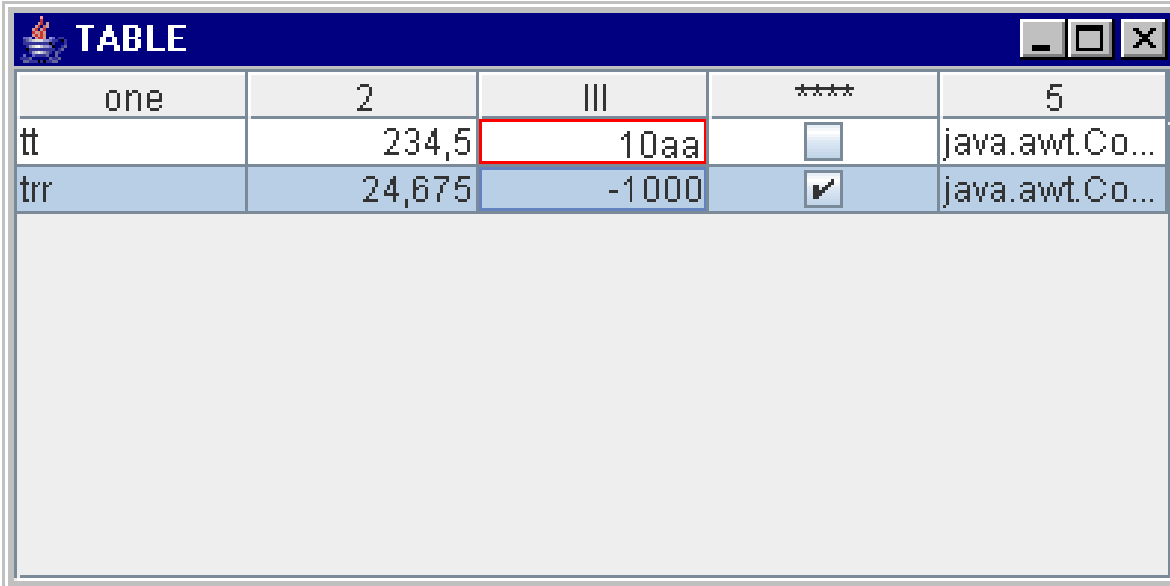

Собственная табличная модель(продолжение)



The image shows a screenshot of a Java Swing window titled "TABLE". The window contains a table with 5 columns and 2 data rows. The columns are labeled "one", "2", "III", "****", and "5". The first row has values "tt", "234,5", "10", an unchecked checkbox, and "java.awt.Co...". The second row has values "trr", "24,675", "-1000", a checked checkbox, and "java.awt.Co...".

one	2	III	****	5
tt	234,5	10	<input type="checkbox"/>	java.awt.Co...
trr	24,675	-1000	<input checked="" type="checkbox"/>	java.awt.Co...

Собственная табличная модель(продолжение)



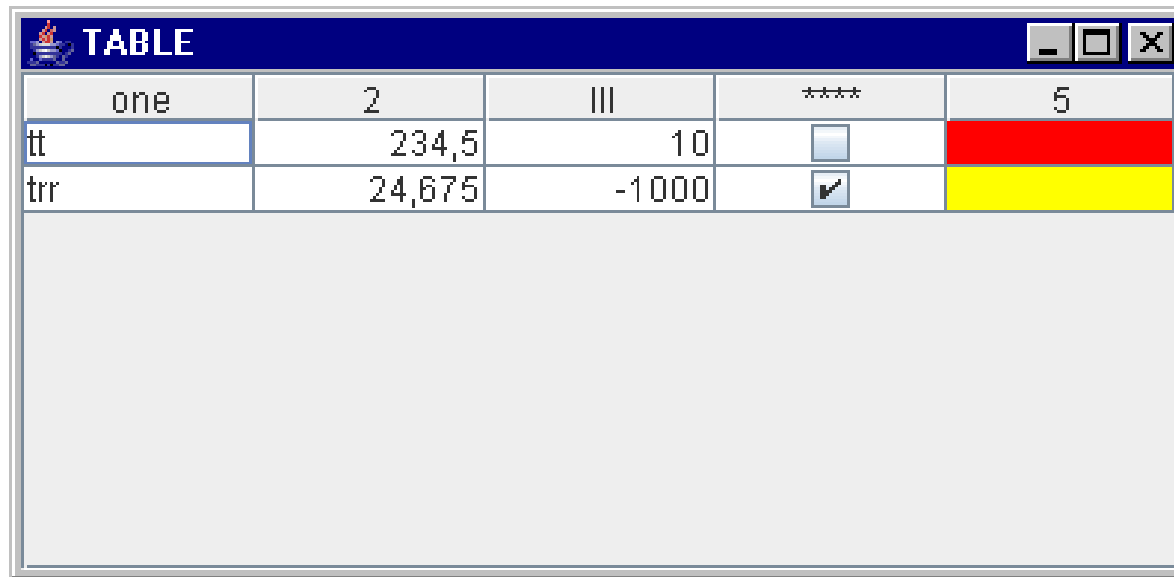
one	2	III	****	5
tt	234,5	10aa	<input type="checkbox"/>	java.awt.Co...
trr	24,675	-1000	<input checked="" type="checkbox"/>	java.awt.Co...

Изменение вида ячеек

```
class ColorRenderer extends DefaultTableCellRenderer {  
    public void setValue(Object value) {  
        setBackground((Color) value);  
    }  
}
```

```
JTable tabl = new JTable (new FirstModel());  
tabl.setDefaultRenderer(Color.class, new ColorRenderer());
```

Изменение вида ячеек



The image shows a window titled "TABLE" with a table containing 5 columns and 2 rows. The first row contains the headers "one", "2", "III", "****", and "5". The second row contains the values "tt", "234,5", "10", an unchecked checkbox, and a red cell. The third row contains the values "trr", "24,675", "-1000", a checked checkbox, and a yellow cell. The table is displayed within a window with standard Windows-style window controls (minimize, maximize, close) in the top right corner.

one	2	III	****	5
tt	234,5	10	<input type="checkbox"/>	
trr	24,675	-1000	<input checked="" type="checkbox"/>	