JTextComponent 1

[Jython JTextComponent 1][Jython Swing][Swing][Jython][Python]

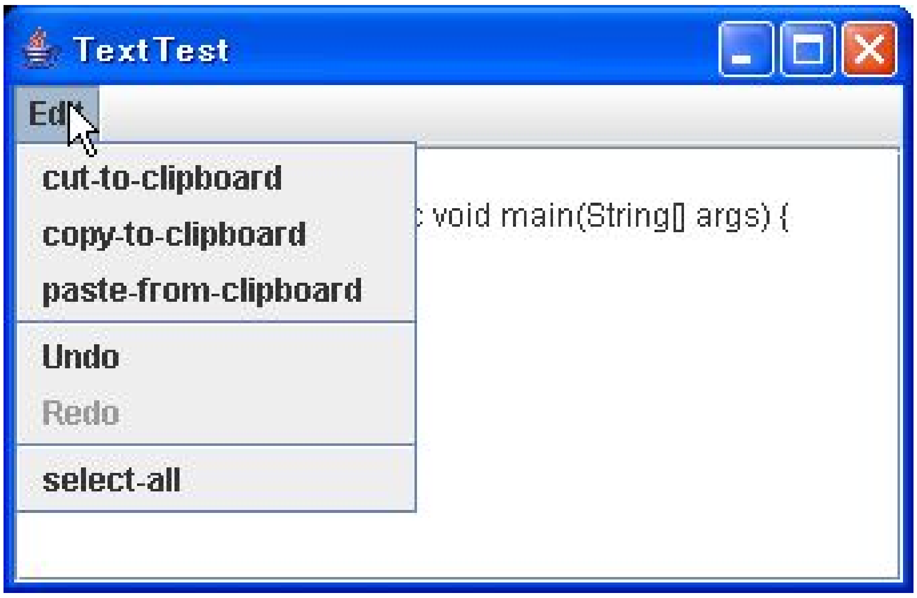
・http://java.sun.com/docs/books/tutorial/uiswing/components/generaltext.html#commands

内容

・メニューとアクションの割付
・キーボード押下との割付
・Undo、Redo の実装
・Document の利用
・Document のイベント感知

ソースコード

実行例



ソースコード

```
import java.awt.BorderLayout;
import java.awt.Event;
import java.awt.event.ActionEvent;
import java.awt.event.KeyEvent;
import java.util.HashMap;
import java.util.Map;

import javax.swing.AbstractAction;
import javax.swing.Action;
import javax.swing.InputMap;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.KeyStroke;
import javax.swing.SwingUtilities;
import javax.swing.event.UndoableEditEvent;
import javax.swing.event.UndoableEditListener;
import javax.swing.text.DefaultEditorKit;
import javax.swing.text.Document;
import javax.swing.text.JTextComponent;
import javax.swing.undo.UndoManager;

/**
 * @see http://java.sun.com/docs/books/tutorial/uiswing/components/generaltext.html#commands
 */
public class JTextTest {
    private JFrame frame;

    private Map<Object, Action> actionMap = new HashMap<Object, Action>();
    private UndoManager undo = new UndoManager();

    private UndoAction undoAction;
    private RedoAction redoAction;

    private JMenu getMenu() {
        JMenu menu = new JMenu("Edit");

        menu.add(actionMap.get(DefaultEditorKit.cutAction));
        menu.add(actionMap.get(DefaultEditorKit.copyAction));
        menu.add(actionMap.get(DefaultEditorKit.pasteAction));
        menu.addSeparator();

        undoAction = new UndoAction();
        menu.add(undoAction);

        redoAction = new RedoAction();
        menu.add(redoAction);

        menu.addSeparator();
        menu.add(actionMap.get(DefaultEditorKit.selectAllAction));

        return menu;
    }
    private void createActionMap(JTextComponent txt) {
        Action[] actions = txt.getActions();
        for(Action action : actions) {
            this.actionMap.put(action.getValue(Action.NAME), action);
        }
    }
    private void createUI() {
        frame = new JFrame("TextTest");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JTextArea txtArea = new JTextArea(10, 30);

        // Document のイベント感知 (Undo Redo 管理を行う )
        Document doc = txtArea.getDocument();
        doc.addUndoableEditListener(
            new UndoableEditListener() {
                public void undoableEditHappened(UndoableEditEvent e) {
                    undo.addEdit(e.getEdit());
                    undoAction.updateState();
                    redoAction.updateState();
                }
            }
        );

        frame.getContentPane().add(new JScrollPane(txtArea), BorderLayout.CENTER);

        // TextComponent が持つ Action を Map に格納
        createActionMap(txtArea);
        // メニューを作成
        JMenu menu = getMenu();
        JMenuBar mb = new JMenuBar();
        mb.add(menu);
        frame.setJMenuBar(mb);

        // テキストの Undo、Redo にキーを割り当てる (Ctrl+z,Ctrl+y)
        InputMap inputMap = txtArea.getInputMap();
        KeyStroke undoKey = KeyStroke.getKeyStroke(KeyEvent.VK_Z,Event.CTRL_MASK);
        KeyStroke redoKey = KeyStroke.getKeyStroke(KeyEvent.VK_Y,Event.CTRL_MASK);
        inputMap.put(undoKey, undoAction);
        inputMap.put(redoKey, redoAction);

        frame.pack();
        frame.setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(
                new Runnable(){
                    public void run() {
                        JTextTest jft = new JTextTest();
                    jft.createUI();
                }
            }
        );
    }

    @SuppressWarnings("serial")
    class UndoAction extends AbstractAction {
        public UndoAction() {
            super("Undo");
            setEnabled(false);
        }
        public void actionPerformed(ActionEvent e) {
            try {
                undo.undo();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
            updateState();
            // Redo を可能に
            redoAction.updateState();
        }
        public void updateState() {
            setEnabled(undo.canUndo());
        }
    }

    @SuppressWarnings("serial")
    class RedoAction extends AbstractAction {
        public RedoAction() {
            super("Redo");
            setEnabled(false);
        }
        public void actionPerformed(ActionEvent e) {
            try {
                undo.redo();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
            updateState();
        }
        public void updateState() {
            setEnabled(undo.canRedo());
        }
    }
}
```