

# JavaFX 8 - RegextextField

## Funktionsweise

Das **RegextextField** überprüft bei Änderungen seines Inhalts über die GUI, ob der Text zu einem Regulären Ausdruck passt. Falls der Text nicht passt, wird der alte Inhalt wiederhergestellt. Änderungen über Aufruf der `setText()`-Methode werden nicht überprüft.

## beispiel.fxml

Da der JavaFX Scene Builder die Klasse **RegextextField** nicht kennt, müssen die Elemente im Texteditor bearbeitet werden.

```
<!-- Package einbinden, in dem sich die Klasse RegextextField befindet -->
<?import regextextfieldtester.*?>

<!-- Eingabefeld für 6-stellige Hexadezimalzahlen -->
<RegextextField regex="[0-9a-fA-F]{0,6}" />

<!-- unbeschränktes Eingabefeld -->
<RegextextField />
```

## RegextextField.java

```
package regextextfieldtester;
import javafx.scene.control.TextField;

public class RegextextField extends TextField {
    //-----
    // Das Attribut "regex" speichert den Regulären Ausdruck,
    // mit dem der Inhalt des TextField überprüft wird
    //-----
    private String regex;

    public RegextextField() {
        this.regex = ".*";
    }

    public String getRegex() {
        return regex;
    }

    public void setRegex(String regex) {
        this.regex = regex;
    }

    //-----
    // Bei Änderung des Inhalts des TextField über GUI wird überprüft
    // ACHTUNG: Bei Aufruf der setText()-Methode wird NICHT überprüft!
    //-----
    @Override
    public void replaceText(int start, int end, String text) {
        // alten Text und Cursorposition merken
        String alterText = this.getText();
        int alteCursorPosition = this.getCaretPosition();
```

```
        // neuen Text zuweisen
        super.replaceText(start, end, text);

        // überprüfen und evtl. zurücksetzen
        ueberpruefen(alterText, alteCursorPosition);
    }

    @Override
    public void replaceSelection(String replacement) {
        // alten Text und Cursorposition merken
        String alterText = this.getText();
        int alteCursorPosition = this.getCaretPosition();

        // neuen Text zuweisen
        super.replaceSelection(replacement);

        // überprüfen und evtl. zurücksetzen
        ueberpruefen(alterText, alteCursorPosition);
    }

    private void ueberpruefen(String alterText, int alteCursorPosition) {
        String neuerText = this.getText();
        if (!neuerText.matches(regex)) {
            // alten Text und alte Cursorpoition wiederherstellen
            this.setText(alterText);
            this.positionCaret(alteCursorPosition);
        }
    }
}
```