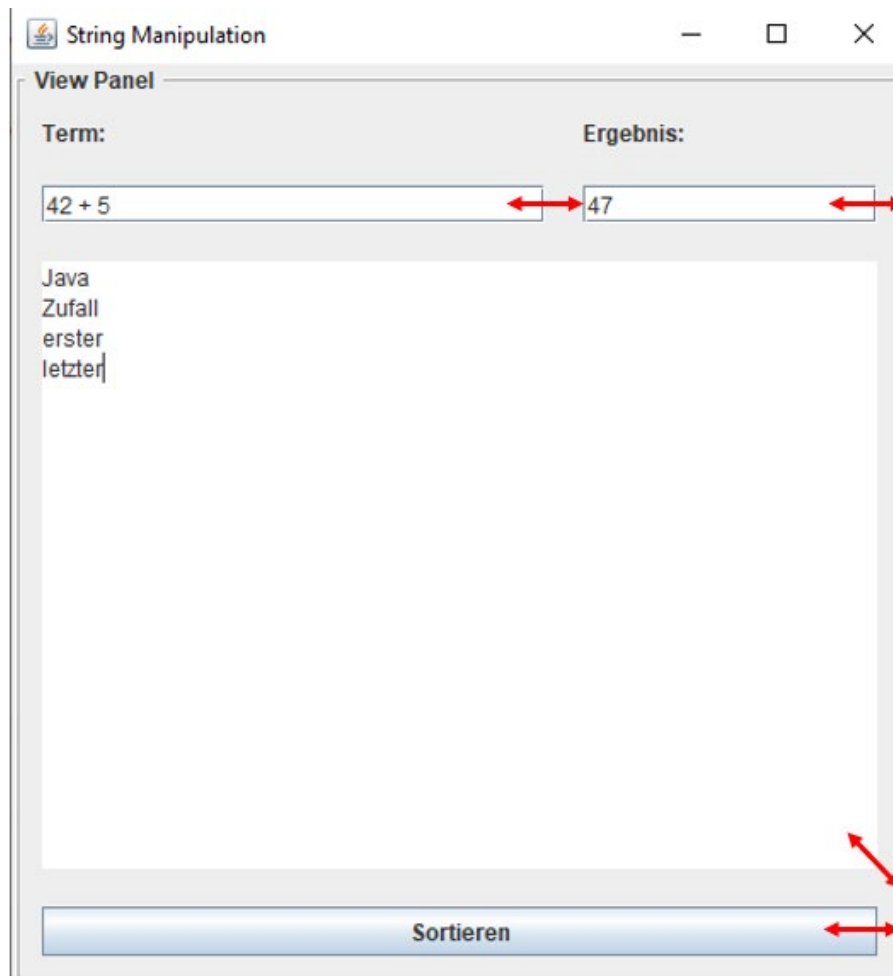


Übung: String Manipulation mit Klassen der Standardbibliothek

Ziel dieser Übung ist es mit Strings zu arbeiten und mit Methoden aus der Java Klassenbibliothek Strings zu parsen und zu zerlegen. Das ist vor allem mit Regularen Ausdrücken ein beliebig komplexes Unterfangen. Wir beschränken uns auf einen einfachen Parser, der einen Rechenausdruck in eine durchführbare Rechenoperation überführt.

Das zugrundeliegende GUI für die Übung soll die folgende Gestalt haben.



Das GUI besteht aus der View mit zwei **TextField** für den Term und das Rechenergebnis. Darunter erscheint ein **TextArea** wo Wörter eingegeben werden können, die beim Betätigen des **Button** in alphabetische Reihenfolge gebracht werden sollen. Das GUI ist im **GridBagLayout** organisiert und das Verhalten bei Größenänderung ist durch die Pfeile in der obigen Skizze beschrieben.

Um den Parser nicht zu komplex werden zu lassen stellen wir klare Anforderung an das Eingabe Format und beschränken die Operationen auf Addition und Subtraktion. Prinzipiell soll zwischen Zahl und Operator immer mindestens ein Leerzeichen sein. Etwaige Fehleingaben müssen nicht abgefangen werden.

Selbstverständlich soll das Programm nach dem MVC-Konzept entworfen werden, wie Sie dem Klassendiagramm entnehmen können.

Aufgabe 1:

Erzeugen Sie als erstes in der Klasse **JavaLibraryFramework** das gewohnte MVC Pattern. Bauen Sie dann das GUI im Konstruktor der Klasse View gemäss der Skizze oben auf. Sorgen Sie als nächstes für die benötigte "MVC-Infrastruktur" damit Ereignisse über den Controller die entsprechenden Model Methoden auslösen (siehe Klassendiagramm und Doku im Code). Der Controller ist in dem Fall nicht besonders intelligent.

Aufgabe 2:

Realisieren Sie in der Methode **berechneErgebnis()**, dass ein einfacher Term aus zwei Operanden addiert oder subtrahiert wird z.B. "42 + 5". Wir gehen davon aus, dass zwischen Operand und Operator immer mindestens ein Leerzeichen folgt.

Über die Observer Schnittstelle soll die **View** upgedated und das Ergebnis der Rechnung dargestellt werden.

Aufgabe 3:

Erweitern Sie die Methode **berechneErgebnis()** dahingehend, dass mehrere Operatoren berücksichtigt werden können z.B. "42 + 5 – 8 + 7..."

Aufgabe 4:

Durch Drücken des Buttons "Sortieren" werden die Textzeilen im **TextArea** alphabetisch sortiert. Realisieren Sie das in der Methode **alphabetSort()** und sorgen Sie für die Darstellung der umsortierten Zeilen in der **View**.

Hinweis: Als einfacher Sortieralgorithmus könnte BubbleSort verwendet werden.

Aufgabe 5 (optional):

Der Parser kann für zusätzliche Operationen und Eingabeflexibilität beliebig komplex gemacht werden

Klassendiagramm:

