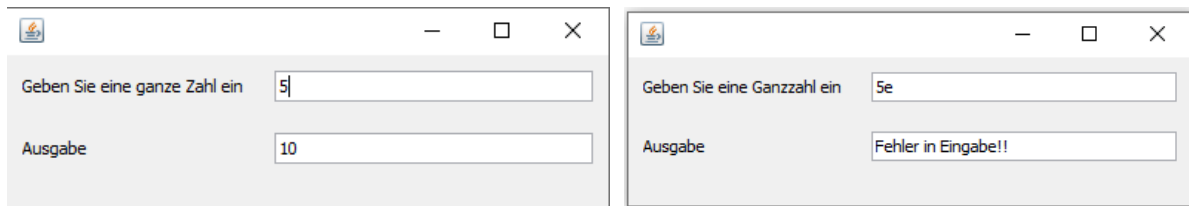


## Übung: Exceptions

Ziel dieser Übung ist es, das Behandeln von Ausnahmen (Exceptions) in Java zu üben und weiter zu vertiefen. Hierzu erstellen wir ein kleines GUI, mit der in einem Textfeld eine Ganzzahl eingelesen werden kann. In einem Ausgabe Textfeld soll der doppelte Werte dargestellt werden. Falls der Benutzer keine Ganzzahl eingibt, würde das Programm zur Laufzeit abstürzen, da die Umwandlung der Zeichenkette fehlschlägt. Dies soll mit dem Konzept des Exception Handling verhindert werden und anstelle eines Absturzes eine Fehlermeldung erscheinen

### Bildschirmausgabe:



### Aufgabe 1

In der Vorlage zur Übung finden Sie ein vorbereitetes File

- Erstellen Sie zunächst im Konstruktor des *JPanels* das GUI im *GridBagLayout*

### Der ActionListener soll folgendermassen erweitert werden:

- Die Eingabe im ersten Textfeld in eine lokale Ganzzahl *zahl* kopieren.
- Den Wert der Variable *zahl* verdoppeln und im zweiten Ausgabefeld darstellen.
- Durch zufügen der *try / catch / finally* Blöcke sollen nun Fehler bei der Zeichenkette-Ganzzahl-Konvertierung aufgefangen werden. Fügen Sie an geeigneten Stellen Konsole-Ausgaben hinzu

### Testen des Programms auf die geforderten Bedingungen

- Testen Sie die Applikation im Normal- und Fehlerfall. Verwenden Sie hierzu auch den Debugger und schauen Sie sich hierbei auch die einzelnen Verzweigungen im Programmablauf genauer an.
- Verwenden Sie den eingegebenen Zahlenwert als Divisor. Erzeugen Sie eine Division durch Null. Wird diese Ausnahme aufgefangen? Falls nicht: Ändern Sie ihren Code so, dass diese Ausnahme aufgefangen wird!

## Aufgabe 2

In dieser Aufgabe wollen wir das Arbeiten mit Exceptions noch etwas verfeinern. Die Fehlerursache soll möglichst genau an den Benutzer weitergegeben werden. Hierzu werden wir den Fehler in einer eigenen Methode erkennen und entsprechend behandeln.

Der Leitgedanke bei der Ausnahmenbehandlung sollte wenn möglich immer sein, Fehler möglichst lokal zu behandeln und keine unnötig komplexe Kontrollstrukturen aufzubauen.

### Ziel:

Das Programm aus Aufgabe 1 soll so erweitert werden, dass die Berechnung des Ausgabewertes in eine Methode ausgelagert wird. Der eingegebene Wert soll weiterhin verdoppelt werden, es soll aber sichergestellt werden, dass keine Werte grösser 100 eingegeben werden.

**Tipp:** Wenn Sie die bestehende Klasse `ExceptionDemo` in ein anderes Package kopieren, können Sie den bisherigen Code beibehalten

Testen Sie Ihr Programm nach jedem Teilschritt

- Schreiben Sie eine Methode *berechnung()*. Die Ganzzahlvariable *zahl* wird als Parameter übergeben. Die Methode soll den Wert verdoppeln und das Resultat zurückgeben.
- Die Berechnung darf nur mit Zahlenwerten < 100 durchgeführt werden. Falls eine grössere Zahl an die Methode *berechnung()* übergeben wird, soll eine *NumberFormatException* ausgelöst werden.
- Da wir jetzt keine Möglichkeit mehr haben, um die genaue Fehlerursache zu eruieren, benötigen wir zusätzliche Informationen: Verwenden Sie Anstelle des Default-Konstruktors der Klasse *NumberFormatException* den Konstruktor, dem als Parameter eine Zeichenkette übergeben werden kann. Im catch-Teil kann mit *getMessage()* der entsprechende Text wieder aus dem Objekt gelesen werden. Mit Hilfe dieser Unterscheidung, können Sie für jede Fehlerursache eine passende Meldung anzeigen.

## Aufgabe 3

Im Letzten Aufgabenabschnitt wollen wir nun noch einen Schritt weiter gehen und eine eigene Ausnahme-Klasse erstellen. Damit können wir Fehlerursachen und entsprechende Informationen nach Belieben unterscheiden.

- Programmieren Sie eine Klasse *ValueToBigException*, welche von der Klasse *NumberFormatException* erbt. Sorgen Sie nun dafür, dass bei einem zu grossen Zahlenwert eine *ValueToBigException* und keine *NumberFormatException* ausgelöst wird.

### Hinweis:

Der Konstruktor der Klasse soll einen Fehlertext und eine ErrorID übergeben bekommen und diese setzen. Der Fehlertext wird mit dem Konstruktor der Superklasse gesetzt. Die Klasse *ValueToBigException* soll dann noch eine *getErrorID()* Methode bekommen, um den ErrorCode auslesen zu können.