Aufgabe 9.1 - Fehlerbehandlung zu Aufgabe 3.8

Erweitern Sie Ihr Ceasar Code Programm (Übung 3.8).

Beim Einlesen von Ceasar-Code soll die Eingabe überprüft werden. Es dürfen nur die Kleinbuchstaben von a bis z eingegeben werden. Ansonten soll eine Exception geworden werden.
Fangen Sie dabei folgende Fehler ab:

1. Eingabe von Großbuchstaben: **IllegalCapitalLetter**-Exception
2. Eingabe von allen anderen Zeichen, außer Kleinbuchstaben: **InvalidCharacter**-Exception

Aufgabe 9.2 - Logging mit Log4j

Schreiben Sie ein einfaches Programm, welches den Log4j-Logger verwendet. (Laden Sie hierzu die Log4j.jar-Datei herunter)

Kopiere Sie folgenden Abschnitt in Ihr Programm und setzen Sie verschiedene Levels.

```
logger.debug( "Meine Debug-Meldung" );
logger.info(  "Meine Info-Meldung"  );
logger.warn(  "Meine Warn-Meldung"  );
logger.error( "Meine Error-Meldung" );
logger.fatal( "Meine Fatal-Meldung" );
```

Aufgabe 9.3 - Logging mit Interface

Aufgabe 9.3.1 - Logging Interface

Erstellen Sie ein package **log**, das ein Interface **LogInterface** enthält. Dieses soll eine Schnittstelle für das Loggen (d. h. Protokollieren) von error, warning und debug (log Level) Nachrichten bereitstellen. Verwenden Sie für die Abbildung des log Levels ein **Enum**.

Das Interface soll die folgenden Methoden enthalten:

- void setLogLevel(LogLevel level)
- void error(String msg)
- void warning(String msg)
- void debug(String msg)

Aufgabe 9.3.2 - ConsoleLogging-Klasse

Fügen Sie nun dem **log** package die Klasse **ConsoleLog** hinzu, welche das **LogInterface** implementiert, indem die Nachrichten auf dem Bildschirm ausgegeben werden. Dabei sollen für das log Level DEBUG alle Nachrichten ausgegeben werden, für das log Level WARNING sollen nur die Nachrichten des log levels warning oder „schlimmer“, also error, angezeigt werden und für das log Level ERROR sollen nur Nachrichten des log Levels error ausgegeben werden. Schreiben Sie ein Programm, die eine Instanz der **ConsoleLog** Klasse erstellt und deren Funktionen testet.

Aufgabe 9.4 - Exception anhand einer Bar

Schreiben Sie ein Java-Programm, welches eine Bar simuliert, bei der nur kalte Getränke serviert werden können und an virtuelle Kunden ausgeliefert wird. Die optimale Temperatur eines kalten Getränkes beträgt 8° Celsius, wobei eine Toleranz von +-2° akzeptiert wird.

Bei der Servierung eines solchen kalten Getränkes können einige Fehler enstehen. Lösen sie diese **Fehlererkennung und die anschließende Fehlerbehandlung** mithilfe von **Exceptions**.

Auf folgende Probleme muss unterschiedlich reagiert werden.

- Das Getränk ist zu kalt
- Das Getränk ist zu warm
- Der Vorrat im Lager ist leer

Schreiben Sie ein läufiges Programm, welches das Verhalten simuliert und auf die entsprechenden Fehlermeldung passend reagiert.

Hier ein Beispiel:

```
Gebe die Temperatur eines kalten Getränkes ein: 12
Neuer Vorrat: 9
Getränk ist zu warm


Gebe die Temperatur eines kalten Getränkes ein: 10
Neuer Vorrat: 8
Das Getränk ist optimal und der Kunde ist zufrieden


Gebe die Temperatur eines kalten Getränkes ein: 9
Neuer Vorrat: 7
Das Getränk ist optimal und der Kunde ist zufrieden


Gebe die Temperatur eines kalten Getränkes ein: 6
Neuer Vorrat: 6
Getränk ist zu kalt
```