Pflichtaufgabe 5 – Exception and Generic

(Teil 1)

Fehler beseitigen (3 Punkte)

Übernehmen Sie den Quellcode der Java-Datei AusnahmeTestOhneBehandlung.java in den Editor. Ändern Sie das
Programm so ab, dass kein Laufzeitfehler mehr auftritt. Es gibt (mind.) zwei Lösungen, bitte finden Sie diese beiden? Schauen Sie sich den Integer-Wrapper an und prüfen Sie welche Methoden Sie noch finden.

```
/**
 * Einfaches Programm, das eine NumberFormatException ausloest beim
 * Konvertieren einer Zeichenkette in eine ganze Zahl.
 *
 * @author MGE

 * @version 1.0, 03/2021

 */
public class AusnahmeTestOhneBehandlung {

public static void main(String[] args) {
        int i;
        int basis = 0;
        for (basis = 10; basis >= 2; --basis) {
            i = Integer.parseInt("40", basis);
            System.out.println("40 zur Basis " + basis + " = " + i);
        }
    }
}
```

Ausnahme abfangen (2 Punkte)

Lösen Sie das Problem in der vorherigen Aufgabe, in dem Sie die Ausnahme abfangen und eine entsprechende Nachricht auf die Konsole, mit Erklärung, warum diese Ausnahme geworfen wird, mitteilen.

Ausnahme Weiterleiten (2 Punkte)

Implentieren Sie eine neue Klasse. Kopieren Sie sich den Rumpf der ursprünglichen main-Methode und fügen Sie diesen in eine neue Methode der neuen Klasse ein. Leiten Sie die Fehlermeldung weiter.

Rufen Sie in der nun leeren main-Methode die neue Methode auf und fangen Sie an dieser Stelle die Ausnahme aus. Geben Sie eine entsprechende Fehlermeldung aus. Debuggen Sie Ihr Programm, indem Sie Breakpoints zu Kontrolle setzen. Überlegen Sie an welcher Stelle der Breakpoint aufgerufen wird und ob zuvor die entsprechende Fehlermeldung auf die Konsole ausgegeben werden.

(Teil 2)

GanzzahlDivisionCatch (4 Punkte)

Implementieren Sie eine Methode, die zwei Konsolen-Eingaben aufnimmt und diese beiden Werte dividiert. Importieren Sie dafür die Klasse Scanner (https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Scanner.html) von java.util.Scanner.

Fangen Sie mind. die InputMismatchException und die ArithmeticException auf, sobald Sie über die Konsole falsche Parameter eingeben. Kommentieren Sie die Fehlermeldung entsprechend. Erweitern Sie die Fehlerbehandlung um eine beliebige Fehlermeldung Ihrer Wahl.

GanzzahlDivisionMultiCatch (2 Punkte)

Schaffen Sie die Fehler mit einem Catch abzufangen, trotzdem Sie weiterhin nach ArithmeticException und InputMismatchException "catchen"?

(Teil 3)

Generische Klasse und Generische Klassenmethoden (3 Punkte)

Schreiben Sie bitte eine generische Klasse GenericNumberHelper. Der generische Klassentyp erweitert Number.
Die Klasse hat zwei generische Parameter alpha und beta.
Ergänzen Sie einen Konstruktor, der die zwei Parameter bei der Objektinitiierung der Klasse annimmt.

(2 Punkte)

Ergänzen Sie weiterhin eine Methode division und compare und geben Sie bei division den entsprechenden generischen Typ zurück. Der Rückgabewert bei compare ist frei wählbar.

(2 Punkte)

Testen Sie die Klasse, indem Sie 1) zwei int-Werte, 2) zwei double-Werte und 3) zwei float-Werte bei der Objektinitiierung übergeben. Beachten Sie den richtigen Wrapper (Wrapper-Klasse) für die primitiven Datentypen int, float und double für die generische Klasse zu wählen.

Bitte geben Sie mit @author in allen Klassen einen Autor, bzw. beide Autoren an.
Implementieren Sie die Klassen möglichst effizient, d.h. so, dass Sie unter Berücksichtigung der Vererbung unnötige Wiederholungen im Code vermeiden. Kommentieren Sie das Programm ausreichend.

Laden Sie alle Klassen in einem ZIP-File hoch. Das ZIP-File benennen Sie bitte mindestens mit Ihrem/Ihren Nachnamen! (sonst Punktabzug)

(Gesamt 20 Punkte)