

Vorlesung „Programmieren“

Wintersemester 2007/2008

10. Übungsblatt

1. Ausnahmebehandlung (Programmieraufgabe) (12 Punkte)

Folgender Java-Quelltext sei gegeben:

```
import java.util.*;
import java.io.*;

public class ExceptionTest {
    public static void main(String [] parm) {
        Scanner s = new Scanner(System.in);
        int a=0, b=0, c=0;
        try {
            // Einlesen von 2 ganzzahligen Werten
            System.out.print("Geben Sie zwei ganzzahlige Zahlen ein: ");
            a = Integer.parseInt(s.next());
            b = Integer.parseInt(s.next());
            if (a==0 & b==0)
                throw new Exception ("Ich streike bei Eingabe von nur Nullen!");
            c = a/b;
            System.out.println(a+"/"+b+" = "+c);
        }
        catch(NoSuchElementException e) {
            System.out.println("\nDateifehler: "+e);
        }
        catch(NumberFormatException n) {
            System.out.println("\nZahlformatfehler: "+n);
        }
        catch(Exception ex) {
            System.out.println("\nSonstiger Fehler: "+ex);
        }
        catch(ArithmeticException ae) {
            System.out.println("\nRechenfehler: "+ae);
        }
    }
}
```

Aufgaben:

a) Die obige Klasse enthält einen Fehler in der Ausnahmebehandlung. Wieso entsteht dieser Fehler und wie kann man diesen Fehler korrigieren? Bitte beschreiben Sie den Fehler und geben Sie die korrigierte Version des obigen Java-Quelltextes ab. (1 Punkt)

b) Welche Art von Ausnahmen werden in der Ausnahmebehandlung für `Exception`-Ausnahmen behandelt werden, obwohl sie in einer speziellen Ausnahmebehandlung gesondert behandelt werden könnten? Wieso erzeugt der Übersetzer beim Kompilieren keine Fehlermeldung für diese Ausnahme, auch wenn diese Ausnahme in keiner Ausnahmebehandlung behandelt würde, wie es zum Beispiel in dem folgenden Java-Quelltext geschieht? (1 Punkt)

```
import java.util.*;
import java.io.*;

public class ExceptionTest {
    public static void main(String [] parm) {
        Scanner s = new Scanner(System.in);
```

```

int a=0, b=0, c=0;

try {
    // Einlesen der Inhalte von a und b
    System.out.print("Geben Sie zwei int-Zahlen ein: ");
    a = Integer.parseInt(s.next());
    b = Integer.parseInt(s.next());
    if (a==0 & b==0)
        throw new ArithmeticException
            ("Nur mit Nullen rechne ich nicht!");
    c = a/b;
    System.out.println(a+"/"+b+" = "+c);
}
catch(NoSuchElementException e) {
    System.out.println("\nDateifehler: "+e);
}
catch(NumberFormatException n) {
    System.out.println("\nFormatfehler: "+n);
}
catch(ArithmeticException ae) {
    System.out.println("\nRechenfehler: "+ae);
}
}
}

```

- c) Der in Aufgabe a) angegebene Java-Quelltext soll dahingehend verändert werden, dass
- anstelle von Exception ("Ich streike bei Eingabe von nur Nullen!") eine spezielle (eigens zu erstellende) geprüfte Ausnahme OnlyNullsException für diese Art von Ausnahmen verwendet wird. (2 Punkte)
 - drei Methoden (anstatt der Methode main(String [] parm)) verwendet werden:
 - Die main(String [] parm)-Methode soll das Endergebnis, welches das Resultat einer Division zweier einzugebender Zahlen ist, ausgeben oder "\nDas Endergebnis konnte nicht berechnet werden!", wenn das Endergebnis auf Grund eines Fehlers nicht berechnet werden konnte. Die main-Methode soll dabei die int compute()-Methode verwenden, um das Endergebnis zu berechnen. (2 Punkte)
 - Die int compute()-Methode soll das Endergebnis berechnen, in dem es die int read()-Methode zum Einlesen von den zwei ganzzahligen, einzugebenden Zahlen verwendet. (3 Punkte)
 - Die int read()-Methode soll eine ganzzahlige Zahl einlesen und zurückgeben, in dem es die Scanner-Klasse analog zu dem gegebenen Quelltext verwendet. (3 Punkte)

Hierbei sollen die Ausnahmen in derselben Methode abgefangen werden, in der sie auf Grund von Java-API-Methoden oder Operationen auftreten können, und entsprechende Fehlerhinweise analog zu dem gegebenen Java-Quelltext ausgegeben werden. Die Ausnahmebehandlung soll *sinnvoll* eingesetzt werden.

Bemerkungen:

- Dieses ist eine Programmieraufgabe. Bitte geben Sie die Lösung *sowohl* auf **Papier** *als auch* den Java-Quelltext in elektronischer Form per **Email** an Ihren Tutor ab.

2. Java-Quelltexte auf Kommandozeile kompilieren (2 Punkte)

Die Eingabe von javac oder java ohne Kommandozeilenparameter in der Kommandozeile gibt eine Kurzübersicht über mögliche Kommandozeilenparameter.

Nehmen Sie an, Sie befinden sich in einem Verzeichnis, zu dem in dem relativen Verzeichnis src/myClasses zu übersetzende Java-Quelldateien liegen, in dem relativen Verzeichnis lib/thirdPartyLibraries verwendete, übersetzte Klassen und in dem relativen Verzeichnis output/myClasses die Ausgabe der übersetzten .class-Dateien erfolgen soll.

Aufgaben:

- a) Wie lautet der Befehl in der Kommandozeile zum Kompilieren der entsprechenden Java-Quelldateien (ohne das aktuelle Verzeichnis zu wechseln). (1 Punkt)

b) Wie lautet der Befehl in der Kommandozeile zum Ausführen einer übersetzten Klasse `MyClass` (ohne das aktuelle Verzeichnis zu wechseln). (1 Punkt)

Bemerkungen:

- Jede Seite soll oben rechts den Namen der Abgebenden und die Übungsgruppennummer (wichtig!) enthalten.
- Lösungen für die Übungsaufgaben sind (in der Regel) zu zweit abzugeben.
- Kommentieren Sie Ihre Lösungen! Besteht eine Lösung aus mehreren Zetteln, so sind diese zusammen zu heften. Bitte keine Hüllen, Mappen, o.ä..
- Bitte schicken Sie *Programmieraufgaben zusätzlich zur Abgabe auf Papier in elektronischer Form per Email* an Ihren jeweiligen Tutor.
- Kommentieren Sie ihren Quelltext bei Programmieraufgaben. Dabei sollen keine Trivialitäten kommentiert werden, also bitte keine Kommentare wie

~~x=5; // Wir weisen nun der Variablen x den Wert 5 zu~~

sondern sinnvolle Kommentare, die Ideen des Quelltextabschnittes beschreiben oder auf Unteraufgaben (z. B. a), b), ...) hinweisen.

- **Hinreichende Bedingung für die Zulassung zur Klausur:** 50% der erreichbaren Punkte bei jedem Übungszettel (bis auf zwei) und einmaliges Vorrechnen in der Übung
- **Zertifikatskriterium:** Das Bestehen der Klausur am Ende des Semesters

Abgabetermin: Freitag, 25.1.2008, nach der Vorlesung