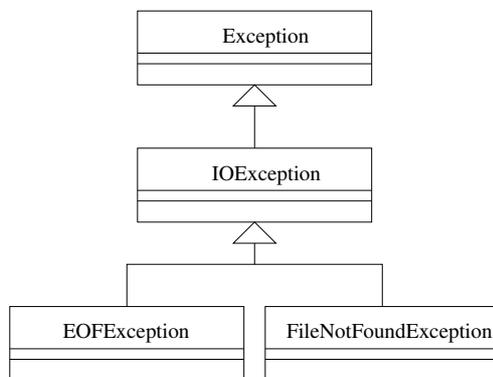


Übungen zu Einführung in die Informatik I

Aufgabe 28 **Ausnahmen**

Gegeben sei folgender Ausschnitt aus der Klassenhierarchie von Ausnahmen in Java und folgende Java-Implementierung der Klasse `ExceptionTest`:



```
import java.io.*;
public class ExceptionTest {
    public static void main (String[] args) {
        try {
            ... // Inneres des try-Blockes
        }
        catch (EOFException e) {
            System.out.println("EOFException");
        }
        catch (IOException e) {
            System.out.println("IOException");
        }
        catch (Exception e) {
            System.out.println("Exception");
        }
        System.out.println("ENDE");
    }
}
```

- a) An der durch „...“ gekennzeichneten Stelle im `try`-Block stehe ein Programmstück, durch das Ausnahmen vom Typ `IOException`, `EOFException` oder `FileNotFoundException` ausgelöst werden können.

Was wird bei Ausführung der `main`-Methode ausgedruckt, falls dabei im `try`-Block

- i) als erstes eine Ausnahme vom Typ `EOFException` ausgelöst wird,
 - ii) als erstes eine Ausnahme vom Typ `FileNotFoundException` ausgelöst wird, oder
 - iii) gar keine Ausnahme ausgelöst wird?
- b) Was wird bei Ausführung der `main`-Methode ausgedruckt, falls dabei im `try`-Block als erste Ausnahme eine Division durch 0 auftritt?

Aufgabe 29 **Fehlerbehandlung**

Schreiben Sie eine Java-Programm, das eine Textdatei einliest und die darin enthaltenen Zahlen summiert. Der Dateiname soll vom Benutzer eingegeben werden können. Welche Fehler können auftreten? Diese Fehler sollen an entsprechender Stelle per `try ... catch` abgefangen und behandelt werden.

Aufgabe 30 Hash-Tabellen

- a) Entwickeln Sie eine generische Klasse `HashTable<K, V>`. Ein Objekt der Klasse `HashTable` soll eine Hash-Tabelle sein, die Objekte vom Typ `K` (*Keys*) auf Objekte vom Typ `V` (*Values*) abbildet. Nehmen Sie dabei an, dass Objekte vom Typ `K` geeignete Methoden `int hashCode()` und `boolean equals(Object obj)` haben.

Die Klasse `HashTable` soll folgende Methoden zur Verfügung stellen:

- Ein Konstruktor `HashTable(int size)`, der eine Tabelle mit `size` leeren Einträgen erzeugt.
 - `void insert(K key, V value)`, die die Abbildung von `key` auf `value` in die Tabelle einträgt.
 - `V lookup(K key)`, die den Wert zurückliefert, der `key` zugeordnet ist, oder `null` wenn es keinen Eintrag für `key` in der Tabelle gibt.
- b) Entwickeln Sie eine Klasse zur Berechnung von Binomialkoeffizienten. Zur Vermeidung von Mehrfachberechnungen der Binomialkoeffizienten, soll die Klasse eine Hash-Tabelle einsetzen. Bevor ein Wert $\binom{n}{k}$ berechnet wird, soll in der Tabelle nachgeschlagen werden, ob ein Eintrag für (n, k) enthalten ist. Anderenfalls soll er rekursiv berechnet und eingetragen werden.

(Hinweis: Die Hash-Tabelle soll Paare (n, k) von Integer-Objekten auf den Binomialkoeffizient $\binom{n}{k}$ als Long-Objekt abbilden. Die `Pair`-Klasse aus Aufgabe 26 soll um geeignete Methoden `hashCode` und `equals` erweitert werden.)