



## Aufgabenblatt 5

### Aufgabe 1 (Exceptions und Vererbung, 10 Punkte)

Es sei der folgende Quelltext gegeben:

```
import java.io.*;
public class Currywurst extends Fastfood {
    public static void main(String[] args) throws IOException {
        new Currywurst().eat();
    }
    // insert code here
}
class Fastfood {
    void eat() throws IOException { }
}
```

Für welche der folgenden Methodendefinitionen, jeweils eingefügt für die Zeile mit dem Kommentar “insert code here”, lässt sich der Quelltext kompilieren?

- (a) `void eat() { }`
- (b) `void eat() throws Exception { }`
- (c) `void eat(int x) throws Exception { }`
- (d) `void eat() throws FileNotFoundException { }`
- (e) `void eat() throws RuntimeException { }`

Erläutern Sie jeweils in einem Satz, warum eine Kompilierung möglich bzw. nicht möglich ist.

### Aufgabe 2 (Generische Bibliotheksklasse nutzen, 10 Punkte)

Informieren Sie sich mit Hilfe der Java-API-Dokumentation über die generische Klasse `ArrayList<E>` (aus dem Paket `java.util`). Diese Klasse dient zur Verwaltung einer Liste von Objekten des Typs `E` (Komponententyp). Intern wird die Liste mit Hilfe eines Feldes repräsentiert, wobei automatisch ein größeres Feld erzeugt wird, wenn die Kapazität des aktuellen Feldes erschöpft ist.

Wichtige Methoden, die Sie in dieser Aufgabe benötigen, sind:

- `void add(E e)`

Fügt ein Objekt `e` vom Typ `E` am Ende der Liste ein.

- `int size()`  
Liefert die Anzahl der Objekte, die sich in der Liste befinden.
- `E get(int i)`  
Liefert das  $i$ -te Element der Liste.

Schreiben Sie ein Java-Programm, das die Zeilen einer Textdatei (oder auch der Standardeingabe) in umgekehrter Reihenfolge ausgibt.

**Beispiel:** Wenn die Eingabedatei die Zeilen

```
in der Mensa
gibt es heute
Schweinebraten mit Knoedel und Rotkohl
```

enthält, dann sollte ihr Programm die Ausgabe

```
Schweinebraten mit Knoedel und Rotkohl
gibt es heute
in der Mensa
```

erzeugen.

**Hinweise:**

- Die Klasse `BufferedReader` aus dem Paket `java.io` stellt Ihnen eine Methode `readLine()` zur Verfügung, mit der Sie eine Datei zeilenweise lesen können. Nutzen Sie z.B. die folgenden Anweisungen, um einen `BufferedReader` für die Datei `meineDatei.txt` zu erzeugen:

```
import java.io.*;
...
BufferedReader buf = new BufferedReader(new FileReader("meineDatei.txt"));
```

Anschließend können Sie mit Hilfe von `readLine()` die Datei zeilenweise lesen. Achten Sie auch auf Exceptions und schließen Sie den `BufferedReader` mit der Methode `close()`, wenn Sie alle Zeilen gelesen haben.

- Nutzen Sie eine `ArrayList` zur Zwischenspeicherung der Zeilen. Vergessen Sie hierzu nicht die `import`-Anweisung.

### Aufgabe 3 (Generische Klasse implementieren, 10 Punkte)

Unter einem *Paar* wollen wir genau zwei Objekte des gleichen Typs  $T$  verstehen, die zusammengehören.

- (a) Implementieren Sie eine generische Klasse `Paar<T>` zur Repräsentation von Paaren mit beliebigem Komponententyp  $T$ .

Die beiden Objekte, die ein Paar bilden, sollen dem Paar-Konstruktor als Parameter übergeben werden. Darüberhinaus sollte die generische Klasse `Paar<T>` folgende Methoden aufweisen:

- getObject1() und getObject2() um auf die beiden Objekte eines Paares zugreifen zu können,
- swap() um die beiden Objekte innerhalb eines Paares zu vertauschen und
- toString() um für ein Paar eine String-Darstellung zu erhalten.

(b) Implementieren Sie in einer separaten Klasse eine *generische* Klassenmethode (vgl. Folie 130), die für eine `ArrayList` von Paaren als Parameter alle in der Liste enthaltenen Paare textuell ausgibt.

(c) Schreiben Sie ein kleines Testprogramm, mit dem Sie

- zwei `ArrayList`en für Paare mit unterschiedlichen Komponententypen erzeugen,
- diese Listen mit Paaren füllen und
- für jede der beiden Listen Ihre Methode aus (b) aufrufen.

Hilfe zur Lösung der Aufgaben erhalten Sie in den Übungen der KWen 18 und 19. Abgabe der Lösungen bis **Sonntag, 12. Mai 2013, 24 Uhr** via E-Mail an die Adresse `dat-alg13@mail.inf.h-brs.de`.