

## Übung 1

### 1. Encodings (5 Punkte)

Java-I/O-Klassen unterstützen Encodings. Schreibe eine Java-Klasse Transcoder mit einer public-Methode, die eine Textdatei1 mit einem spezifizierten Encoding1 in eine neue Textdatei2 mit einem ebenfalls spezifizierten Encoding2 transkodiert. Die Namen der Encodings (Charsets im JDK-Jargon) sollen als Strings übergeben werden. Verwende gepufferte Lese- und Schreib-Operationen und achte darauf, dass die Dateien beliebig groß sein dürfen, d.h. nicht erst vollständig in den Arbeitsspeicher gelesen werden sollen. In der Vorlesung wurde skizziert, mit welchen Klassen Leseoperationen durchgeführt werden (jedoch sollen die Dateien *zeichenweise* gelesen und geschrieben werden). Suche für die Implementierung in der JDK-API-Dokumentation bzw. der angegebenen [Literatur](#) nach entsprechenden Klassen auch für die Schreiboperationen (package java.io). Testdateien: [utf8.txt](#) (UTF-8-kodiert) und [latin1.txt](#) (ISO-8859-1) auf der Kurs-Homepage.

### 2. Unicode (10 Punkte)

Implementiere eine Klasse EncodingBytesPrinter, welche für eine gegebene Textdatei (Testdatei: [deja5euros.txt](#), utf-8-kodiert) für jede Eingabetextzeile die Ausgabe wie auf [Folie 18](#) („Déjà 5€?“) erzeugt. Es sollen also für die (im Sun JDK vordefinierten) Ausgabeencodings ISO-8859-15, windows-1252, IBM500 (EBCDIC), UTF-8, UTF-16BE, UTF-16LE sowie UTF-32 für jede Eingabezeile jeweils 2 Ausgabezeilen gedruckt werden: zuerst die einzelnen (Unicode-) Zeichen als Java chars bzw. Strings (ersetze dabei Steuerzeichen durch ein X, eine Methode zum Testen auf Steuerzeichen existiert in java.lang.Character), gefolgt vom Encodingnamen, und in der Zeile darunter die hexadezimalen Byterepräsentation im jeweiligen Ausgabeencoding mit dem vertikalen Strich bündig ausgerichtet. Schreibe dazu zwei Methoden: eine statische main-Methode, welche eine Datei mit einem spezifizierten Encoding (Encodingname und Dateiname als Kommandozeilenargumente in args[] übergeben) einliest und für jede Eingabezeile und für jedes der o.a. Encodings die zweite Methode, void printByteRepresentation(String zeile, String encoding), aufruft, die jeweils die beiden Ausgabezeilen ausgibt. Statische Hilfsmethoden für hexadezimal-Umwandlung gibt es in der Klasse java.lang.Integer; noch einfacher geht es mit der Methode String.format("%02X") – bitte in der Java API-Dokumentation zur Klasse String nachlesen!

Beispielausgabe für java EncodingBytesPrinter utf-8 deja5euros.txt (1. Eingabezeile):

```
D| é| j| à| | 5| €| ?| (ISO-8859-15)
44|E9|6A|E0|20|35|A4|3F|
D| é| j| à| | 5| €| ?| (windows-1252)
44|E9|6A|E0|20|35|80|3F|
D| é| j| à| | 5| €| ?| (IBM500)
C4|51|91|44|40|F5|3F|6F|
D|   | é| j|   | à|   | | 5|   |   | €| ?| (UTF-8)
44|C3|A9|6A|C3|A0|20|35|E2|82|AC|3F|
  D|   | é|   | j|   | à|   | | 5|   |   | €|   | ?| (UTF-16BE)
00 44|00 E9|00 6A|00 E0|00 20|00 35|20 AC|00 3F|
  D|   | é|   | j|   | à|   | | 5|   |   | €|   | ?| (UTF-16LE)
44 00|E9 00|6A 00|E0 00|20 00|35 00|AC 20|3F 00|
    D|   |   |   |   | é|   |   |   | j|   |   |   |   | à|   |   | ... (UTF-32)
00 00 00 44|00 00 00 E9|00 00 00 6A|00 00 00 E0|00 00 00 20| ...
```

- 2 -

### Hinweise zu den Übungen:

- Java-Code bitte sorgfältig kommentieren!
- → [Coding Standards](#) beachten!
- Eclipse-Einrichtung und Shortcuts: → [Folien](#)
- Bitte pro Aufgabe ein eigenes Package anlegen.
- Bitte die angegebenen Packagenamen benutzen und auf die Schreibweise achten: `javaKurs2.uebung<n>.aufgabe<m>`
- Java ist case-sensitive, d.h. in Bezeichnernamen für Variablen, Klassen, Methoden usw. wird zwischen Groß- und Kleinschreibung unterschieden!
- Bitte nur die .java Dateien schicken und die .class Dateien weglassen.
- Bitte das Archiv mit den Lösungen idealerweise so packen, dass das oberste Verzeichnis `uebung<n>` ist. Das macht Jörg das Leben einfacher, wenn er auspackt.
- Für jede Aufgabe bitte eine eigene Testklasse schreiben, die Jörg sofort laufen lassen kann und die zeigt, dass euer Code eine sinnvolle Ausgabe produziert.
- Das mindeste, was wir von einer Lösung erwarten, ist, dass sie kompiliert. Wenn ihr das nicht hinkriegt, bitte die fehlerhaften Stellen auskommentieren.
- Bei Fragen, Unklarheiten nicht genieren, sondern die → [Mailing-Liste](#) verwenden um uns zu kontaktieren und auch die übrigen Kursteilnehmer an einer möglichen Diskussion teilhaben zu lassen.