Christian Bunse & Jöran Pieper
E-Mail: Vorname.Nachname@hochschule-stralsund.de
Büro: Haus 4/306b&331

Hochschule Stralsund

Aufgaben zur Lehrveranstaltung
Laborpraktikum Software
SMSB, SMIB

B. Aufgabenblatt

Die Präsenzaufgaben dieses Aufgabenblattes werden in den Laboren ab dem 21.10.2019 bearbeitet. Die Hausaufgaben dieses Aufgabenblattes sind bis **8:00 Uhr des Abgabetages (siehe ILIAS)** abzugeben. Verspätete Abgaben werden nicht berücksichtigt.

B. Aufgabenblatt

B.1 Hausaufgabe (Enum)

[0,5 Punkte]

Nennen Sie drei Anwendungsbeispiele für Enums und implementieren Sie diese jeweils mittels eines Anwendungsbeispiels. Die Definition des reinen Datentypen oder das Auswechseln von Bezeichnern ist nicht ausreichend. Hier ein Beispiel für einen Enum mit Anwendung:

```
public class iCrunchedEnumExample {

    public enum Company {
        EBAY(30), PAYPAL(10), GOOGLE(15), YAHOO(20), ATT(25);
        private int value;

        private Company(int value) {
            this.value = value;
        }
    }

    public static void main(String[] args) {
        for (Company cName : Company.values()) {
            System.out.println("Company Value: " + cName.value + " - Comapny Name: " + cName);
        }
    }
```

B.2 Hausaufgabe (Wert- und Referenztypen, Un-/Boxing)

[0,5 Punkte]

Beschäftigen Sie sich mit den Unterschieden zwischen Wert- und Verweistypen. Beschreiben Sie, worin der Unterschied zwischen Boxing und Unboxing besteht und demonstrieren Sie diesen anhand eines Programmier-Beispiels.

B.3 Hausaufgabe (Lesen und Ausgeben einer Textdatei mit Streams)

[1 Punkt]

Schreiben Sie ein Programm, das eine Textdatei Byte für Byte einliest und auf der `Console` ausgibt. In den Vorlesungsfolien ist beschrieben wie das geht. Der Benutzer soll den Dateinamen eingeben können. Das Programm soll prüfen, ob die Datei existiert (suchen Sie sich hierzu in der Klasse `File` eine geeignete Methode heraus), und ggf. eine Fehlermeldung ausgeben. Die Ausgabe soll nicht in Form von Bytes erfolgen, sondern in Gestalt von Zeichen. Dazu verwenden Sie die Methoden:

```
public static int decodeChar (char c) {
	return ((int) c);
}

public static char encodeChar (int i) {
	return ((char) i);
}
```

B.4 Hausaufgabe (Exception)

[1 Punkt]

Kopieren Sie Ihr Programm von Aufgabe B.3 und ändern Sie es wie folgt ab. Das Programm soll nun nicht explizit prüfen, ob die Datei existiert, sondern stattdessen eine `Exception` abfangen, die beim öffnen erzeugt wird, wenn die gewünschte Datei nicht existiert. Geben Sie die in der `Exception` enthaltene Nachricht auf der `Console` aus und beenden Sie in diesem Falle das Programm.

B.5 Präsenzaufgabe (Debugging und Schreiben einer Textdatei)

Debuggen Sie das folgende Programm. Setzen Sie einen Breakpoint auf die markierte Zeile.

```
package ArbeitsblattB;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class B5 {

	public static void main(String[] args) {
		try {
			BufferedWriter out = new BufferedWriter(new FileWriter("test.txt"));
			out.write("Lorem ipsum dolor sit amet");
			out.flush();
			out.close();
		} catch (IOException e) {
```

```
                e.printStackTrace();
            }
        }
}
```

Beobachten Sie, dass die erzeugte Datei beim Erreichen des Breakpoints lediglich 0 Bytes enthält. Gehen Sie nun einen Schritt im Debugger weiter. Beobachten Sie anschließend, dass die Datei nun 26 Bytes groß ist. Erläutern Sie die Ursachen Ihre Beobachtung.

B.6 Präsenzaufgabe (Encodings)

Sie benötigen dieses Programm für die folgenden Hausaufgaben! Bitte sichern Sie Ihr Programm auf einem USB-Stick o.ä.

Schreiben Sie ein Programm, das die Zeichenfolge „Die Welt kostet 17 €“ mit einem anschließenden Zeilenumbruch in UTF-8-Kodierung in eine Datei schreibt. Lesen Sie die Datei anschließend byteweise ein und geben Sie die **Byte-Folge** auf dem Bildschirm aus. Geben Sie die Bytes in Dezimalschreibweise aus.

Wenn Sie alles richtig gemacht haben, gibt Ihr Programm folgendes aus:
68 105 101 32 87 101 108 116 32 107 111 115 116 101 116 32 49 55 32 226 130 172 13 10

Vergleichen Sie die Byte-Folge bitte ***genau*** mit der Ausgabe Ihres Programms.

B.7 Hausaufgabe (Encodings)

[1 Punkt]

Betrachten Sie die Ausgabe Ihres Programms aus Aufgabe B.6. Erläutern Sie das Ergebnis. Wieso sind gerade diese Bytes heraus gekommen? Notieren Sie bitte detailliert in einer Tabelle der folgenden Form Ihre Antwort:

Byte	68	105	101	32	87	…		
Bedeutung	…	…	…	…	…	…		

Unterstützung erhalten Sie hier:

- Java-API: http://docs.oracle.com/javase/7/docs/api/ `OutputStreamWriter`- bzw. `PrintWriter`-Konstruktor mit Encoding-Parameter
- http://de.wikipedia.org/wiki/UTF-8 (insb. bzgl. des Euro-Zeichens)
- http://en.wikipedia.org/wiki/Byte_Order_Mark

Sollten Sie Informationen über die Hexadezimal-Schreibweise benötigen, können Sie diese z. B. hier bekommen: http://de.wikipedia.org/wiki/Hexadezimal.

B.8 Hausaufgabe (Encodings)

[1 Punkt]

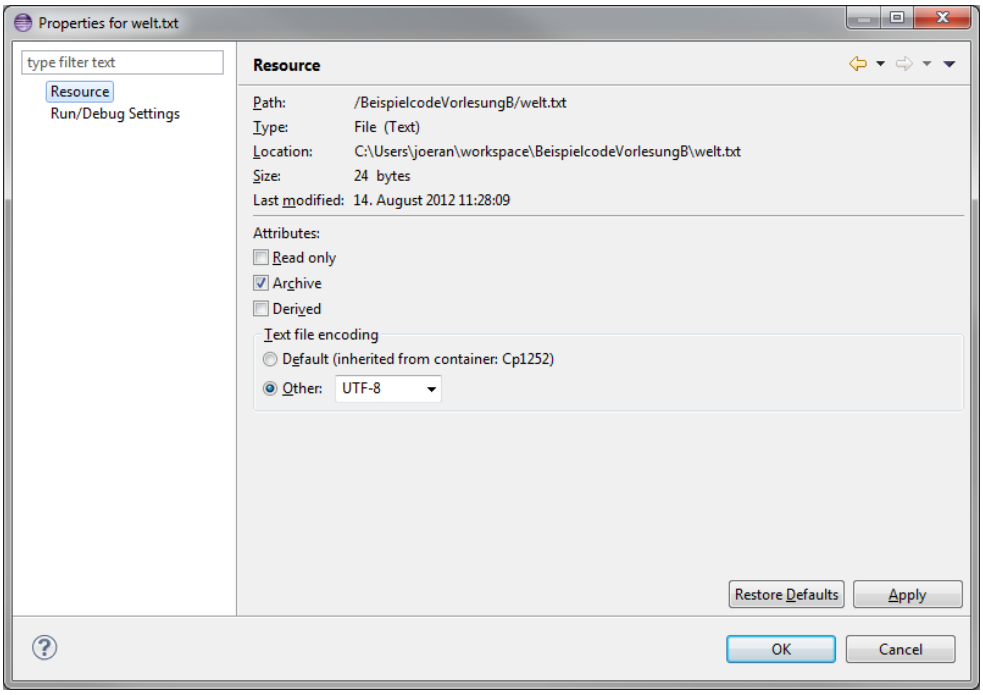
Verwenden Sie IntelliJ oder ECLIPSE, um eine neue Textdatei anzulegen. Gehen Sie dazu wie folgt vor: Rechte Maustaste im Projektmappen-Explorer auf dem Projekt → New → File → Namen vergeben → [Finish]

IntelliJ:

Settings/Preferences mittels (Ctrl+Alt+S) öffnen -> Klick auf „File Encodings“ im Bereich „Editor“ -> Encoding für Datei auf UTF-8 einstellen -> [Apply] -> [OK]

Eclipse:
Rechtsklick auf die Datei im Projektexplorer → Properties → unter „Resource“ → Attributes: → „Text file encoding“ auf „Other: UTF-8“ setzen → [Apply] →[OK]

Schreiben Sie in die Textdatei den Text „Die Welt kostet 17 €“ (plus Zeilenumbruch). Speichern Sie anschließend die Textdatei.



Hinweis: Die Speicheroptionen werden für jede Datei separat konfiguriert, das heißt andere, evtl. geöffnete Quellcodedatei wird weiterhin in der konfigurierten Default-Encodierung gespeichert.

Lesen Sie die so erzeugte Datei mit Ihrem Programm aus Aufgabe B.6 ein und geben den Inhalt byteweise aus. Vergleichen Sie die Ausgabe mit der Ausgabe von Aufgabe B.6. Beschreiben und erläutern Sie das Ergebnis des Vergleichs.

Führen Sie denselben Vorgang ein weiteres Mal durch, speichern Sie eine neue Textdatei nun jedoch in der Codierung „Other: UTF-16“. Vergleichen Sie die Ausgabe erneut mit der Ausgabe von Aufgabe B.6. Beschreiben und erläutern Sie das Ergebnis des Vergleichs.

B.9 Hausaufgabe (Debugging, String, Parsing, i18n + l10n, lokalisierte Formate)

[3 Punkte]

Ihre Aufgabe besteht darin, aus einer gegebenen Datei von Aktienfonds-Daten den Jahresanfangskurs (erster Tageskurs) des Jahres 2011 und den Jahresendkurs (erster Tageskurs) des Jahres 2011 herauszufiltern. Die beiden Kurse sollen verglichen werden. Wenn der Kurs gestiegen ist, soll die Ausgabe „bull“, sonst „bear“ lauten.

Das Programm „DWS“, welches Sie von Ihrem Teamkollegen erhalten haben, versucht diese Aufgabe zu lösen, scheitert jedoch kläglich. Ihre Aufgabe ist nun, das Programm so weit zu korrigieren und zu vervollständigen, dass es das richtige tut. Außerdem sollen Sie das Programm robuster, wartbarer und benutzerfreundlicher machen (Kommentare einfügen, Exceptions abfangen, ggf. Zwischen- oder Fehlermeldungen an den Benutzer, im Fehlerfall

Zeilennummer ausgeben, Änderung des Suchjahres). Bedenken und dokumentieren Sie auch Ihre Annahmen und die Voraussetzungen, unter denen Ihr Programm funktioniert!

Verwenden Sie bei Ihrer Arbeit die Möglichkeit, Haltepunkte zu setzen und sich Variablen-Inhalte anzeigen zu lassen. Vorgegebener Programmcode: LPSWB-DWS.zip (zum Download von der Kursseite im ILIAS). Darin ist auch die Datei mit den Fond-Daten enthalten.

B.10 Hausaufgabe (String, Parsing, Streams)

[3 Punkte]

Schreiben Sie ein Java-Programm, welches eine Eingabe (Folge von eingegebenen Zeichen) wie folgt korrigiert und die korrigierte Version zusammen mit einer Statistik ausgibt.

- Mehr als einmal auftretenden Leerzeichen und Tabulatoren sind zu eliminieren.
  Bsp: "Das Wetter ist schön. -> "Das Wetter ist schön."
- Der erste Buchstabe ('a' – 'z'; Umlaute müssen nicht berücksichtigt werden) der Eingabe sind immer in Großbuchstaben zu konvertieren. Vorausgehende Leerzeichen oder Tabulatoren sollen dabei keine Rolle spielen und ignoriert werden.
  Bsp: " das wetter ist schön." -> "Das wetter ist schön."
- Nach einem Satzzeichen (hierzu zählen: . ; : ! ?) sollen ebenfalls Buchstaben (siehe dazu oben) in Großbuchstaben konvertiert werden. Leerzeichen oder Tabulatoren zwischen dem Satzzeichen und dem Buchstaben sind dabei egal.
  Bsp: "Heute schön! morgen regen. -> "Heute schön! Morgen regen."
  Bsp.: Eine Programmsitzung könnte wie folgt aussehen:

Autokorrektur
Bitte geben Sie eine Testzeile an:
Das Wetter ist schön .
Das Wetter ist schön.
31 gelesen, 22 ausgegeben, 9 wegkomprimiert, 0 konvertiert
Noch eine Umwandlung? (j|n): j
Bitte geben Sie eine Testzeile an:
das Wetter ist schön.
Das Wetter ist schön.
27 gelesen, 22 ausgegeben, 5 wegkomprimiert, 1 konvertiert
Noch eine Umwandlung? (j|n): j
Bitte geben Sie eine Testzeile an:
heute schön! morgen regen.
Heute schön! Morgen regen.
32 gelesen, 27 ausgegeben, 5 wegkomprimiert, 2 konvertiert
Noch eine Umwandlung? (j|n): n