Programmierzertifikat Objekt-Orientierung mit Java, Blatt Nr. 5
http://proglang.informatik.uni-freiburg.de/teaching/java/2008/

Abgabe bis zum 24.6.2008, 10 Uhr per Mail an die Tutoren!

Dieses Blatt beschäftigt sich mit der Verwendung von Libraries in Java-Projekten. Zu diesem Zweck haben wir die beiden hinreichend bekannten Libraries log4j und htmlparser ausgesucht. Bevor Sie mit dem Bearbeiten des Blattes beginnen können, müssen Sie ein neues Projekt erstellen und diese Libraries sowie einige Klassenskelette für ihre Verwendung einbinden. Gehen Sie dazu folgendermaßen vor:

1. Laden Sie alle Dateien die Sie unter http://proglang/teaching/java/2008/src/blatt5/ finden herunter. Sie sollten jetzt folgende Dateien in einem lokalen Verzeichnis haben:
   - log4j-1.2.15.jar
   - htmlparser.jar
   - table.html
   - LibExercise.java
   - ReadingVisitor.java
   - TableVisitor.java
2. Erstellen Sie ein neues Java-Projekt mit dem Titel „LibExcercise“ in Eclipse.
3. Öffnen Sie in dem neu erstellten Projekt durch Rechtsklick auf den Projektnamen den Dialog „Build Path → Add external archives...“ und fügen Sie dort die vorher heruntergeladenen .jar Dateien dem Projekt hinzu.
4. Fügen Sie die Klassenskelette zu Ihrem Projekt hinzu.
   Klicken Sie dazu mit der rechten Maustaste auf den Ordner „src“ in Ihrem Projekt und öffnen Sie den Dialog „Import...“.
   Wählen Sie in diesem Dialog „General → File System“ und klicken Sie auf den Button „Next“.
   Geben Sie nun das lokale Verzeichnis an, in dem Sie Ihre Dateien gespeichert haben. Markieren Sie alle Dateien außer den bereits vorher importierten .jar Dateien und klicken Sie auf den Button „Finish“.
5. Versuchen Sie nun das Projekt auszuführen. Sie sollten keine Fehlermeldungen und die Konsolen-Ausgabe „Ich wurde richtig eingebunden!“ erhalten.

5.1 Aufgabe, Loggen mit log4j

Die Library log4j ermöglicht viele spezialisierte Operationen um die Ausgabe von textuellen Informationen zentral zu kontrollieren. In der Klasse LibExercise wird in Zeile 47 die Nachricht

Ich wurde richtig eingebunden!

mit Hilfe dieser Bibliothek ausgegeben. Sie erscheint in der Konsole genau wie in diesem Text. Sie können jedoch durch log4j auch automatisch andere Informationen an Ihre Ausgabe anhängen. Dazu müssen Sie das an den Logger angebundene Layout ändern. Ändern Sie mit Hilfe des Layouts Ihre Ausgabe zu

2008-06-14 14:14:19,619 INFO [LibExercise.java:47]: Ich wurde richtig eingebunden!

Geben Sie diese Ausgabe zusätzlich noch in eine Datei aus. Verwenden Sie dazu einen geeigneten Appender dieser Bibliothek.

In Zeile 50 wird außerdem der Level der Logausgabe gesetzt. Die Unterscheidung zwischen verschiedenen Leveln hilft Benutzern wie Entwicklern die Dichte der Informationen Ihres Programms zu kontrollieren.

Machen Sie sich beim Lesen der Dokumentation mit diesem Konzept vertraut und ändern Sie den Wert geeignet ab.

Denken Sie sich außerdem eine Möglichkeit aus, mit log4j eine Replay-Funktion für Ihr Tetris-Projekt zu schreiben (Sie müssen diese nicht implementieren).

Unter http://logging.apache.org/log4j/1.2/index.html finden Sie die Dokumentation für log4j. Sie finden dort alle Informationen die Sie zum Bearbeiten dieser Aufgabe benötigen.

Kommentieren Sie jede Methode und jedes Klasse Ihrer Abgabe mit korrekten Javadoc Kommentaren. Ohne Kommentare wird die Abgabe nicht gewertet!

5.2 Aufgabe, Parsen und Ausgeben einer HTML Datei mit htmlparser

Die Library htmlparser stellt viele Methoden zum Lesen und Manipulieren von Websites aus verschiedenen Quellen bereit. Sie wandelt dazu eine Website in eine Baumstruktur um, in der jedes HTML-Tag und jeder Text einen Knoten darstellt. Benutzer können auf diese Struktur auf verschiedene Arten zugreifen. Neben dem manuellen Zugriff auf den Baum stellt htmlparser noch Filter und Visitors (s.u.) zur Verfügung.

Die Klasse ReadingVisitor überschreibt alle Methoden der Visitor-Basisklasse NodeVisitor, die von htmlparser zur Verfügung gestellt wird. Ändern Sie diese Klasse so ab, das Sie den kompletten Sourcecode der HTML-Datei in einem String zusammen fügt, den Sie dann in LibExercise mit log4j ausgeben können. Verwenden Sie dazu das StringBuilder Objekt und die toString() Methode, die Sie im Klassenskelett finden. Entfernen Sie außerdem alle für Sie unnötigen Methoden aus der Klasse und kommentieren Sie die restlichen mit korrekten Javadoc Kommentaren.

In LibExercise finden Sie die Methode runVisitor(String url, NodeVisitor visitor), mit der Sie Ihren Visitor auf die Datei table.html anwenden können. Kommentieren Sie dazu die Anweisung in Zeile 54 von LibExercise aus und ersetzen Sie den zweiten Parameter durch eine Referenz auf eine Instanz Ihrer ReadingVisitor Implementation.

Unter http://htmlparser.sourceforge.net/ finden Sie die Dokumentation für htmlparser. Sie finden dort alle Informationen die Sie zum Bearbeiten dieser Aufgabe benötigen.

Unter http://en.wikipedia.org/wiki/Visitor_pattern finden Sie eine Erklärung der grundlegenden Funktionsweise eines Visitors.

Kommentieren Sie jede Methode und jedes Klasse Ihrer Abgabe mit korrekten Javadoc Kommentaren. Ohne Kommentare wird die Abgabe nicht gewertet!

5.3 Aufgabe, Parsen und Sortieren einer HTML Tabelle

Die von Ihnen bereits gelesene Datei table.html beinhaltet eine einzelne HTML-Tabelle mit drei Spalten. Lesen Sie diese Tabelle mit einem geeigneten Visitor ein und sortieren Sie sie nach der ersten Spalte.
Fügen Sie dazu jede Zeile der Tabelle in Ihre Implementation eines binären Suchbaums aus Blatt 4 ein. Verwenden Sie das erste Feld einer Zeile als Schlüssel und alle folgenden Felder als Wert eines Schlüssel-Wert-Paares.
Die erste Spalte enthält sowohl Zahlen als auch Text. Ihre Datenstruktur sollte zuerst alle Textschlüssel nach ihrer lexikalischen Ordnung und danach alle Zahlen nach ihrer natürlichen Ordnung sortieren.
Falls Sie Blatt 4 nicht vollständig abgeschlossen haben, suchen und verwenden Sie eine geeignete Implementierung des Interfaces java.util.SortedMap.
Verwenden Sie für diese Aufgabe die Klasse TableVisitor, die ein weiteres Skelett für einen Visitor bereitstellt und entfernen Sie auch hier alle für Sie unnötigen Methoden.
Geben Sie danach die so entstandene Datenstruktur mit Hilfe der convertMapToString(Map map) Methode in LibExercise aus.

Tipp: Verwenden Sie Listen für die Werte einer Zeile. Verwenden Sie **switch**-Statements um in den Methoden Ihres Visitors für verschiedene Zustände verschiedene Aktionen auszuführen.

Kommentieren Sie jede Methode und jedes Klasse Ihrer Abgabe mit korrekten Javadoc Kommentaren. Ohne Kommentare wird die Abgabe nicht gewertet!

5.4 Aufgabe, Aendern der Sortierung der Tabelle in der HTML Datei

Schreiben Sie einen neuen Visitor namens RewritingVisitor der die Tabelle in table.html entsprechend Ihrer Sortierung aus Aufgabe 5.3 verändert. Alle für die Sortierung nicht relevanten Daten der HTML-Datei (wie z.B. die Attribute der HTML-Tags) sollen bei dieser Transformation erhalten bleiben. Geben Sie die neue HTML-Datei in die Datei table_out.html aus.
Die Library htmlparser unterstützt Sie nicht bei der Ausgabe der neuen Datei. Finden Sie mit Hilfe der Java API geeignete Mittel, um einen String in eine Datei zu schreiben oder verwenden Sie Methoden der log4j Library.

Überlegen Sie sich, wie Sie die verschiedenen Methoden aus den Aufgaben 5.2, 5.3 und 5.4 zur Ausgabe einer Highscore-Liste im HTML Format für Ihr Tetris-Projekt verwenden könnten.

Tipp: Verwenden Sie die selbe Vorgehensweise wie in TableVisitor in dieser Aufgabe wieder. Nutzen Sie die Tatsache, das sich die Tabelle in der Datei in Ihrer Größe und Form nicht geändert hat. Leiten Sie von der Klasse ReadingVisitor ab, um die Ausgabe des Source Codes ohne zusätzlichen Aufwand zu bekommen.

Kommentieren Sie jede Methode und jedes Klasse Ihrer Abgabe mit korrekten Javadoc Kommentaren. Ohne Kommentare wird die Abgabe nicht gewertet!