



Universität
Zürich^{UZH}

Institut für Informatik



Software Engineering Übung 4

Architektur, Modulentwurf

1 Informationen

1.1 Daten

- Ausgabe Di 09.11.2010
- Abgabe So 21.11.2010 bis 23:59 Uhr
- Besprechung am Di 30.11.2010 um 11:50 Uhr

1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex[n]_[NameA_NameB_NameC].pdf** abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA_NameB_NameC] die Nachnamen der Gruppenmitglieder sind. Die PDF Datei sollte ausserdem ebenfalls Ihre Namen und Matrikelnummern beinhalten.

Mailen sie Ihre Lösungen vor dem Abgabetermin an charrada@ifi.uzh.ch und wueest@ifi.uzh.ch. Der Betreff der E-mail sollte mit **[SE EX HS10]** beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, mailen Sie bitte ein Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen sie das Archiv anhand der oben erwähnten Konventionen.

Die Übungen sollen in 3er Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

2 Aufgabenstellung

Die folgenden Fragen nehmen Bezug auf die Projekte Heilenspital (Übung 2) und ZXing (Übung 1).

2.1 Architekturstile (8 Punkte)

Im folgenden beschreibt die Spitalleitung einige Anforderungen an das neue System. Ihre Aufgabe ist es, für jeden der zwei Fälle jeweils einen adäquaten Architekturstil auszuwählen. Begründen Sie Ihre Auswahl, indem Sie erklären, wie der Architekturstil die genannten Anforderungen adressiert.

2.1.A Verlassen des Spitals (4 Punkte)

Wenn ein Patient das Spital verlässt, schliesst der Empfang die medizinische Akte. Das ihm zugeordnete Bett wird automatisch als verfügbar markiert und die Bettenverwaltung erhält eine Benachrichtigung, dass das Zimmer gereinigt werden kann. Eine Kopie der Patientenakte wird automatisch der Rechnungsstelle übergeben.

2.1.B Abrechnung (4 Punkte)

Für den Abrechnungsprozess werden der Patientenakte zuerst sämtliche Informationen entnommen, welche für die Abrechnung relevant sind. Dazu gehören die Länge des Aufenthaltes, das verbrauchte Spitalmaterial, die involvierten Ärzte und weitere Daten. Der Preis wird berechnet, indem die Dauer der jeweiligen Posten mit einer Liste von Minutenpreisen für jede Aktivität abgeglichen wird. Die Preise und die Kontaktinformationen des Patienten werden anschliessend als Textfile abgelegt, welches die Rechnung darstellt. Als letzter Schritt wird die Rechnung gedruckt und an den Patienten versendet.

2.2 Entwurfsmuster (Design Patterns) (12 Punkte)

Um diese Aufgabe zu lösen, benötigen Sie eventuell zusätzliches Material über Entwurfsmuster. Einige Kopien von [1] sind zur Konsultation in unserer Forschungsgruppe erhältlich (schauen Sie einfach im Assistentenbüro, BIN 2.B.17, vorbei). Sie können auch auf dem Web viele Informationen zu Entwurfsmustern finden.

Die folgenden Teilaufgaben behandeln einige der 23 Entwurfsmuster von [1]. Das Dokument *design_patterns.pdf* gibt einen Überblick über diese Entwurfsmuster.

Die folgenden Fragen nehmen Bezug auf das ZXing Projekt (Übung 1). Es können jedoch Widersprüche zwischen den Aussagen in der Übung und der Softwareimplementierung auftreten. In solchen Fällen erstellen Sie Ihre Antworten bitte basierend auf den Annahmen in der Übung.

2.2.A Strukturelle Muster (3 Punkte)

Diese Frage bezieht sich auf das Klassendiagramm in Bild 1.

Es gibt zwei verschiedene Arten von Readern im ZXing Projekt: Einfache Reader, die einen bestimmten, bekannten Typ von Barcode decodieren (solche Klassen erben von `OneDReader`), und Multi-Format Reader, die verschiedene Reader vereinen. Welches Strukturpattern kommt hier zum Einsatz? Weshalb wird es verwendet?

2.2.B Erzeugungsmuster (5 Punkte)

Wir würden gerne die Erzeugung des passenden Readers in Abhängigkeit des zu dekodierenden Barcodes gestalten. Welches Designpattern schlagen Sie für diesen Zweck vor und weshalb? Zeichnen Sie ein Klassendiagramm Ihres Vorschlages. Implementieren Sie einen Prototypen Ihrer Lösung, in welchem Sie die Schlüsselklassen, Methoden und die Beziehungen zwischen ihnen spezifizieren (nur Prototyp, keine richtige Implementierung notwendig!)

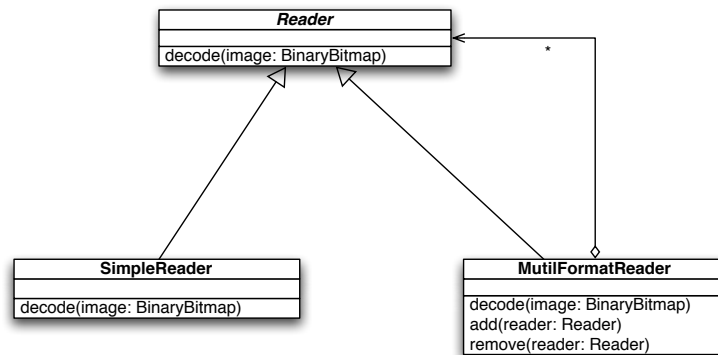


Abbildung 1: Barcode-Leser

2.2.C Verhaltensmuster (4 Punkte)

Die Dekodierroutine basiert auf folgendem Algorithmus: Zuerst wird das Bild gescannt und nach einer dekodierbaren Reihe durchsucht. Wird keine dekodierbare Reihe gefunden, rotieren wir das Bild um 180 Grad und scannen es erneut. Anschliessend wird die Reihe dekodiert. Jeder der Reader implementiert die folgenden Methoden (die Implementierung kann je nach Barcode unterschiedlich sein):

```

Row scan(Image image)
void reverseImage(Image image)
Result decodeRow(Row row)
  
```

Welches Designpattern würden Sie verwenden, um sicher zu gehen, dass alle Reader dem genannten Algorithmus folgen? Zeichnen Sie ein Klassendiagramm Ihres Vorschlages. Implementieren Sie einen Prototypen Ihrer Lösung, in welchem Sie die Schlüsselklassen, Methoden und die Beziehungen zwischen ihnen spezifizieren (nur Prototyp, keine richtige Implementierung notwendig!)

Literatur

- [1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.