Institut für Theoretische Informatik
Logik und Formale Methoden
Prof. Dr. P. H. Schmitt,
D. Faragó, C. Scheben, M. Ulbrich, B. Weiß

13.12.2010

Praktikum
Formale Entwicklung objektorientierter Software
Übungsblatt 6: Praktikumsprojekt

Aufgabe 15 — Spezifikation und Implementierung

Das bereits aus früheren Aufgaben bekannte Bankenszenario soll nun vollständig in Java implementiert werden. Hierzu ist im Folgenden eine Liste von Anforderungen angegeben, die umzusetzen sind. Als "optional" markierte Anforderungen müssen nicht unbedingt umgesetzt werden. Verwenden Sie beim Entwickeln des Programms JML, um Ihr System möglichst weitgehend zu spezifizieren. Setzen Sie nach eigenem Ermessen auch RAC, JMLUnit und/oder ESC/Java2 ein, um Ihr System zu testen / teilweise zu verifizieren.

Es soll ein Banksystem modelliert werden.	REQ-0

In dieser Aufgabe soll nur das Banksystem selbst modelliert werden. Kunden brauchen nicht modelliert zu werden.

Konten

Es existieren Bankkonten.	REQ-43
Es können jederzeit neue Konten eröffnet werden.	REQ-9'
Ein bereits existierendes Konto kann nicht erneut eröffnet werden.	REQ-10
Konten werden über ihre Kontonummer identifiziert.	REQ-44
Kontonummern sind ganze Zahlen zwischen 0 (einschliesslich) und einer festgelegten oberen Schranke.	REQ-45
Über auf Konten ausgeführte Aktionen wird Buch geführt.	REQ-46

Implementieren Sie die Liste der für ein Konto gespeicherten Aktionen als Array mit dynamischer Größe. Benutzen Sie hierzu eine Methode, die ein neues, größeres Array anlegt, wenn das alte voll ist, und die Einträge kopiert. Definieren Sie desweiteren eine Methode für das Hinzufügen neuer Aktionen, die obige Methode bei Bedarf aufruft.

Kontostände

Jedes Konto hat einen Saldo.	REQ-11
Der Saldo wird als ganze Zahl von Geldeinheiten gespeichert.	REQ-12
Konten können überzogen werden.	REQ-13

Neu eingerichtete Konten sind zu Beginn leer.	REQ-14

Kunden können beliebige positive Beträge auf eines ihrer eigenen Konten einzahlen.	REQ-16

Auf welche Weise die Einzahlung erfolgt interessiert uns nicht weiter, es muss nur eine entsprechende Operation auf Bankkonten definiert sein.

Überziehungslimit

Es gibt einen Minimalbetrag, der für alle Konten derselbe ist.	REQ-22'

Der Minimalbetrag für Kontostände ist nicht positiv.	REQ-24

Wählen Sie z. B. einen Minimalbetrag von $-100\,€$. Stellen Sie aber sicher, dass er leicht geändert werden kann.

Zentralrechner

Es gibt einen Zentralrechner, der alle Bankkonten der Bank verwaltet.	REQ-47

Kontoeröffnungen werden vom Zentralrechner der Bank abgewickelt.	REQ-48

Zentralrechneroperationen finden am Bankschalter (hier nicht modelliert) statt.

Bankautomaten und Bankkarten

Es existieren Bankkarten für Konten.	REQ-25

Jede Bankkarte ist genau einem Konto zugeordnet, wobei es mehrere Karten für ein Konto geben kann (oder auch gar keine).	REQ-26

Es können jederzeit neue Karten für existierende Konten ausgegeben werden.	REQ-27

Kartenausgaben werden vom Zentralrechner der Bank abgewickelt.	REQ-49

Es gibt nur Karten für existierende Konten.	REQ-50

Es existieren Bankautomaten.	REQ-51

Bankautomaten sind per Netzwerk mit dem Zentralrechner verbunden (online).	REQ-52

Die Verbindung eines Automaten zum Zentralrechner kann unterbrochen werden, der Automat ist dann offline.	REQ-53

Wer im Besitz einer Karte ist, kann an einem Bankautomaten verschiedene Dienste für das zugehörige Konto in Anspruch nehmen.	REQ-28'

Authentifizierung am Automaten

Jede Karte hat eine unveränderliche, vierstellige PIN.	REQ-29
Karten können in Bankautomaten eingeführt werden, zu jedem Zeitpunkt ist höchstens eine Karte im Automaten.	REQ-34
Zu jedem Zeitpunkt ist eine Karte in höchstens einem Automaten.	REQ-35
Ein Benutzer kann seine eingeführte Karte jederzeit vom Automaten ausgeben lassen.	REQ-36
Der Karteninhaber muss sich am Automaten nach Einführen der Karte durch Eingabe der PIN authentifizieren.	REQ-37
Bei Eingabe der korrekten PIN ist der Benutzer authentifiziert, bis die Karte wieder ausgegeben wird.	REQ-38
Wird für eine Karte dreimal in Folge (also bei drei aufeinanderfolgenden Authentifizierunsversuchen, auch an verschiedenen Automaten) eine falsche PIN eingegeben, so ist die Karte danach gesperrt.	REQ-39
Eine gesperrte Karte wird vom Automaten sofort eingezogen und nicht wieder ausgegeben. Der Automat ist dann wieder kartenfrei.	REQ-40
Ein Automat kann die PIN einer Karte nicht auslesen. Die Funktionalität zum Authentifizieren des Benutzers und zum Sperren der Karte muss von der Karte selbst bereitgestellt werden.	REQ-41
Automaten können von Karten abfragen, ob sie gesperrt sind.	REQ-42
Ein Kunde kann eine gesperrte Karte, die zu einem seiner Konten gehört, wieder freischalten lassen.	REQ-33
Gesperrte Karten werden durch den Zentralrechner freigeschaltet.	REQ-54

Die hierzu nötige Hardware ist irrelevant.

Dienste an Automaten

Sämtliche (kontenbezogene) Dienste am Bankautomaten können nur in Anspruch genommen werden, solange der Kunde authentifiziert ist.	REQ-55

Kontostandabfrage

Am Automaten kann genau im Onlinefall (d. h. wenn der Automat mit dem Zentralrechner verbunden ist) der aktuelle Kontostand abgefragt werden.	REQ-56

Kontoauszüge (optional)

Am Automaten kann die Zusendung eines schriftlichen Kontoauszugs angefordert werden.	REQ-57

Die Anforderung eines schriftlichen Kontoauszugs wird im Onlinefall direkt an den Zentralrechner weitergeleitet, im Offlinefall im Automaten gespeichert.	REQ-58

Ein Kontoauszug pro Monat und Konto ist kostenlos, für weitere Kontoauszüge wird eine Gebühr von je 1 € erhoben.	REQ-59

Gebührenpflichtige Kontoauszüge werden nur dann versandt, wenn die Gebühr erfolgreich vom Konto abgebucht werden konnte, sonst wird die Anfrage ignoriert.	REQ-60

Überweisungen (optional)

Kunden können von ihren Konten Geld auf existierende Konten überweisen.	REQ-18

Wir betrachten hier keine Konten bei anderen Banken.

Überweisungen können am Bankautomaten getätigt werden. Der Automat muss dazu online sein.	REQ-61

Bei einer Überweisung verringert sich der Saldo des Ausgangskontos um einen Betrag. Um denselben Betrag erhöht sich der Saldo des Zielkontos.	REQ-19

Bei einer Überweisung müssen Quell- und Zielkonto verschieden sein.	REQ-21

Abhebungen

Am Automaten können Geldbeträge in Bar abgehoben werden.	REQ-62

Im Onlinefall wird ein ausgezahlter Geldbetrag sofort vom Konto abgebucht.	REQ-63

Im Offlinefall wird die Auszahlung von Geldbeträgen im Automaten gespeichert.	REQ-64

Von einem Konto dürfen pro Tag höchstens 1000 € abgebucht werden. Diese Grenze kann nur durch Offlineabhebungen überschritten werden.	REQ-65

Bedenken Sie, dass es auch bei Überweisungen und gebührenpflichtigen Kontoauszügen zu Abbuchungen kommt.

Der Gesamtbetrag von Offlineabhebungen darf bis zur Verbuchung auf dem Konto pro Konto und Bankautomat 1000 € nicht überschreiten.	REQ-66

Allgemeines

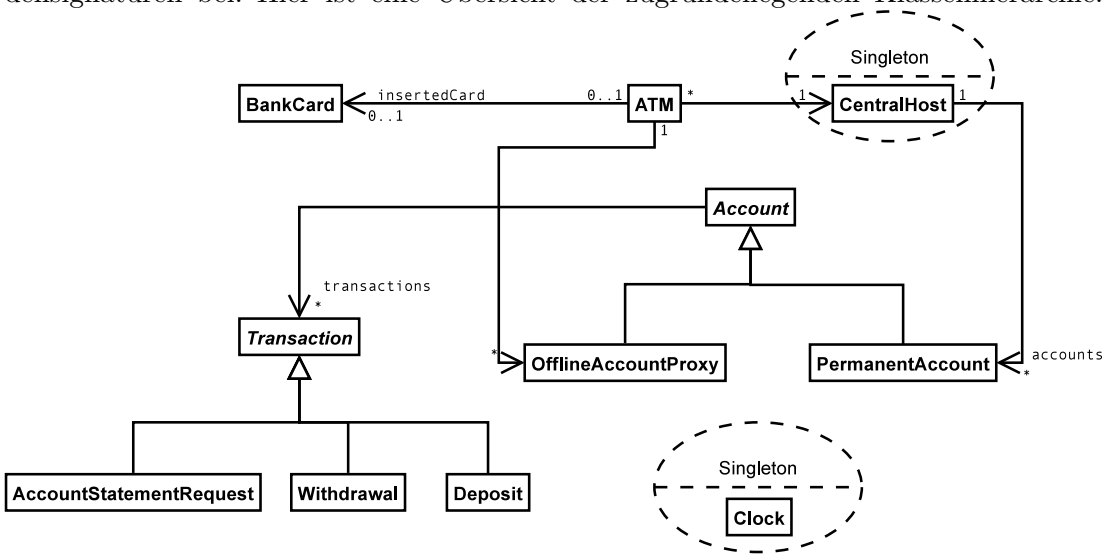
Eine Aktion, die mit einer Abbuchung verbunden ist – ausgenommen Offlineabhebungen –, nach welcher der Minimalbetrag des zugehörigen Kontos unterschritten wäre, wird nicht durchgeführt, sondern abgebrochen.	REQ-67

Der Minimalbetrag kann durch Offlineabhebungen unterschritten werden.	REQ-68

Wenn ein Automat nach einer Offlinephase wieder Verbindung zum Zentralrechner herstellen kann, werden alle gespeicherten, offline durchgeführten Aktionen ausgeführt.	REQ-69

Erläuterungen und Hinweise:

- Bauen Sie Ihre Implementierung auf den vorgegebenen Klassen auf, die Sie auf der Praktikums-Homepage finden (Datei `BankDateien.zip`). Sie können die Klassen um weitere Attribute und Methoden ergänzen, behalten Sie aber die gegebenen Methodensignaturen bei. Hier ist eine Übersicht der zugrundeliegenden Klassenhierarchie:



- Um Ihre Implementierung auszuprobieren, finden Sie ebenfalls auf der Praktikumsseite in der Datei `BankGui.zip` eine graphische Benutzeroberfläche (GUI). Die `main`-Methode befindet sich in der Klasse `BankGui`. Bitte halten Sie aber die Programmlogik von der GUI strikt getrennt – die Programmlogik soll von der Existenz der GUI nichts wissen.

- Der Einfachheit halber wird eine globale Uhr angenommen. Ausgeführte und auszuführende Aktionen werden in `Transaction`-Objekten gespeichert. Wenn ein Bankautomat offline ist, führt er Kontenoperationen auf lokalen (also im Automaten gespeicherten) Stellvertretern vom Typ `OfflineAccountProxy` durch.

- Da Ihr Code zumindest teilweise auch verifiziert werden soll, sollten Sie sich von vorneherein an einige Einschränkungen halten, um ESC/Java2 und KeY nicht zu überfordern. Insbesondere sollten Sie die folgenden Java-Features *nicht* verwenden:

  – *Jegliche Klassen und Interfaces der Java-Klassenbibliothek (!)*, mit Ausnahme der folgenden:

    * `java.lang.Object`
    * `java.lang.Throwable` sowie seine in `java.lang` definierten Unterklassen

    * `java.lang.String`.

    Verwenden Sie z.B. Arrays anstelle von Klassen wie `java.util.Vector`.
  - Features, die in *Java 5* neu eingeführt wurden, wie Generics, Enumerations, Java-5-Annotationen, oder for-each-Schleifen.
  - Gleitkommazahlen, d.h. die Typen `float` und `double`.
  - automatische Konvertierungen anderer Typen in Strings, wie z.B. in dem Ausdruck `"Hello World" + 7`,
  - Unicode-Zeichen in .java-Dateien, die über den 7-bit-ASCII-Zeichensatz hinausgehen (also insbesondere keine Umlaute).

- Auch auf JML-Seite wollen wir nicht die komplette (sehr umfangreiche) Sprache verwenden. Beschränken Sie sich in Ihren Spezifikationen bitte auf folgende Elemente von JML:
  - JML-Ausdrücke einschließlich `==>`, `<==>`, `\result`, `\old`, `\forall`, `\exists`, `\fresh`
  - Methodenverträge mit Vor- und Nachbedingungen und evtl. diverges-Klauseln (`requires`, `ensures`, `signals`, `signals_only`, `normal_behavior`, `exceptional_behavior`, `also`, `diverges`)
  - `assignable`-Klauseln einschließlich der zugehörigen Spezialausdrücke (`a[i..j]`, `a[*]`, `\nothing`, `\everything`), sowie `pure`-Methoden
  - Klasseninvarianten (`invariant`)
  - Sichtbarkeitsmodifikatoren (`private`, `protected`, `public`, `spec_protected`, `spec_public`)
  - `non_null` und `nullable`
  - Assertions (`//@ assert`), Schleifeninvarianten (`loop_invariant`) und Schleifenvarianten (`decreasing`)

  Benutzen Sie insbesondere *keine* Modellfelder.

- Aufrufe von `/*@pure@*/`-Methoden in Spezifikationen können Sie zwar verwenden, es ist aber empfehlenswert, die Spezifikationen stattdessen so weit wie möglich direkt mit den zugrundeliegenden Daten (Attribute, Array-Einträge) zu formulieren, um die Verifikation einfacher zu machen.

- Die Möglichkeit von Integer-Überläufen können Sie ignorieren. Dieser Aspekt wird auch bei der Verifikation hier im Praktikum nicht betrachtet.

Wir wünschen Euch ein frohes Fest und einen guten Rutsch ins neue Jahr!

Abgabe am 28.01.2011

Die Abgabe erfolgt durch Vorführen während des Praktikumstermins. Bitte dokumentieren Sie Ihre Lösungen ausreichend und seien Sie darauf vorbereitet, sie auf Nachfrage zu erklären.

Praktikums-Webseite: `http://lfm.iti.kit.edu/keyprakt1011.php`

David Faragó:	R. 308,	Tel. 608-7322,	E-Mail: `farago@ira.uka.de`
Christoph Scheben:	R. 106,	Tel. 608-4338,	E-Mail: `scheben@ira.uka.de`
Mattias Ulbrich:	R. 106,	Tel. 608-4338,	E-Mail: `mulbrich@ira.uka.de`
Benjamin Weiß:	R. 309,	Tel. 608-4324,	E-Mail: `bweiss@ira.uka.de`