

3. Aufgabenblatt - Smalltalk

Ausgabe: 12. Mai 2009 Abgabe: 19. Mai 2009 Revision: 3094

1 Bankkonten

Implementieren Sie zwei Klassen `Account` und `AccountWithoutOverdraft`, die jeweils ein Bankkonto repräsentieren. Erstere bildet ein Konto ab, das überzogen werden kann. Letztere eines, das nicht überzogen werden kann, d.h. der Kontostand kann nie unter 0 fallen.

1.1 Vorgaben und Spezifikation

Die Klasse `Account` soll Oberklasse von `AccountWithoutOverdraft` sein. Die Klassen stellen zwei Methoden zur Verfügung:

1. `spend: amount` Über diese Methode kann Geld ausgegeben bzw. ausgezahlt werden. Der einzige Parameter gibt an, wie viel Geld vom Konto genommen werden soll.
2. `deposit: amount` Mit dieser Methode kann Geld eingezahlt werden. Der einzige Parameter gibt an, wie viel Geld auf das Konto eingezahlt werden soll.

Weiter soll die Methode `printOn: aStream` der Klasse `Object` überschrieben werden, um das Konto in einer lesbaren Form ausgeben zu können.

Würde auf einer Instanz von `AccountWithoutOverdraft` die Methode `spend` mit einem Betrag aufgerufen, der größer als das aktuelle Guthaben ist, so soll eine Ausnahme (Exception) geworfen werden.

Das folgende Listing zeigt beispielhaft, wie mit den beiden Konten interagiert werden können soll:

```
1 account := Account new.  
2 account printNl.  
3  
4 account deposit: 10.  
5 account printNl.  
6
```

```
7  account spend: 5.
8  account printNl.
9
10 account spend: 10.
11 account printNl.
12
13 noOverdraft := AccountWithoutOverdraft new.
14 noOverdraft printNl.
15
16 noOverdraft deposit: 10.
17 noOverdraft printNl.
18
19 noOverdraft spend: 15.
20 noOverdraft printNl.
```

Listing 1: Geld einzahlen und abheben

Dabei wäre folgende Ausgabe zu erwarten:

```
1  an Account with balance: 0
2  an Account with balance: 10
3  an Account with balance: 5
4  an Account with balance: -5
5  an AccountWithoutOverdraft with balance: 0
6  an AccountWithoutOverdraft with balance: 10
7  Object: AccountWithoutOverdraft new "<0x1317a78>" error: Not
   enough money
8  Error(Exception)>>signal (AnsiExcept.st:216)
9  Error(Exception)>>signal: (AnsiExcept.st:226)
10 AccountWithoutOverdraft(Object)>>error: (AnsiExcept.st:1565)
11 AccountWithoutOverdraft>>spend: (bankaccount.st:42)
12 UndefinedObject>>executeStatements (bankaccount.st:65)
13 an AccountWithoutOverdraft with balance: 10
```

Listing 2: Erwartete Ausgabe

1.2 Dokumentation

Dokumentieren Sie Ihre Implementierung durch Kommentare im Quellcode. Ihr Programmcode muss durch dritte Personen schnell verstanden werden können: Stellen Sie sich vor, Sie müssen Ihren Quellcode einem anderen Programmierer erklären. Der Programmierer ist unerfahren in dieser Programmiersprache.

1.3 Abgabe Ihrer Lösung

Drucken Sie Ihre Implementierung des Unterprogrammes aus und werfen Sie den Ausdruck bis zum **19. Mai 2009 um 9:00 Uhr** in den Briefkasten des Lehrstuhles für Softwaretechnik¹. Vergessen Sie dabei nicht, Ihre Implementierung mit Ihrem Namen und Ihrer Matrikelnummer zu versehen.

Bringen Sie zum Proseminar ebenfalls einen Ausdruck mit. Wir werden zufällig einen Studenten auswählen, der seine Lösung erklärt.

2 Diskussion

Diskutieren Sie folgende Fragestellungen schriftlich auf etwa einer DIN A4 Seite. Die schriftliche Bearbeitung dient im wesentlichen Ihrer Vorbereitung auf das Proseminar. Dort werden wir die Fragen gemeinsam diskutieren. Wir erwarten von jedem Teilnehmer, dass er sich an der Diskussion aktiv beteiligt.

Werfen Sie Ihre Ausarbeitung zusammen mit dem Ausdruck Ihrer Implementierung in den Briefkasten ein.

2.1 Objekte und Nachrichten

Welche der in Smalltalk direkt eingebauten, ausführbaren Konstrukte werden im folgenden Programm verwendet? Welche Nachrichten werden an welche Objekte gesendet?

```
1 m := 2 + 2.  
2 m even  
3   ifTrue: [ 'm_ist_gerade' printNl ]  
4   ifFalse: [ 'm_ist_ungerade' printNl ]
```

Listing 3: Objekte und Nachrichten

2.2 Virtuelle Maschine

Welche Vorteile hat eine "Virtuelle Maschine", wie sie in Smalltalk eingesetzt wird? Welche Rolle spielt "Byte Code" in diesem Zusammenhang?

¹Gebäude E1 1, neben dem InfoPointes des Rechenzentrums.

2.3 Automatische Speicherbereinigung

Erläutern Sie kurz die Funktionsweise der in Smalltalk eingebauten "Automatischen Speicherbereinigung". Welche Programmierfehler können durch dieses Verfahren vermieden werden? Worin besteht der Nachteil? Denken Sie dabei z.B. an C / C++ und Echtzeitsysteme.

3 Links und Hinweise

Sie sollten den GNU Smalltalk Compiler² verwenden, dieser ist auf den Rechnern des Studentenrechnerpools im Verzeichnis `/installer/import/linux/compilers` installiert. Es handelt sich dabei um die GNU Smalltalk Version 3.1. Loggen Sie sich bitte auf einem der Sun Compute Server³ ein.

Ein guter Startpunkt ist das Wiki der German Smalltalk Users Group⁴.

²<http://smalltalk.gnu.org/>

³appsrv1.studcs.uni-saarland.de oder appsrv2.studcs.uni-saarland.de

⁴<http://swiki.gsug.org/>