

Programmierparadigmen

Wintersemester 2012/13
Torsten Görg, Sandro Degiorgi

6. Übung

Einsendung bis spätestens 21.01.2013, 13:00 Uhr
Besprechungen am 23./24./25. Januar 2013

Aufgabe 6.1 22 Punkte

Gegeben sei der folgender Teil aus einer Ada-Paketspezifikation.

Details zur objektorientierten Programmung in Ada finden Sie z.B. unter

http://en.wikibooks.org/wiki/Ada_Programming/Object_Orientation.

```
1  type A is tagged private;  
2  type B is new A with private;  
3  type C is new A with private;  
4  type D is new C with private;  
5  
6  procedure Method_1(Param1:in out A);  
7  procedure Method_1(Param1:Float; Param2:in out A);  
8  procedure Method_1(Param1:in out D);  
9  procedure Method_1(Param1:Integer; Param2:in out B);  
10 procedure Method_1(Param1:Float; Param2:in out C);  
11 procedure Method_2(Param1:Float; Param2:Boolean);  
12 procedure Method_2(Param1:Boolean);  
13 procedure Method_3(Param1:in out B; Param2:in out B; Param3:Integer);  
14 procedure Method_3(Param1:in out D; Param2:in out D; Param3:Integer);  
15 procedure Method_3(Param1:in out A; Param2:in out A; Param3:Integer);  
16 procedure Method_4(Param1:Integer; Param2:A; Param3:A);  
17 procedure Method_4(Param1:C; Param2:C);  
18 procedure Method_4(Param1:Float);  
19 procedure Method_4(Param1:D; Param2:D; Param3:Integer);
```

1. Welche Vererbungshierarchie bilden die Tagged-Types in diesem Programm? Geben Sie dazu alle direkten Vererbungsbeziehungen zwischen den Tagged-Types an.
2. Eine primitive Operation ist eine Operation, die einem Tagged-Type zugeordnet ist, also eine Methode. Welche der obigen Operationen (bitte Zeilennummer angeben) sind primitive Operationen, und welchem Tagged-Type sind sie jeweils zugeordnet?
3. Identifizieren Sie (unter Angabe der jeweiligen Zeilennummern), welche der Operationen in obigem Quelltext Überladungen zueinander sind, und welche Operation eine Redefinition von welcher anderen Operation ist.
4. Geben Sie für jeden Parameter aller primitiven Operationen an, mit welcher Varianz dieser Parameter bei einer Vererbung der primitiven Operation an den direkt abgeleiteten Tagged-Type vererbt wird bzw. würde (invariant, kovariant oder kontravariant).
5. Geben Sie für jede primitive Operation, die nicht in der direkt darunter liegenden Ebene der Vererbungshierarchie redefiniert wird, die Operations-Signatur an, mit der diese Operation durch die Vererbung implizit im direkt abgeleiteten Tagged-Type erscheint.

Aufgabe 6.2 18 Punkte

1. Erstellen Sie Java-Klassendefinitionen aus den folgenden, textuell beschriebenen Anforderungen für ein einfaches Modell für Universitäten. Jedes im Text vorkommende Substantiv soll durch eine Java-Klasse repräsentiert werden. Identifizieren Sie dazu alle in den textuellen Anforderungen angedeuteten Beziehungen zwischen den Substantiven und drücken Sie diese Beziehungen in Ihren Java-Klassendefinitionen in geeigneter Weise aus. Beachten Sie dabei, dass die Beziehungen, die zwischen Klassen bzw. Objekten bestehen, unterschiedlicher Art sein können (Vererbung, Containment, Realisierung, etc.). In Anlehnung an die Ausführungen im Skript lässt sich die Art einer Beziehung jeweils an der verwendeten Formulierung erkennen.

Die Komponenten in den Klassen sind jeweils als `private` zu deklarieren und sollen über geeigneten Zugriffsmethoden lesbar und veränderbar sein. Mengenwertige Objektbeziehungen sollen mit der Java-Standardklasse `Vector` abgebildet werden. Die Implementierungen der Klassen sind soweit auszugestalten, dass die Einhaltung der angegebenen Constraints sichergestellt ist.

- Professoren, Mitarbeiter und Studenten sind Personen.
 - Jeder Professor hält beliebig viele Vorlesungen.
 - Jeder Vorlesung ist genau ein Hörsaal zugeordnet, in dem sie stattfindet.
 - Ein Student ist mit beliebig vielen anderen Studenten befreundet und besucht eine oder mehrere Vorlesungen.
 - Zu einer Universität gehören Studenten, Professoren und Hörsäle.
 - Ein Professor arbeitet an Forschungsprojekten und beschäftigt mehrere Mitarbeiter.
 - Ein Forschungsprojekt ist entweder ein DFG-Projekt oder ein Drittmittelprojekt.
2. Auf welche Probleme würde man bei dem Versuch stoßen, das Java-System aus dem ersten Aufgabenteil in gleicher Weise für die folgenden, zusätzlichen Anforderungen zu erweitern? Welches Problem könnte darüber hinaus in Abhängigkeit von der Benennung der Komponenten in den Klassen ggf. bestehen?
 - Ein Firmenchef ist eine Personen und beschäftigt mehrere Mitarbeiter.
 - Ein Klinikleiter ist ein Professor.
 - Ein Klinikleiter ist ein Firmenchef.

Aufgabe 6.3 20 Punkte

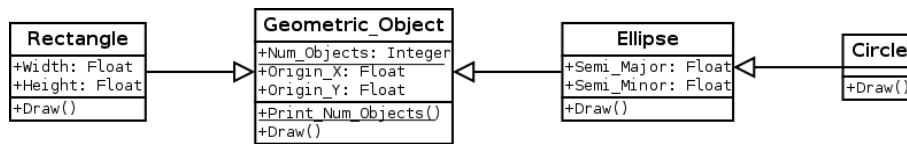
Gegeben sei folgender objektorientierter Ada-Code (als Mischung aus einer Paket-Spezifikation und der zugehörigen Paket-Implementierung):

```
1  type A is tagged null record;
2  procedure Method1( Param : A ) is begin null; end Method1;
3  procedure Method2( Param : A ) is begin null; end Method2;
4
5  type B is new A with null record;
6  procedure Method2( Param : B ) is begin null; end Method2;
7  procedure Method1( Param : B ) is begin
8      Method2 (Param);
9      Method2 (A (Param));
10     Method2 (B (Param));
11     Method2 (A'Class (Param));
12 end Method1;
13
14 type C is new B with null record;
15 procedure Method2( Param : C ) is begin
16     Method2 (A (Param));
17     Method2 (B (Param));
18 end Method2;
19
20 type D is new B with null record;
21 procedure Method2( Param : D ) is begin
22     Method2 (A (Param));
23     Method2 (B (Param));
24 end Method2;
25
26 procedure Main is
27     Object_C : C;
28     Object_D : D;
29 begin
30     Method2 (Object_C);
31     Method2 (Object_D);
32     Method2 (A (A'Class (Object_C)));
33     Method2 (B (A'Class (Object_C)));
34     Method1 (A'Class (Object_C));
35     Method1 (A'Class (Object_D));
36 end Main;
```

1. Mit Beginn der Ausführung der Prozedur `Main` werden nacheinander die Methodenaufrufe in den Zeilen 30 - 35 ausgeführt. Geben Sie für jede dieser Zeilen jeweils die komplette Kette der aufgerufenen Methoden an (bitte als qualifizierte Methodennamen oder als Zeilennummern). Achten Sie dabei insbesondere darauf, bei welchen Aufrufen Dispatching bzw. Redispaching erfolgt und bei welchen nicht.
2. Konvertieren Sie den vorliegenden Ada-Code - soweit möglich - in ein äquivalentes Java-Programmfragment. Für welche Stellen ist es nicht unmittelbar möglich, äquivalenten Java-Code anzugeben und warum nicht?

Aufgabe 6.4 20 Punkte

Gegeben sei folgende Klassenhierarchie (vgl. Übungsblatt 5, Aufgabe 4):



Bitte beachten Sie:

- Analog zur genannten Aufgabe ist der Kreis eine Spezialisierung der Ellipse. Nutzen Sie die Komponente `Semi_Major` zur Speicherung des Radius.
- Sollten Sie mit UML noch auf Kriegsfuß stehen, finden Sie auf der Veranstaltungs-Webseite die Datei `A6_4.zip`, welche die Klassenhierarchie in Java umsetzt. Die Angaben zu UML unter <http://linuxwell.com/2011/08/13/uml-class-diagram/> reichen zur Bearbeitung dieser Aufgabe aus.

Ihr Aufgabe ist es, diese Klassenhierarchie auf einem Ada83 System umzusetzen. Es stehen Ihnen also keinerlei objektorientierte Sprachmittel zur Verfügung, insbesondere keine `tagged types`. Versuchen Sie dennoch, so nahe wie möglich an der Vorlage zu bleiben.

1. Implementieren Sie eine *Konstruktor*-Funktion `Create_Geometric_Object`, die entweder ein Rechteck, eine Ellipse oder einen Kreis zurückgibt. Es soll nur eine einzige solche Konstruktor-Funktion für alle möglichen geometrischen Objekte geben.
2. Implementieren Sie eine (einzige) Prozedur `Draw`, welche, je nach Objekttyp, die Parameter des Objektes (`Width`, `Height`, usw.) ausgibt, gefolgt von den X,Y-Koordinaten des Objektes.
3. Führen Sie die (statische) Variable `Num_Objects` (und ggf. notwendige Methoden) ein, um die Anzahl der instanziierten geometrischen Objekte zu führen.
4. Für ein (von Ihnen zu erstellendes) Testprogramm, nutzen Sie bitte folgende Deklarationen:

```
Obj_1 : Geometric_Object := Create_Geometric_Object (Rectangle, 10.0, 2.0, 500.0, 300.0);
Obj_2 : Geometric_Object := Create_Geometric_Object (Circle, 6.0, 3.0, 123.0);
Obj_3 : Geometric_Object := Create_Geometric_Object (Ellipse, -2.0, -6.0, 10.0, 30.0);
```

Ihre Test-Prozedur soll für alle drei Objekte jeweils die `Draw` Prozedur aufrufen. Anschließend soll der Radius des Kreises und die Anzahl der instanziierten geometrischen Objekte ausgegeben werden. Bitte geben Sie bei der Einsendung die Bildschirmausgabe Ihres Testprogrammes an.