

Aufgabe 1

5 Punkte

Aus welchen Bestandteilen besteht ein (Smalltalk-)Programm? Ordnen Sie vom primitivsten zum komplexesten!

Lösung

1. Literal
2. Ausdruck
3. Anweisung
4. Methode
5. Klasse

Aufgabe 2

6 Punkte

Mit Hilfe welcher Beziehungen lassen sich die Klassen eines Smalltalk-Systems strukturiert anordnen?

Lösung

1. Subklassenbeziehung (Vererbung) bzw. Generalisierung/Spezialisierung
2. Instanzbeziehung von einer Klasse zu ihrer Metaklasse

Aufgabe 3

2 + 4 Punkte

- a) Wie drückt man in einem objektorientierten Programm Beziehungen zwischen Objekten aus?
- b) Welche Arten von Beziehungen zwischen Objekten unterscheiden Sie?

Lösung

- a) durch Instanzvariablen
- b) :1- und :n-Beziehungen, Aggregation und Komposition (mehr oder weniger daselbe) und solche, die es nicht sind.

Aufgabe 4**2 + 2 Punkte**

- a) Was ist ein Block?
- b) Aus welchen Bestandteilen besteht er?

Lösung

- a) eine Folge von Anweisungen
- b) Variablendeklarationen (Parameter) und Anweisungen.

Aufgabe 5**2 + 4 + 2 + 4 Punkte**

- a) Was ist eine Kontrollstruktur?
- b) Welche kennen Sie?
- c) Welche davon halten Sie für unverzichtbar?
- d) Gibt es eine Kontrollstruktur, die für die objektorientierte Programmierung charakteristisch ist?

Lösung

- a) ein Sprachkonstrukt, das den Programmablauf beeinflusst
- b) Sequenz, Verzweigung, Wiederholung und Unterprogrammaufruf
- c) Unverzichtbar sind nur die Sequenz und der dynamisch gebundene Methodenaufruf.
- d) der dynamisch gebundene Methodenaufruf

Aufgabe 6**3 + 3 Punkte**

Was ist ein Iterator und wofür braucht man ihn hauptsächlich?

Lösung

Ein Iterator ist eine Kontrollstruktur, die die Elemente einer Collection nacheinander einer Variablen zuweist und damit einen bestimmten Anweisungsblock ausführt. Man braucht ihn vor allem zum Durchwandern der Elemente einer n -Beziehung.

Aufgabe 7

5 Punkte

Erläutern Sie den Begriff der Vererbung in einem Satz.

Lösung

Bei der Vererbung werden Eigenschaften von der Definition einer Klasse, der Super- oder Elternklasse, auf eine andere übertragen, ohne daß sie in deren Definition explizit auftauchen würden.

Aufgabe 8

3 Punkte

Was bezeichnet das Schlüsselwort `self` in Smalltalk?

Lösung

eine Pseudovariablen, die sich immer auf das Empfängerobjekt der Methode bezieht, in der sie vorkommt

Aufgabe 9

7 Punkte

Erläutern Sie den Zusammenhang zwischen Klassen und Typen und sagen Sie, wozu man Typen braucht.

Lösung

Klassen können, je nach Programmiersprache, Typen definieren. Typen dienen der Typprüfung und die wiederum im wesentlichen der Erhöhung der Lesbarkeit eines Programms, der Regelung des Speicherlayouts von Objekten bzw. Variablen, der Auswahl von zu den Operanden passenden Operatoren (Performanz) sowie der

Aufdeckung von (logischen) Programmierfehlern bei nicht passenden Typen in Wertzuweisungen.

Aufgabe 10

7 Punkte

Erläutern Sie, warum es problematisch sein kann, wenn man eine geerbte Eigenschaft (Methode oder Instanzvariable) in der erbenden Klasse löscht.

Lösung

Dies ist nur problematisch, wenn die erbende Klasse zugleich einen Subtyp des Typs von der Klasse, die sie beerbt, bildet. Mit dem Begriff des Subtypings ist nämlich die Substituierbarkeit verbunden, also das Versprechen, daß eine Instanz der Subklasse überall da auftreten kann, wo eine Instanz der Superklasse erwartet wird. Fehlt aber eine Eigenschaft, die bei Instanzen der Superklasse vorhanden ist, und wird diese Eigenschaft tatsächlich gebraucht, ist die Substituierbarkeit auch nicht mehr gegeben.

Aufgabe 11

4 Punkte

Wie unterscheidet sich ein nominales von einem strukturellen Typsystem?

Lösung

Ein strukturelles Typsystem verlangt für die Zuweisungskompatibilität, daß der Typ auf der rechten Seite alle Eigenschaften des Typs auf der linken Seite umfaßt. Ein nominales Typsystem verlangt darüber hinaus, daß der Typ auf der rechten Seite deklariert, ein Subtyp des Typs auf der linken Seite zu sein.

Aufgabe 12

2 + 4 Punkte

- Was versteht man unter einem generischen oder parametrischen Typ?
- Was versteht man unter beschränktem parametrischem Polymorphismus? Wozu braucht man ihn?

Lösung

- a) ein Typ, dessen Definition Variablen, also nicht festgelegte Bestandteile enthält
- b) Ein beschränkt parametrisch polymorpher Typ ist ein generischer Typ, dessen Typvariablen nicht vollkommen frei, sondern durch andere Typen beschränkt sind. Die Typvariable kann z. B. auf Typen eingeschränkt werden (darf nur Typen enthalten), die Subtypen eines (als Schranke) vorgegebenen Typen sind. Dies ist dann sinnvoll, wenn man vom tatsächlichen (d. h. dem für die Typvariable eingesetzten) Typ bestimmte Eigenschaften erwartet, die dann durch den als Schranke angegebenen Typ vorgegeben werden. Man braucht ihn also fast immer.

Aufgabe 13**3 + 3 + 3 + 3 Punkte**

Nennen Sie jeweils ein charakteristisches Merkmal der Programmiersprachen Smalltalk, Java, C++ und Eiffel, das sie von jeweils allen anderen unterscheidet.

Lösung

Smalltalk: kein Typsystem, nur zwei Kontrollstrukturen

Java: rekursiv parametrischer Polymorphismus, syntaktische Trennung von Interfaces und abstrakten Klassen

C++: Friend-Konzept, mangelnde dynamische Typsicherheit begünstigt schwere Laufzeitfehler

Eiffel: kovariante Redefinition, Vor- und Nachbedingungen als explizites Sprachkonstrukt

Aufgabe 14**2 Punkte**

Welches praktische Problem stellt das dynamische Binden bei der Entwicklung objektorientierter Programme dar?

Lösung

schlechte Tracebarkeit aufgrund ständig wechselnder Kontexte

Aufgabe 15

3 Punkte

Was versteht man in der objektorientierten Programmierung unter Aliasing?

Lösung

den Umstand, daß mehrere Namen (Variablen) auf dasselbe Objekt verweisen können

Aufgabe 16

8 Punkte

Gegeben sei das folgende Programmfragment in Smalltalk:

• Klasse	• Skier
• benannte Instanzvariablen	• roommate
• Instanzmethoden	•
1 share: otherskier	
2 roommate := otherskier	
• Klasse	• Boy
• Superklasse	• Skier
• Klasse	• Girl
• Superklasse	• Skier

Versuchen Sie, das Fragment so zu erweitern, daß Jungen und Mädchen nicht gemischt untergebracht werden können.

Lösung

• Klasse	• Skier
• benannte Instanzvariablen	• roommate
• Instanzmethoden	•
3 share: otherskier	
4 self class == otherskier class	
5 ifFalse: [self error: "so nicht!"]	
6 roommate := otherskier	

Aufgabe 17

24 Punkte

Schreiben Sie ein Smalltalk-Programm für folgendes Problem: Ein Versandunternehmen nimmt von seinen Kunden Pakete entgegen, die nach Übersee verschickt werden sollen. Die Pakete werden einzeln gewogen und dann nach Größe und Ziel zu größeren Gebinden (auch rekursiv) zusammengefaßt und in Container verladen. Entwerfen Sie ein kleines Smalltalk-Programmfragment, das das Gewicht des Inhalts eines Containers bestimmen kann, wobei der Container nur seine unmittelbaren Gebinde und jedes Gebinde nur seine unmittelbaren Bestandteile kennen soll. Versuchen Sie dabei, eine geeignete Klassenstruktur zu finden und alle notwendigen Fallunterscheidungen über dynamisches Binden zu treffen.

Lösung

• Klasse	• Behaelter
• benannte Instanzvariablen	• behaelter
• Instanzmethoden	•

```

7   gewicht
8   | summe |
9   summe := 0.
10  behaelter do: [ :b | summe := summe + b gewicht].
11  ^summe

12  hinzufuegen: einBehaelter
13  behaelter ifNil: [behaelter := OrderedCollection new].
14  behaelter add: einBehaelter

```

• Klasse	• Container
• Superklasse	• Behaelter

• Klasse	• Gebinde
• Superklasse	• Behaelter

• Klasse	• Paket
• Superklasse	• Behaelter
• benannte Instanzvariablen	• gewicht
• Instanzmethoden	•

```

15  gewicht: eingewicht
16  gewicht := eingewicht

17  hinzufuegen: einBehaelter
18  ^self shouldNotImplement

```