TENTAMEN

Programmeren 1

vakcode: 213500
datum: 10 juli 2004
tijd: 9:00-12:30 uur

Algemeen

- Bij dit tentamen mag gebruik worden gemaakt van het boek van Niño/Hosch, en van de handleiding van Programmeren 1.
- Dit tentamen bestaat uit 4 opgaven, waarvoor in het totaal 100 punten behaald kunnen worden. Het minimale aantal punten per opgave bedraagt 0.
- Bij de opdrachten in dit tentamen hoeven *geen* pre- en postcondities te worden gegeven, tenzij *expliciet* anders vermeld. Neem wel commentaar op waar dat nuttig is voor het begrijpen van uw oplossing.

Opgave 1 (15 punten)

Een database van personen slaat voor elke persson ook de *huwelijkse staat* op; mogelijke waarden zijn in ieder geval "gehuwd" en "ongehuwd". Bespreek tenminste vier verschillende datatypen die voor de representatie van dit gegeven gebruikt zouden kunnen worden, op de volgende manier:

- Geef het type en de representatie van de verschillende (mogelijke) waarden, waaronder in ieder geval de hierboven genoemde.
- Bedenk tenminste drie criteria waaraan uw representatie zou moeten voldoen, en beoordeel elk van uw representatiekeuzes volgens elk criterium, met uitleg (plm. 1 regel). Een voorbeeldcriterium is: *uitbreidbaarheid* (d.w.z., kunnen er nieuwe mogelijke waarden worden gedefinieerd).

Opgave 2 (25 punten)

In deze opgave gaat het om een `static` methode `index(List elementen, Object gezocht)`, die de index in `elementen` van `gezocht` oplevert, of `-1` als `gezocht` niet in `elementen` voorkomt. (Van de klasse `List` heeft u uitsluitend de methoden `Object get(int index)` en `int size()` nodeig¿)

a. *15 punten.* Geef een implementatie van de methode, met postconditie en een lusinvariant; maak aannemelijk dat de lusinvariant correct is en dat de postconditie gegarandeerd is.

b. *5 punten.* Stel dat de lijst n elementen bevat. Wat is het gemiddelde aantal stappen (= aantal keren dat de lus doorlopen wordt) die uw methode `index` nodig heeft om de index van een object te bepalen dat in de lijst aanwezig is? Kunt u een met kleiner aantal stappen toe als u weet dat de elementen in de lijst geordend zijn — bijvoorbeeld, als u weet dat de lijst uit geordende `String`s bestaat en `gezocht` ook een `String` is? Zo ja, hoe dan (in woorden omschreven), en wat is het aantal stappen? Zo nee, waarom niet?

c. *5 punten.* Kunt u van een willekeurig `Object` bepalen of het "kleiner is dan" een ander `Object`? Zo ja, wat betekent dit "kleiner dan" zijn? Zo nee, wat is er dan nodig (d.w.z., wat moet u van de `Object`en weten) om dit wél te kunnen bepalen?

Opgave 3 (50 punten)

U schrijft een programma dat, gebruik makend van beschikbare web services, de snelste verbinding tussen twee steden bepaalt. Alle web services zijn als Java-klassen beschikbaar die de hieronder gedefinieerde interface `Vervoer` implementeren.

```
public interface Vervoer {
    /**
     * Levert de benodigde tijd, in minuten, om van een
     * gegeven plaats naar een andere te komen. Levert -1 als
     * geen verbinding tussen de gegeven plaatsen gevonden wordt.
     * @param van de beoogde vertrekplaats
     * @param naar de beoogde bestemming
     */
    public int tijd(String van, String naar);
}
```

a. Schrijf een klasse `VervoerMatrix` die `Vervoer` implementeert op basis van twee intern bijgehouden instantievariabelen, met de volgende declaraties:

```
private List plaatsen;  // de plaatsen die dit vervoermiddel aandoet
private int[][] tijden; // de tijden op basis van plaatsindex
```

`plaatsen` is de lijst bekende plaatsnamen; `tijden` is een matrix waarin `tijden[i][j]` de tijd tussen `plaatsen.get(i)` en `plaatsen.get(j)` aangeeft. Een voorbeeld, gebaseerd op treintijden:

`plaatsen`

0	Hengelo
1	Deventer
2	Zutphen

`tijden`

	0	1	2
0	0	30	40
1	30	0	20
2	40	20	0

Hieruit blijkt dat het sneller is direct van Hengelo naar Zutphen te reizen dan via Deventer; en ook dat Boekelo geen treinstation heeft. In dit voorbeeld verwachten we de volgende antwoorden:

- `tijd("Hengelo","Zutphen")` levert `40` op;
- `tijd("Hengelo","Boekelo")` levert `-1` op;
- `tijd("Boekelo","Boekelo")` levert `0` op.

Gebruik in uw implementatie de methode `index` uit Opgave 2.

b. Schrijf een klasse `VervoerKeuze` die intern een `List` van `Vervoer`-objecten bijhoudt, en zelf de interface `Vervoer` implementeert, waarbij de implement van `tijd` de kortste benodigde tijd tussen `van` en `naar` oplevert, volgens de `Vervoer`-objecten in de intern bijgehouden lijst.

c. Schrijf een interface `BetaaldVervoer`, die `Vervoer` uitbreidt met een methode `double kosten(String van, String naar)` die de kosten van een reis tussen `van` en `naar` met dit vervoermiddel oplevert, of `-1` als met dit vervoer zo'n reis niet mogelijk is.

d. Schrijf een klasse `PerMinuurVervoer`, die `BetaaldVervoer` implementeert door twee instantievariabelen bij te houden:

```
private Vervoer vervoer;
private double prijsPerMinuut;
```

`PerMinuutVervoer` dient `tijd` te implementeren door de methode aan `vervoer` door te geven, en `kosten` door de benodigde tijd met de factor `prijsPerMinuut` te vermenigvuldigen. Daarnaast dient `PerMinuutVervoer` een constructor te krijgen waarin `vervoer` van een waarde wordt voorzien, alsmede methoden om `prijsPerMinuut` op te vragen en van waarde te veranderen.

e. Neem aan dat de volgende postcondities gespecificeerd zijn:

- Voor de methode `tijd` in `Vervoer`:

  ```
  @ensure result >= -1
  ```

- Voor de methode `kosten` in `BetaaldVervoer`:

  ```
  @ensure result == -1 als tijd(van,naar) == -1; anders result >= 0
  ```

Geef pre- en postcondities voor de constructor en methoden van `PerMinuutVervoer`, en leg uit waarom uw implementaties in `PerMinuutVervoer` de postcondities garanderen.

f. Geef een klassendiagram van de klassen en interfaces in deze opgave, incl. instantievariabelen maar zonder methoden of afhankelijkheden. Neem ook `List` en `Object` als klassen op.

Opgave 4 **(10 punten)**

Bij een matrix zoals in Opgave 3 gebruikt, waarin de getallen de "afstand" aangeven tussen twee dingen[1] hoort het altijd zo te zijn dat de afstand van x naar z hoogstens zo lang is als de som van de afstanden van x naar y en van y naar z.

a. *2 punten.* Geef een voorbeeld van een 3×3-matrix waarin deze voorwaarde niet vervuld is.

b. *8 punten.* Schrijf een methode `int[] correct(int[][] a)`, die test of `a` aan deze voorwaarde voldoet; zo ja, dan dient de methode `null` op te leveren, anders een `int`-array met lengte `3`, met daarin de waarden voor x, y, z waarvoor de voorwaarde niet geldt. Geef op basis van uw eigen voorbeeld aan wat de uitkomst van de methode zou moeten zijn.

[1] in het geval van Opgave 3 is de "afstand" de benodigde tijd en de "dingen" plaatsen