

Faculteit Wiskunde en Informatica

Tentamen Datastructuren

Vrije Universiteit

12 juni 1997

tijdsduur : 3 uur

Opgave 1.

- a) Indien in een public class een object geïmplementeerd wordt waarvan het de bedoeling is dat dit object in principe in ieder programma gebruikt kan worden (bijvoorbeeld een Datum-object), moet er gebruik gemaakt worden van information hiding en encapsulation (data abstractie).

Leg uit wat er precies onder information hiding en encapsulation verstaan wordt en hoe ze gerealiseerd worden.

Gebruik in je uitleg de begrippen interface, class, private velden, public velden, specificatie object, implementatie object, PRE-conditie en POST-conditie.

- b) Leg uit hoe het met behulp van inheritance mogelijk is om een array te declareren waarvan de elementen willekeurige objecten kunnen zijn.

Geef ook de declaratie van een variabele generiekeRij die in de declaratie geïnitialiseerd wordt als zo'n array met maximaal 10

- c) Als het de bedoeling is dat de elementen van een array totaal verschillende soorten objecten kunnen zijn, maar niet ieder willekeurig object, dan wordt dit niet met behulp van inheritance maar met behulp van interfaces gerealiseerd.

Leg uit en laat zien hoe je het zou kunnen aanpakken om te zorgen dat de classes

```
class X {          class Y {          class Z {  
....              ....              ....  
}                  }                  }
```

de enige drie soorten objecten definiëren die aan elementen van een array xyzRij, met maximaal 10 elementen, mogen worden toegekend.

Geef ook de declaratie van de variabele xyzRij.

- d) Wanneer is er sprake van overloading?
Wanneer is er sprake van polymorfisme (polymorphism)?
Wat is een abstract class?
Wat is het verschil tussen een abstract class en een interface?
Wat zijn superclasses en subclasses?

Opgave 2.

- a) Een queue is een eenvoudige standaard datastructuur voor de opslag van gegevens in een rij.

Geef een interface voor een queue van int-waarden met de standaard queue-methodes `enqueue()` en `dequeue()` en een methode `isEmpty()` die test of de queue wel of niet leeg is, een methode `init()` om de queue leeg te maken en een methode `first()` om de waarde van de eerste int in de queue te retourneren.

Geef bovenaan in de interface een abstracte beschrijving van een queue via "elementen", "structuur" en "domein". Vermeld bij iedere methode de PRE- en POST-conditie.

- b) Implementeer de interface uit onderdeel a) van deze opgave in een public class.

Realiseer de queue met behulp van een array dat initieel 16 elementen bevat en dat zich verdubbelt als er een element moet worden toegevoegd terwijl het vol is.

Maak een parameterloze constructor en een copy-constructor.

Maak hulpmethodes voor eventuele deelproblemen.

Opgave 3

Gegeven zijn de onderstaande classes.

```
class BoomKnoop {  
  
    int    data;  
  
    BoomKnoop left, right;  
  
    public BoomKnoop (int d) {  
        this(d, null, null);  
    }  
  
    public BoomKnoop (int d, BoomKnoop l, BoomKnoop r) {  
        data = d;  
        left = l;  
        right = r;  
    }  
  
}  
  
public class BinaireZoekboom {
```

```
private BoomKnoop wortel;  
  
....  
  
....  
  
}
```

- a) Maak een parameterloze constructor voor de class BinaireZoekboom die de lege boom aanmaakt.
- b) Wat is de hoogte (depth) van een knoop in een binaire zoekboom?

Maak een methode

```
public int hoogt()
```

in de class BinaireZoekboom die de hoogte van de wortel van de binaire zoekboom retourneert.

Implementeer de methode hoogt() door gebruik te maken van een recursieve hulpmethode.

- c) Beschrijf het algoritme om een knoop te verwijderen uit een binaire zoekboom. N.B. er wordt dus niet gevraagd om het algoritme helemaal te programmeren.

Voorzie de beschrijving van het algoritme van tekeningen van alledrie de gevallen en vermeld duidelijk hoe deze worden aangepakt.

♀

d) Wanneer is een binaire zoekboom een AVL-boom?

Maak een methode

```
public boolean AVLBoom ()
```

die true retourneert als de binaire zoekboom een AVL-boom is en false indien dit niet het geval is.

Maak hiervoor eerst een recursieve hulpmethode

```
private int hoogteAVLBoom (BoomKnoop k) throws Exception
```

die ieder knoop in de boom met wortel k precies 1 keer bezoekt, een exception geeft als de boom met die wortel geen AVL-boom is en de hoogte van die boom retourneert indien het wel een AVL-boom is.

TIPS: De absolute waarde van een int kan worden genomen met de static

methode abs uit de class Math. Bijvoorbeeld "Math.abs(-3) == 3".

Het maximum van twee int-waarden kan worden genomen met de static

methode max uit de class Math. Bijvoorbeeld "Math.max(1, 2) == 2".

Beoordelingstabel :

Opgave | a b c d e f g | Totaal

1 | 10 6 6 10 | 32

2 | 12 16 | 28

3 | 5 10 5 10 | 30

TOTAAL 90

Het eindcijfer E voor het tentamen volgt uit het puntentotaal T als volgt: $E = T/10 + 1$