

TENTAMEN
Imperatief Programmeren
14 mei 2003

■ Opgave 1 Winkelen

Een probleem waarmee eigenlijk elke Nederlander, maar zeker elke student mee zit is het zo efficiënt mogelijk besteden van het huishoudgeld. We stellen ons voor dat er in een bepaalde plaats een aantal supermarkten is die allemaal minstens een basispakket bestaande uit een aantal artikelen aanbieden. De prijzen van deze artikelen kunnen per winkel aanzienlijk verschillen.

Als je nu zeeën van tijd hebt, dan is het eenvoudig uit te maken in welke winkel je welk artikel moet kopen, nl. in de winkel waar dat artikel voor de laagste prijs wordt aangeboden. Heb je wat minder tijd (er zijn ook studenten die een deel van hun tijd aan studeren besteden!), dan zal het in het algemeen niet doenlijk zijn alle winkels te bezoeken.

deel 1

In het eerste deel van deze opgave gaan we er van uit dat je ten hoogste twee winkels bezoekt om het totale pakket aan goederen aan te schaffen.

De bedoeling is nu om een programma te maken dat aangeeft welke twee winkels je moet bezoeken om je boodschappen zo goedkoop mogelijk in huis te krijgen.

Veronderstellingen/beperkingen :

- Elke winkel biedt een basispakket bestaande uit hetzelfde aantal artikelen.
- Deze artikelen zijn genummerd (vanaf 0).
- Bij elke winkel is de prijs (in eurocenten) van elk artikel bekend.
- Van elk van deze artikelen moet er 1 exemplaar worden gekocht.

We ontwikkelen het programma in een aantal fasen.

- [4 pt] 1. Een winkel wordt gerepresenteerd door een object van de klasse `Winkel`. Beschrijf welke data-elementen deze klasse moet hebben, declareer deze elementen en schrijf een constructor die de nodige initialisaties uitvoert.
- [5 pt] 2. Implementeer in de klasse `Winkel` de methode die voldoet aan de volgende specificatie:

```
int minKosten (Winkel other) {
    (* RETURN het minimale bedrag dat je nodig heb om
    *      bij de winkels this en other het pakket
    *      aan artikelen te kopen.
    *)
    ..
} // minKosten
```

lees verder

We veronderstellen nu dat er een array van winkels beschikbaar is volgens de declaratie:

```
Winkel [] winkel;
```

en dat dit (niet lege) array is voorzien van data.

- [7 pt] 3. Schrijf een programma-fragment dat afdruckt welke twee winkels uit deze rij je moet bezoeken om zo goedkoop mogelijk uit te zijn.

deel 2.

Slechts twee winkels mogen bezoeken is misschien wel wat al te krap. We verruimen daarom de mogelijkheden: de gebruiker kan aangeven hoeveel winkels hij/zij (maximaal) wil bezoeken.

In deel 1 was het mogelijk om een keuze van twee winkels te representeren met twee variabelen. Nu het aantal te bezoeken winkels niet meer vantevoren vaststaat, kunnen we niet volstaan met een vast aantal variabelen die de nummers van te bezoeken winkels administreren. Daarom introduceren we het array

```
boolean [] bezoek = new boolean [winkel.length];
```

die bij elk winkelnummer aangeeft of de betreffende winkel wordt bezocht.

- [7 pt] 4. Schrijf een methode `minKosten` die bij de gegeven arrays `winkel` en `bezoek` het minimale bedrag oplevert waarvoor je het pakket aan artikelen kunt aanschaffen als je alleen de winkels bezoekt die in `bezoek` zijn aangegeven.

- [8 pt] 5. Implementeer de methode `minBedrag` die voldoet aan de specificatie

```
int minBedrag (int k, int n) {
    (* RETURN het minimale bedrag dat je moet besteden om
    *      het pakket aan artikelen te kopen waarbij de
    *      te bezoeken winkels die geadministreerd zijn in
    *      bezoek[k .. winkel.length)
    *      worden aangevuld met n winkels uit
    *      [0 .. k)
    *)
    ..
} // minBedrag
```



■ Opgave 2 Tjon

In het land Karian maakt iedereen al in zijn jeugd kennis met het nationale spel Tjon. In het land zijn dan ook diverse verenigingen waar dit spel in competitieverband wordt gespeeld.

De secretaris van een van deze clubs heeft de ledenadministratie geautomatiseerd. Omdat de overheid het beoefenen van het nationale spel zeer belangrijk vindt, krijgen de clubs zo veel subsidie, dat er geen contributie geheven hoeft te worden. De ledenadministratie stelt dan ook niet zo veel voor: van ieder lid wordt de naam, de totale jaarscore en een lijstje met de tien hoogste wedstrijdscoringen bijgehouden:

```
class Lid {
    String naam;
    int    totaal = 0;
    int [] highScore = {-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

    Lid (String nm) {
        naam = nm;
    } // constructor

    ..

} // Lid
```

Het array `highScore` is geordend, met de hoogste score bij index 0. Deze ordening moet bij alle operaties behouden blijven.

- [6 pt] 6. Geef deze klasse een methode `verwerkScore` die een behaalde wedstrijdscoring verwerkt.

Ga nu uit van het bestaan van een array

```
Lid [] lid;
```

en neem aan dat alle elementen van dit array zinvolle data bevatten. Verder is gegeven dat de lijst alfabetisch is geordend op naam.

- [7 pt] 7. Geef een methode `indexOf` die bij een string `str` de kleinste index oplevert van het element in het array `lid` waarvan de naam overeenkomt met `str`. Komt dit niet voor, dan is het resultaat -1.

Omdat de ledenlijst erg lang is, is hier een efficiënte oplossing vereist.

lees verder

Hoewel de overheid het nationale spel nog steeds een warm hart toedraagt, moet er vanwege de verslechterende economische toestand bezuinigd worden. Daarom zullen hier en daar in het land clubs moeten fuseren.

[6 pt] □ 8. Laat gegeven zijn

```
Lid [] lijst1;
```

```
Lid [] lijst2;
```

en neem aan dat beide lijsten alfabetisch geordend zijn op naam.

Schrijf een programmafragment dat de beide lijsten samenvoegt tot een nieuwe (alfabetisch geordende) lijst.

➤ einde

Maximaal aantal punten: 50