

Tentamen Imperatief Programmeren

8 maart 2000, 9-12 uur

- Het tentamen bestaat uit 4 opgaven, die alle vier even zwaar tellen.
- Schrijf op elk ingeleverd blad je naam, en op het eerste blad ook je collegekaartnummer en het aantal ingeleverde bladen (indien >1).
- Toon bij het inleveren je collegekaart.
- Als je een deel van de opgave niet kunt maken (bijvoorbeeld een van de methoden) probeer dan toch de rest te doen! Daarbij mag je niet-geschreven methoden gewoon aanroepen.
- In alle programma's mag je de *import*-regels weglaten.

Opgave 1

- Schrijf een methode *aantalDelers* die als resultaat oplevert hoeveel delers een als parameter meegegeven geheel getal heeft. Je mag zonder controle aannemen dat de parameter >0 is. Een deler is een geheel getal (groter dan 0) waardoor het getal zonder rest deelbaar is. Voorbeeld: 12 heeft 6 delers (1, 2, 3, 4, 6 en 12), 15 heeft 4 delers (1, 3, 5 en 15), 17 heeft 2 delers (1 en 17).
- Bekijk het volgende programmafragment:

```
int x, y;
for (y=0; y<10; y++)
    for (x=0; x<y; x++)
        gr.drawString("#", 20*x, 20*y);
```

Schrijf een programmafragment met dezelfde werking, maar zonder daarbij de for-opdracht te gebruiken, en zonder de operator ++ te gebruiken.
- In de header van een klasse kan het woord **implements** worden gebruikt. Wat staat er na dat woord? Welke verplichtingen geeft dat?
- Als een methode een parameter verwacht met het objecttype A, is het onder een bepaalde voorwaarde ook mogelijk om een object van type B als parameter mee te geven. Welke voorwaarde is dat? Geef ook een voorbeeld van een standaard-methode waarbij dit zinvol is.
- Beschrijf hoe je een object van de klasse *GridLayout* kunt gebruiken, en in welke situatie dat nodig is.

Opgave 2

Schrijf een statische methode *vaaksteInLangste* met de volgende specificaties.

De methode heeft als parameter de naam van een file.

Op elke regel van de file waarvan de naam als parameter is meegegeven staat precies één woord. De methode bepaalt welk van de 26 letters van het alfabet het vaakst voorkomt in de totale file. Hoofdletters en kleine letters moeten daarbij als dezelfde letter worden beschouwd. Andere symbolen dan de 26 letters van het alfabet mogen worden genegeerd.

Als resultaat van de methode moet worden opgeleverd hoe vaak deze letter voorkomt in het langste woord dat in de file werd aangetroffen.

In gevallen waarbij meerdere letters het vaakst voorkomen, of meerdere woorden het langste zijn, mag je er zelf een kiezen.

Als de file helemaal geen woorden bevat, moet als 0 worden opgeleverd. Als de file niet bestaat of niet leesbaar is, moet de waarde -1 worden opgeleverd.

Voorbeeld: de file zou de volgende vier woorden kunnen bevatten:

Appel
Peer
Aardbei
Banaan

De letter A komt het vaakst voor in deze file (6 keer). Het langste woord is Aardbei. De letter A komt daarin tweemaal voor, dus het resultaat is 2.

Zie Ommezijde

Opgave 3

Schrijf een applet met de volgende specificatie.

Boven in beeld is een drietal knoppen zichtbaar, met als opschrift respectievelijk “Schoon”, “Tekst” en “beweeg”. De afmetingen van het window zijn 500 breed bij 300 hoog (dat staat in de HTML-file, die je niet hoeft te schrijven).

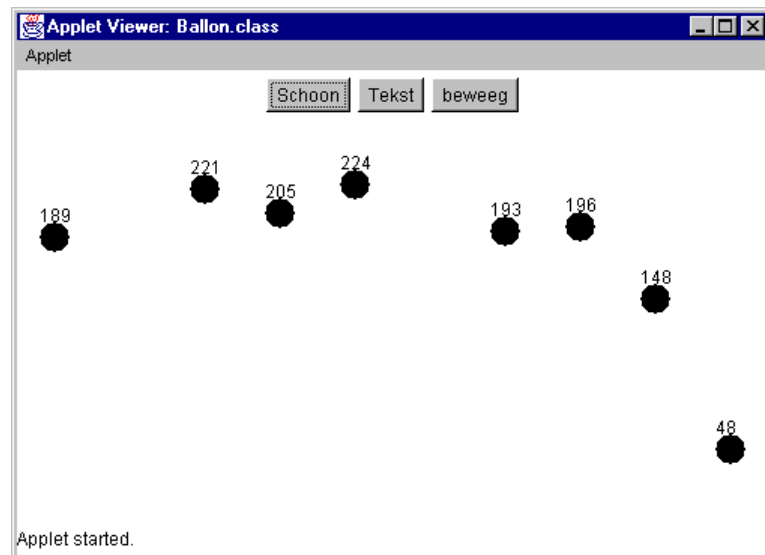
Het window is in tien denkbeeldige verticale stroken verdeeld. Als de gebruiker ergens in het window klikt, verschijnt er een opgevulde cirkel in de aangeklikte strook. Het middelpunt ligt op de hoogte van het aangeklikte punt, en horizontaal ligt de cirkel in het midden van de strook.

Klikt de gebruiker later nog eens in dezelfde strook, dan gebeurt er niets; de cirkel blijft dus waar hij stond.

Als de gebruiker op de knop “Tekst” drukt verschijnt bij elke cirkel als bijschrift de afstand (in beeldpunten) van het middelpunt van de cirkel tot de onderrand van het window. Als de gebruiker nogmaals op de knop “Tekst” drukt, verdwijnen de teksten weer; bij een derde druk op de knop verschijnen ze weer, enzovoorts.

Als de gebruiker op de knop “Schoon” drukt, verdwijnen alle cirkels, en kan de gebruiker weer beginnen met het aanklikken van nieuwe punten.

Als de gebruiker op de knop “beweeg” drukt, beginnen de cirkels met een snelheid van 10 beeldpunten per seconde naar onderen te kruipen. Eventueel aanwezige teksten lopen mee. Als een cirkel onderaan uit beeld loopt, verdwijnt hij; de strook wordt daarmee weer beschikbaar voor een nieuwe cirkel. Als de gebruiker nogmaals op “beweeg” drukt, stopt de beweging; bij een derde druk op de knop gaan de cirkels weer bewegen, enzovoorts.



Opgave 4

In deze opgave verplaatsen we ons in de schrijver van het package *java.lang*: je moet zelf (een deel van) de klassen *String* en *Integer* schrijven. Van de bestaande klassen *String*, *Integer* en *Double* mag je alleen de aangegeven twee methoden gebruiken. Wel mag je de operator `+` gebruiken om een string en een char samen te voegen (maar niet om twee strings samen te voegen).

- Neem aan dat de klasse *String* al voor een deel is geschreven: de objectvariabelen zijn gedeclareerd (maar je weet niet welke dat zijn), en de methoden *length* en *charAt* zijn al beschikbaar. Voeg nu zelf de volgende methoden toe, met dezelfde werking als de “echte” methoden:
 - `equals`
 - `substring` (de versie met twee int-parameters)
 - `indexOf`
 - `concat`
- Schrijf de methode *parseInt* uit de klasse *Integer*. Je mag zonder controle aannemen dat de parameter uit minstens één symbool bestaat, en dat alle symbolen cijfer-tekens zijn.

Einde