

Tentamen Imperatief Programmeren

22 december 1999, 14-17 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

- a. Schrijf een statische methode die als resultaat oplevert of een als parameter meegegeven geheel getal een priemgetal is. (Een priemgetal is een getal dat uitsluitend deelbaar is door 1 en zichzelf). Je mag er zonder controle van uitgaan dat de parameter groter dan 1 is.
- b. Wat voor verschil maakt het voor objecten van klasse A als in de header van deze klasse wordt aangegeven dat het een subklasse is van een klasse B ?
- c. In de body van een methode kan het woord **super** worden gebruikt. Wat wordt hiermee aangeduid? In welke situatie is dit nuttig om te gebruiken?
- d. Beschrijf hoe een constructor-methode afwijkt van een gewone methode voor wat betreft zijn naam, resultaattype, en wijze van aanroep.
- e. Beschrijf hoe je een object van de klasse *MediaTracker* kunt gebruiken, en in welke situatie dat nodig is.

Opgave 2

Schrijf een statische methode *grootste*, met de volgende specificaties.

De methode heeft twee parameters: een filenaam en een array. Zonder controle mag aangenomen worden dat de array 26 gehele getallen bevat. Deze getallen bepalen het "gewicht" van de 26 letters van het alfabet: op positie 0 staat het gewicht van de letter A, op positie 1 het gewicht van de letter B, enzovoorts.

Op elke regel van de file waarvan de naam als parameter is meegegeven staat precies één woord. De *waarde* van zo'n woord is het totaal van de gewichten van alle letters die er in het woord voorkomen. Hoofdletters en kleine letters hebben daarbij dezelfde gewicht. Andere symbolen dan de 26 letters van het alfabet, die mogelijk in het woord voorkomen, dragen niet bij aan het totaal.

De methode *grootste* moet als resultaat het woord uit de file opleveren met de grootste waarde. Als er meerdere van dat soort woorden zijn, mag er daarvan één worden gekozen.

Als de file helemaal geen woorden bevat, moet als resultaat een woord van nul letters worden opgeleverd. Als de file niet bestaat of niet leesbaar is, moet de waarde **null** worden opgeleverd.

Opgave 3

Schrijf een applet met de volgende specificaties.

Boven in beeld is een drietal knoppen zichtbaar, met als opschrift respectievelijk "schoon", "lijnen" en "beweeg". De hele rest van het window kan door de gebruiker overal worden aangeklikt.

Iedere keer als de gebruiker een punt van het window aanklikt, verschijnt gecentreerd op die plaats een vierkantje van 5x5 beeldpunten. Maximaal mogen honderd punten worden aangeklikt. Als de gebruiker toch meer punten probeert aan te klikken, gebeurt er niets.

Als de gebruiker op de knop "lijnen" drukt, wordt *elk* tot nu toe aangeklikt punt met een lijn verbonden met *elk* ander punt. Punten die daarna worden aangeklikt worden eveneens door lijnen verbonden met alle eerder aangeklikte punten.

Als de gebruiker nogmaals op de knop "lijnen" drukt, verdwijnen de lijnen weer; bij een derde druk op de knop verschijnen ze weer, enzovoorts.

Als de gebruiker op de knop "schoon" drukt, verdwijnen alle punten en lijnen, en kan de gebruiker weer beginnen met het aanklikken van nieuwe punten.

Als de gebruiker op de knop "beweeg" drukt, beginnen de punten met een snelheid van 10 beeldpunten per seconde naar rechts te kruipen. De verbindinglijnen kruipen mee. Als een punt rechts uit beeld verdwijnt, komt het links weer tevoorschijn (en klappen de bijbehorende lijnen dus ook om). Als de gebruiker nogmaals op "beweeg" drukt, stopt de beweging; bij een derde druk op de knop gaan de punten weer bewegen, enzovoorts.

Opgave 4

Definieer zelf de klasse *StringTokenizer*, met daarin

- een constructormethode, met twee strings als parameter
- een methode *nextToken* met een String als resultaat
- een methode *hasMoreTokens* met een boolean als resultaat

Je mag daarbij de bestaande klasse *StringTokenizer* niet gebruiken.

Wel mag je de klasse *String* en methodes daarvan gebruiken.

- De constructormethode heeft twee parameters: een string die gesplitst moet worden (bijvoorbeeld "aap+noot=mies") en een string met scheidings tekens (bijvoorbeeld "+=").
- De methode *nextToken* moet het eerste deel van de string opleveren dat geen scheidings tekens bevat (in het voorbeeld is dat "aap"). De volgende keer dat *nextToken* wordt aangeroepen, krijg je het volgende woord ("noot" in het voorbeeld), enzovoorts. In het voorbeeld is "mies" het derde woord, ondanks het feit dat daar geen scheidingstekens meer op volgt.
- De methode *hasMoreTokens* beantwoordt de vraag of er nog meer tokens beschikbaar zijn. Na twee aanroepen van *nextToken* in het voorbeeld is dat *true*, want "mies" is er nog.

Als er in de te splitsen string twee scheidingstekens naast elkaar staan, dan telt de lege string daartussen als apart token.

Einde.