

# Eerste deeltoets Concurrency

11 oktober 2013, 8.30 – 10.30, Educ- $\alpha$ .

Motiveer je antwoorden *kort!* Zet je mobiel uit. Stel geen vragen over deze toets; als je een vraag niet duidelijk vindt, schrijf dan op hoe je de vraag interpreteert en beantwoord de vraag zoals je hem begrijpt. Te halen 21pt, resultaat is totaal plus 1 gedeeld door 2.

- Safety of Liveness:** Zeg van elk van deze uitspraken of ze een *Safety* of een *Liveness* eigenschap beschrijven en waarom (in 1 zin).
  - Berichten worden afgeleverd in dezelfde volgorde als waarin ze zijn verstuurd.
  - Als een bericht aan een niet-bestaand adres wordt verzonden, wordt binnen 1500 ms een foutmelding getoond.
  - Na klikken op *LOAD* wordt het bestand in de edit-buffer geladen.
  - Een krant die in de zon ligt, wordt geel.
  - De waarde van *counter A* is strikt stijgend.
- Atomaire Queue:** (a) Geef de definitie van *atomaire* operaties.  
(b) Geef een voorbeeld van drie operaties op een Queue (inclusief timing) die wel *consistent* zijn maar niet *atomair*.
- Permutatie en Combinatie:** Geef de formule voor de berekening van  $P(n, k)$ . Hoeveel rijtjes van 13 speelkaarten uit een pak van 52 zijn er (geef een getal)?  
Geef de formule voor de berekening van  $C(n, k)$ . Hoeveel deelverzamelingen van 12 kaarten uit een set van 32 zijn er (geef een getal)?
- LockOne:** Hier staan de lock en unlock van de LockOne klasse (voor thread  $i$ ).

```
public void lock()                public void unlock()
{ flag[i] = true;                { flag[i] = false ; }
  while (flag[j]) {} } }
```

  - Aan welke drie eisen moet een lock implementatie voldoen?
  - Welke van deze eisen is/zijn voor LockOne niet voldaan? Waarom?
  - Als je de twee regels in `lock` verwisselt, is dan het probleem opgelost?
- Ballenvraag:** Uit een bak die 20 rode en 20 gele ballen bevat, pak je in het donker willekeurig acht ballen. Wat is de kans dat je hierbij *precies vijf* rode ballen pakt?
- Safe Integer:** Van een *Safe bit* kun je een *Regular bit* maken met de methode van *silent writes*.
  - Wanneer zal een `Write` operatie als *silent* uitgevoerd worden?
  - Is de methode van *silent writes* ook toe te passen om van een *Safe Integer* een *regular Integer* te maken?
- De kaartenkijker:** De Kaartenkijker trekt steeds een kaart uit een pak van 52 *met teruglegging*, dus steeds wordt de kaart teruggelegd en het pak geschud.
  - Hoeveel kaarten moet de Kaartenkijker *verwacht* trekken totdat hij een *rode zeven* ziet?
  - Hoeveel kaarten moet de Kaartenkijker *verwacht* trekken totdat hij *alle* 52 kaarten een keer gezien heeft?

(c) Hoeveel kaarten moet de Kaartenkijker *verwacht* trekken totdat hij 51 verschillende kaarten gezien heeft?

8. **Electie en Consensus:** Een electie-object gebruik je als een bepaalde taak door precies één thread moet worden gedaan, maar je kunt niet van tevoren bepalen welke dat is. Het atomaire object heeft slechts één methode `boolean elect()` en garandeert: *Als `elect()` één of meerdere keren wordt aangeroepen, wordt precies één keer `true` teruggegeven en verder `false`.*

(a) Toon aan, dat als aanroepen na elkaar plaatsvinden, het altijd *de eerste* aanroep is die `true` oplevert.

(b) Laat zien, dat je met het electie-object consensus kunt implementeren voor twee threads.

(c) Kan het ook voor drie threads?

(d) Wat is het consensus-getal van het electie-object?