

Naam:.....

Belangrijk: Schrijf je antwoorden kort en bondig in de daartoe voorziene velden. Indien nodig gebruik je de achterkant van het blad (met duidelijke vermelding van de vraag). Het gebruik van een eenvoudig rekenmachine is toegestaan. Andere hulpmiddelen (smartphone, laptop, cursusnota's, ...) zijn niet toegestaan tijdens het examen. Elke theorie-vraag staat op 2 punten (totaal op 24). De twee oefeningen staan op 4 respectievelijk 2 punten. Het geheel staat op 30 punten.

1. Introduction [... / 2]

Wat is het verschil tussen Iteratief en Incrementeel ontwikkelen?

Iteratief:

gecontroleerd herwerken van een deel van het systeem om verbeteringen aan te brengen [1]

of: we get things wrong before we get them right [1]

.....

Incrementeel:

voortgang in kleine stappen om in een vroeg stadium tastbare resultaten te behalen [1]

of: always have running version [1]

.....

2. Requirements [... / 2]

Wat is een "Minimum Viable Product" (MVP)?

Een minimum viable product (MVP) is een eerste versie van een product met net genoeg functionaliteit om bruikbaar te zijn door eerste gebruikers om feedback te verkrijgen [1,5]

Het is een eerste mijlpaal voor start-ups [0,5]

3. Software Architecture [... / 2]

Definieer de begrippen "sensitivity point" en "tradeoff point" uit de ATAM terminologie?

Sensitivity point:

- Sensitivity point: A sensitivity point is a property of one or more components (and/or component relationships) that is critical for achieving a particular quality attribute response. [1]

.....

Tradeoff point:

- Tradeoff point: A trade-off point involves two (or more) conflicting sensitivity points.[1]

.....

4. Project Management [... / 2]

Waarom heeft het vervangen van een persoon in een ontwikkelingsproces een negatieve impact op de productiviteit? (2 redenen)

Naam:.....

1)

.....

2)

.....

- First, the person that is leaving will have to do additional work to hand over to the successor. Moreover, this person is typically less motivated, must take remaining holidays, [1]

- Second, the replacement will have many questions in the beginning regarding the working of the system. These questions are often asked to co-workers, the newcomer bothers their colleagues. [1]

5. Design by Contract [... / 2]

Wat is behavioural subtyping? Kun je een (tegen)voorbeeld geven?

.....

- [0,5] Dit is een interpretatie van het liskov substitutie principe

- [1] Definitie:

* If S is a subtype of T, then objects of type T may be replaced with objects of type S (i.e., objects of type S may substitute objects of type T) without altering any of the desirable properties of that program (correctness, task performed, etc.).

* Each instance of a superclass may be replacing by instances of a subclass without altering the correct behaviour.

- [0,5] als ze een voorbeeld geven (vb. rechthoek en vierkant)

6. Testing [... / 2]

Hoeveel tests moet je schrijven om van MC/DC coverage te kunnen spreken?

Hoeveel testen voor multiple condition coverage?

MC/DC Coverage:

n + 1 test cases (for a decision with n conditions) [1]

Multiple condition coverage:

2 to the power of n test cases (for a decision with n conditions) [1]

.....

.....

7. Formal Specifications [... / 2]

Geef een wiskundige definitie voor de zwakste pre-conditie (i.e. “weakest

precondition”) voor het Hoare triple {P} S {Q}

.....

∀P' such that {P'} S {Q}, P' ⇒ P,

.....

.....

.....

Naam:.....

8. Domain Models [... / 2]

Geef twee voordelen en twee nadelen van de “clone and own” werkwijze.

Voordelen:

1)
.....

2)
.....

Saves time and reduces costs

Provides independence

Readily available

Nadelen:

1)
.....

2)
.....

Propagating changes

Adapting the clone is difficult

Repetitive tasks are common (=> Bug fixing)

Which variant to clone from?

9. Software Quality [... / 2]

Wat zijn de drie zaken die moeten worden opgenomen in het kwaliteitsplan? Leg uit.

1)
.....

2)
.....

3)
.....

1) gewenste produkt-kwaliteiten aangeven en bespreken hoe deze beoordeeld worden;

2) toe te passen organisatorische standaarden aangeven

3) het kwaliteits assessment proces definiëren

10. Software Metrics [... / 2]

Schrap 2 van de laatste 4 woorden in de volgende zin. Leg kort uit, gebaseerd op Putnam’s model.

Als je vroeger wilt eindigen (geplande tijd verkleinen), dan moet je de effort veel / weinig — vergroten / verkleinen

Waarom:

Naam:.....

.....
.....
.....
.....

veel / ~~weinig~~ — vergroten / ~~verkleinen~~

In de formule voor productiviteit is de geplande tijd de dominante factor; tot de 4^{de} macht. Als je die een klein beetje wil verkleinen moet je dus veel harder werken.

11. Refactoring

[... / 2]

Geef de pre-conditie(s) voor een “rename method” refactoring.

.....
.....
.....

Een methode met dezelfde naam en parameterlijst mag niet bestaan in dezelfde klasse, noch in één van zijn subklassen en superklassen.

12. Conclusion

[... / 2]

Als je het “No Silver Bullet” artikel hebt gelezen:

Waarom is programma verificatie geen silver bullet?

.....
.....
.....
.....

Programma verificatie doet iets aan de accidentele complexiteit: nagaan of een programma voldoet aan zijn specificatie. Maar de specificatie kan fouten bevatten en bovendien nog incompleet zijn. De essentiële complexiteit (= hetgeen inherent moeilijk is aan het bouwen van software) is net het opstellen van een complete en consistente specificatie.

Als je het “Killer Robot” artikel hebt gelezen:

Werd code reviewing toegepast als deel van het kwaliteitsbewakings proces?

Waarom wel/Waarom niet?

.....
.....
.....
.....
.....

*0,5: neen te weinig tijd, taken geminimaliseerd
0.5: code werd gereviewed*

Naam:.....

1: ja efficiënter algoritme gevonden via code reviewing

2: ja maar niet goed, programmeurs stonden niet open voor code reviewing

Naam:.....

13. Oefeningen – Testing

[... / 4]

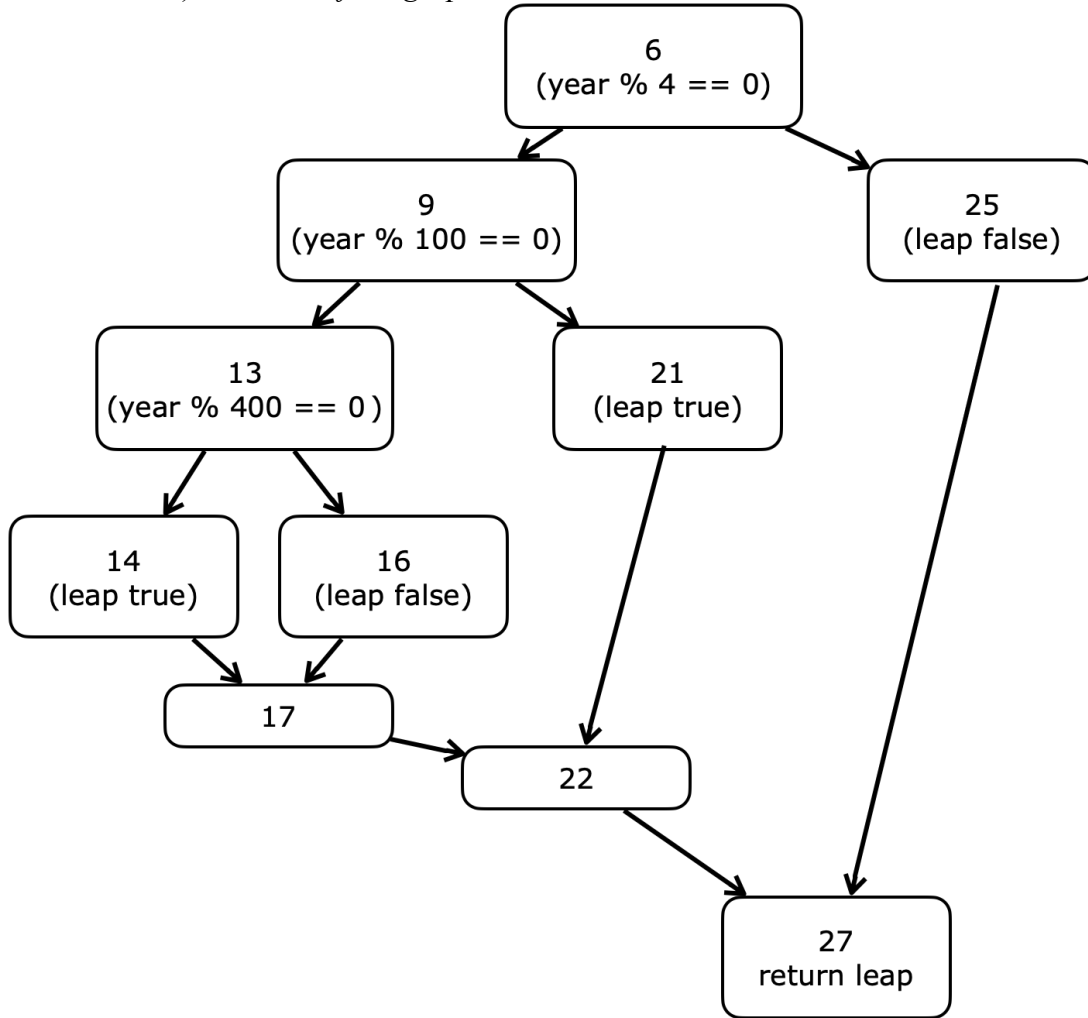
Beschouw de onderstaande functie die antwoord of een gegeven jaartal (een integer) een schrikkeljaar is of niet.

Een schrikkeljaar is deelbaar door 4, maar niet als het deelbaar is door honderd en toch weer wel als het deelbaar is door vierhonderd.

```
1: public static boolean isLeapYear(int year) {
2:
3:     boolean leap = false;
4:
5:     // if the year is divided by 4
6:     if (year % 4 == 0) {
7:
8:         // if the year is century
9:         if (year % 100 == 0) {
10:
11:             // if year is divided by 400
12:             // then it is a leap year
13:             if (year % 400 == 0)
14:                 leap = true;
15:             else
16:                 leap = false;
17:         }
18:
19:         // if the year is not century
20:         else
21:             leap = true;
22:     }
23:
24:     else
25:         leap = false;
26:
27:     return leap;
28: }
```

Naam:.....

a) Teken de *flow graph* voor bovenstaande functie.



Naam:.....

b) Bereken de *cyclomatische complexiteit* en geef kort aan hoe je hiertoe gekomen bent.

.....

#10 nodes #12 edges => $12-10+2 = 4$

#binary conditions = 3 => $3+ 1 = 4$

#regions = 4

.....

c) Bepaal een volledige verzameling *onafhankelijke paden*. (Nummer ze).

.....

(a) [6, 9, 13, 14, 17, 22, 27]

(b) [6, 9, 13, 16, 17, 22, 27]

(c) [6, 9, 21, 22, 27]

(d) [6, 25, 27]

.....

.....

.....

.....

d) Bepaal voor elk pad een test case (input en verwachte output) om dit onafhankelijk pad te kunnen uitvoeren. Gebruik dezelfde nummering als in de vorige oefening om te verwijzen naar een pad.

(a) [6, 9, 13, 14, 17, 22, 27]

a. 2000 => leap true

(b) [6, 9, 13, 16, 17, 22, 27]

a. 2100 => leap false

(c) [6, 9, 21, 22, 27]

a. 1984 => leap true

(d) [6, 25, 27]

a. 1987 => leap false]

.....

.....

Naam:.....

14. Oefeningen – Statechart

[... / 2]

In deze oefening ga je het gedrag van een broodrooster met noodstop modelleren aan de hand van een StateChart.

Beschouw daarbij de volgende specificaties.

1. De sneetjes brood worden *manueel* in en uit de rooster geplaatst; dat moet niet worden opgenomen in de statechart.
2. Er is een sensor voorzien (brood_aanwezig) die aangeeft of er al dan niet een sneetje brood correct in de rooster zit.
3. De broodrooster heeft één aan-uit schakelaar. Als deze aangaat én er een sneetje brood in de rooster zit begint de elektrische hitteregelaar voor 30 seconden te werken. Als die tijdspanne is verlopen wordt de hitteregelaar uitgezet.
4. Er is ook een noodstop voorzien; als die wordt ingedrukt wordt de hitteregelaar onmiddellijk uitgezet.

Zet bovenstaande om in een statechart model. Voorzie de nodige states, (timed) transitie's, guards en zo nodig orthogonale regionen. Zorg ervoor dat de start en eindstaat correct worden weergegeven.

Naam:.....

Teken hier je statechart.

[0,5] Juiste notatie voor start en stop state

[0,5] Juist gebruik van guards

[1] alle nodige transities

Broodrooster

