

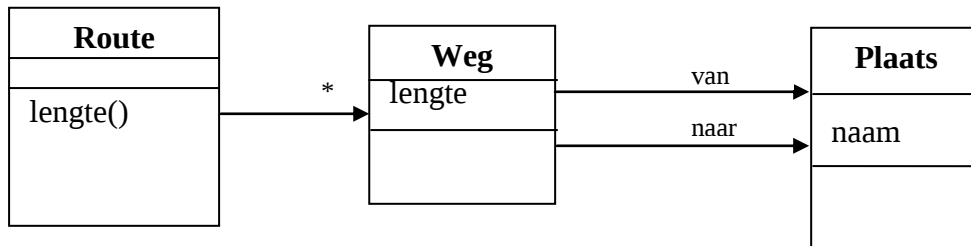
Tentamen in2705 Software Engineering

Voorbeeld
(bijna tweemaal te groot)

U mag meenemen naar dit tentamen: Lethbridge, afdrukken PPT slides, afdrukken handouts.

1. De TU wil een nieuw systeem ontwikkelen voor het beheren van studieresultaten van studenten.
 - a. (4 pt) Welke gebruikers voor dit systeem kunt u onderscheiden?
 - b. (4 pt) Welke kwaliteiten zijn voor dit systeem van belang, en waarom?
 - c. (4 pt) Op welke wijze(n) zou u de requirements voor dit systeem kunnen verzamelen?
 - d. (4 pt) Welke architectuur zou u voor een dergelijk systeem kiezen? Motiveer uw keuze.
2. Euler (<http://mathschallenge.net>) is een systeem ter ondersteuning van een internet wedstrijd in programmeren. Ingeschreven deelnemers kunnen een opgave kiezen en de oplossing invoeren. De oplossing wordt dan gecontroleerd. De oplossing is altijd een geheel getal. De waarde van een probleem is een positief getal in de range 1..20. De score van een deelnemer is de som van de waarden van alle problemen die hij heeft opgelost. Naarmate meer deelnemers een probleem oplossen neemt de waarde van dat probleem af. De score van een deelnemer kan dus ook afnemen, zonder dat hij iets doet.
 - a. (10 pt) Geef een Use-Case beschrijving van het kiezen, oplossen en laten controleren van en opgave.
 - b. (10 pt) Geef een volledig klassendiagram voor het bovenbeschreven systeem. Voorzie associaties van zoveel mogelijk informatie, en onderscheid waar nodig ook aggregatie en compositie. Klassen uit de gebruikersinterface mag u weglaten.
 - c. (10 pt) Geef (gebaseerd op het klassendiagram van onderdeel b) een sequence diagram van het insturen van een oplossing.
3. (10 pt) Het cruise-control systeem van een auto heeft een interface bestaande uit een schakelaar en twee toetsen, A en B. Met behulp van de schakelaar kan het systeem worden ingeschakeld en op elk moment ook weer worden uitgeschakeld. Als het systeem is ingeschakeld is het nog niet meteen actief. Pas als toets A wordt ingedrukt wordt het systeem actief, het zorgt er dan voor dat de auto de snelheid behoudt die hij had op het moment dat toets A werd ingedrukt. In deze toestand kan de snelheid worden gewijzigd door een van beide toetsen in te drukken: indrukken van toets B verhoogt de snelheid iets, indrukken van toets A verlaagt de snelheid. Maakt de bestuurder in deze toestand gebruik van het rem- of koppelingspedaal, dan gaat het systeem op non-actief. Het kan dan op twee manieren weer geactiveerd worden: met toets B wordt de laatst ingestelde snelheid de richtsnelheid, met toets A wordt de huidige snelheid de richtsnelheid. Geef een state diagram voor de toestanden en overgangen van het cruisecontrole systeem.

4. Geef aan de hand van onderstaand klassendiagram OCL-specificaties:
- (4 pt) Begin- en eindpunt van een weg moeten verschillend zijn. Geef een OCL-beschrijving van deze eigenschap.
 - (8 pt) In de klasse Route komt de methode lengte voor. De lengte van een route is de som van de lengten van de wegen in de route. Geef een OCL-specificatie van deze methode.



5. (Zie bijlage B)
- (5 pt) Welke vorm(en) van koppeling vindt u tussen de klasse BinExpressie en de klasse Expressie?
 - (5 pt) Welke vorm(en) van koppeling vindt u tussen de klassen LiteralExpressie en de klasse BinExpressie?
6. (Zie bijlage B)
- (6 pt) Welke design principes worden geschonden door de methode toString in de klasse Expressie?
 - (6 pt) Welke design principes worden geschonden in de klasse BinExpressie?
7. (10 pt) In bijlage A ziet u een implementatie van een Queue met behulp van de LinkedList uit het package java.util. Welk Design Pattern herkent u hier? Geef aan welke rollen uit het template van het pattern door welke elementen uit de implementatie worden gespeeld.
8. (10 pt)
- In een recent project werden 200000 regels C-code geschreven. Wat zijn de nominale kosten in mensmaanden volgens het COCOMO-model, uitgaande van een embedded project? Hoe lang zal het duren? Welke teamgrootte zou u voor dit project adviseren?
 - Een mijlpaal in een project wordt bereikt na 30 weken, vijf weken voor de geplande 35 weken. Daartoe zijn wel meer mensen ingezet: deze mijlpaal kostte 30 mensmaanden, in plaats van de geplande 24 MM. Laat zien hoe deze gegevens in een earned value chart worden weergegeven.
9. (20 pt) In de methode aantalDelers wordt het aantal delers van het getal n geteld. We beginnen met $d = 2$, maar omdat 1 ook tot de delers van n wordt gerekend, krijgt *aantal* startwaarde 1.

```

public int aantalDelers(int n){
    int d = 2;
    int aantal = 1;
    while (d <= n){
        if (n % d == 0)
            aantal = aantal + 1;
        d = d + 1;
    }
    return aantal;
}

```

Geef de flow graaf van de methode aantalDelers. Hoeveel paden zijn er door deze graaf? Beperk u tot paden die de herhalingslus niet meer dan twee maal doorlopen. Geef zo mogelijk voor elk pad een waarde van n die tot dit pad aanleiding geeft.

10. (15 pt) Gegeven is de methode aantalWortels:

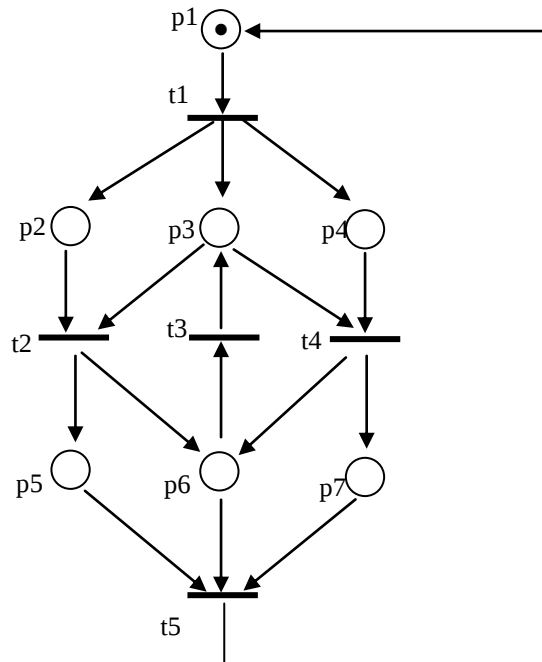
```

public int aantalWortels (int a, int b, int c)

```

Deze methode bepaalt het aantal wortels van de vergelijking $ax^2 + bx + c = 0$.

De methode aantalWortels moet worden getest. Verdeel de mogelijke invoer voor deze methode in equivalentieklassen, en geef een testgeval voor elke klasse. Geef ook testgevallen voor 'randwaarden'.



11. (25 punten) Gegeven is bovenstaand petrinet. Cirkels zijn plaatsen, horizontale streepjes zijn transities en stippen in cirkels zijn tokens.

- Geef vanuit de gegeven marking een firing-sequence die willekeurig vaak herhaald kan worden.
- Geef vanuit de gegeven marking een firing-sequence die tot deadlock leidt.

- c. We geven het net een nieuwe marking, met tokens op p1 en p6. Laat zien dat ook vanuit deze marking deadlock kan ontstaan.
- d. Ga uit van de marking onder c). We noteren deze met de vector: (1,0,0,0,0,1). Welke markings zijn vanuit deze marking bereikbaar?
- e. Is het net met de marking onder c)
 - i. safe?
 - ii. bounded?
 - iii. conservative?

Bijlage A

```
import java.util.*;

interface Queue{
    void enqueue(int[] item);
    int[] dequeue();
    boolean isEmpty();
}

class ListQueue implements Queue{
    private LinkedList list = new LinkedList();

    public void enqueue(int[] item){
        list.addLast(item);
    }

    public int[] dequeue(){
        int [] antwoord = (int [])list.getFirst();
        list.removeFirst();
        return antwoord;
    }

    public boolean isEmpty(){
        return list.isEmpty();
    }
}
```

Bijlage B

```
abstract class Expressie{
    public abstract int eval(int x);
    //public abstract Expressie add(Expressie exp);

    public String toString(){
        if (this instanceof LiteralExpressie)
            return "" + eval(0);
        else
            return "X";
    }
}
```

```

class XExpressie extends Expressie{

    public int eval (int x){
        return x;
    }
}

class LiteralExpressie extends Expressie{
    private int waarde;

    public LiteralExpressie(int k){
        waarde = k;
    }

    public int eval (int x){
        return waarde;
    }
}

class BinExpressie extends Expressie{
    static final char
        PLUS    = '+',
        MIN     = '-',
        MAAL    = '*',
        DEEL    = '/';

    private Expressie left, right;
    private char op;

    public BinExpressie(Expressie l, char op, Expressie r){
        left = l;
        this.op = op;
        right = r;
    }

    public int eval(int x){
        int leftvalue = left.eval(x);
        int rightvalue = right.eval(x);
        switch (op){
            case PLUS : return leftvalue + rightvalue;
            case MIN  : return leftvalue - rightvalue;
            case MAAL : return leftvalue * rightvalue;
            case DEEL : return leftvalue / rightvalue;
        }
        return 0;
    }

    public String toString(){
        return '(' + " " + left.toString() + " " + op +
            " " + right.toString() + " " + ')';
    }
}

```