Tentamen in2705 Software Engineering

28 juni 2007 14:00-17:00

U mag meenemen naar het tentamen: Lethbridge, afdrukken PPT slides, afdrukken **handouts**

Opgaven 1.. 6 hebben betrekking op onderstaande casus.
Lees de casus en de vragen zorgvuldig voordat u begint.

De HIO de Hekker in Juinen heeft meer dan duizend eerstejaars studenten. Dat levert vooral bij het handmatig nakijken van de programmeer-practica problemen op. De school wil nu een geautomatiseerd systeem voor het nakijken van programmeer-opgaven. De opgaven worden voortaan (tot vreugde van de studenten) uitsluitend op werking beoordeeld, en niet meer op programmeerstijl. Alle student-assistenten worden ontslagen. De programma's die worden beoordeeld lezen uit een invoerfile, en schrijven naar een uitvoerfile. Elke inzending wordt gecompileerd, en vervolgens losgelaten op 10 testcases. Elke testcase die correct wordt uitgevoerd levert een punt op. Een inzending is voldoende als er zes punten worden behaald. De student krijgt een mededeling over het behaalde aantal punten.
Elke opgave heeft een deadline. Voor elke werkdag dat de deadline wordt overschreden wordt een punt afgetrokken. De opgaven moeten in volgorde worden ingeleverd: als de voorafgaande niet is goedgekeurd wordt deze niet nagekeken. (Dat geldt natuurlijk niet voor de eerste opgave).

1. Geef gemotiveerd aan welke technieken u hier zou gebruiken om de requirements te verkrijgen.

2. Geef een Use-Case beschrijving van het inzenden van een programma.

3. Geef een klassendiagram van dit systeem. U mag de klassen uit de User Interface weglaten. Houd er rekening mee dat het systeem voor diverse practica gebruikt kan worden. Geef vooral informatie mbt. de relaties tussen de klassen.

4. Geef een sequence diagram voor het insturen en controleren van een programma. Zorg dat uw diagram consistent is met het klassediagram van opgave 3.

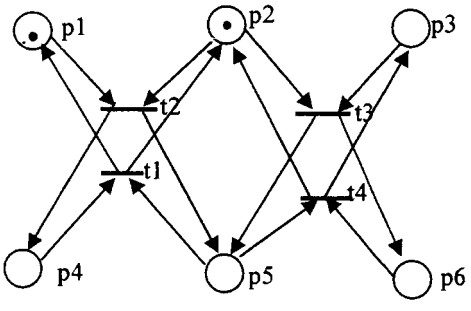
5. Als het goed is komt er in uw klassediagram (opgave 3) een klasse voor die de relatie tussen een student en een opgave representeert. Wat zijn de relevante toestanden voor een object van deze klasse? Geef een State Diagram voor objecten van deze klasse.

6. Welk(e) architectuur patroon (patronen) zou u voor bovengenoemd systeem kiezen? Geef een deployment-diagram van het systeem.

7. In de bijlage vindt u de klasse CompanyMember en een tweetal subklassen.
 a) Welk Design Pattern herkent u hier?
 b) Geef aan welke rollen uit het template van het pattern door welke elementen uit de implementatie worden gespeeld.

8.
 a) Er moet een project van 500 functiepunten gerealiseerd worden, met C++ als programmeertaal. Bereken de nominale kosten in mensmaanden, uitgaande van het COCOMO-model. Ga uit van een ***semidetached*** project.
 b) Een van de mijlpalen in een zeer voorspoedig project wordt na 21 weken bereikt, in plaats van de geplande 26 weken. Het bereiken van deze mijlpaal kostte niet de geplande 30 mensmaanden, maar slechts 24 MM. Laat zien hoe deze gegevens in een earned value chart zichtbaar worden.

9.



Figuur 1

Gegeven is het petrinet van figuur 1. Het is een standaard place-transition net. Cirkels zijn plaatsen, horizontale streepjes zijn transities. De stippen in de cirkels zijn tokens.

a) Geef vanuit de gegeven marking een firing-sequences die oneindig vaak herhaald kan worden.
b) Welke transities worden (bij de gegeven marking) nooit enabled?

Bijlage

```
import java.util.*;

abstract class CompanyMember{
      private String name;

      protected CompanyMember(String nm){
            name = nm;
      }

      public String toString(){
            return name;
      }

      public boolean equals(Object obj){
            return name.equals(((CompanyMember)obj).name);
      }



class Employee extends CompanyMember{
      private int salary;

      public Employee(String nm, int salary){
            super( nm );
            this.salary = salary;
      }



class Department extends CompanyMember{
      private Vector members;

      public Department(String nm){
            super(nm);
            members = new Vector();
      }

      public void includeMember(CompanyMember cm){
            members.addElement(cm);
      }

      public void removeMember(CompanyMember cm){
            members.removeElement(cm);
      }

}
```