TENTAMEN/EXAM

Kurs/course: Programvaruteknik / Software Engineering (5DV087)

Ansvarig lärare/responsible teacher: Jürgen Börstler

Datum/date: 2010-08-28
Tid/time: 09.00–13.00 (4 timmar/hours)

Namn/name: …………………………………………………………………….

Personnummer: …………………………………………………………………

E-mail: …………………………………………………………………………

Kod/code: _____

OBS!

Detta blad kommer att avskiljas innan läraren får skrivningen för rättning. **Ovanstående kod måste därför finnas på samtliga svarsblad** när du lämnar skrivningen till skrivvakten. **Memorera ditt kodnummer** eftersom det även kommer att användas som referens när resultaten meddelas.

Please note!

This sheet of paper will be detached before marking. **The code above must therefore be noted on all answer sheets** when you turn in your exam. **Memorize your code**, since it will be used when results are announced.

Till skrivningsbevakaren:

Avskilj detta försättsblad och stoppa i kuvert som skickas till **Yvonne Löwstedt**, Datavetenskap

Umeå University, SE-901 87 Umeå, Sweden
Department of Computing Science/Jürgen Börstler
Phone: +46 90 786 67 35, Fax: +46 90 786 61 26
E-mail: jubo@cs.umu.se

Umeå University, SE-901 87 Umeå, Sweden
Department of Computing Science/Jürgen Börstler
Phone: +46 90 786 67 35, Fax: +46 90 786 61 26
E-mail: jubo@cs.umu.se

EXAM 5DV087, VT10

Date 2010-08-28

TENTAMEN/EXAM

Kurs/course: Programvaruteknik / Software Engineering (5DV087)

Ansvarig lärare/responsible teacher: Jürgen Börstler

Datum/date: 2010-08-28
Tid/time: 09.00–13.00 (4 timmar/hours)

Kod/code: _____

Instructions:

- You can answer in English or in Swedish

- **Write your code and the question number in the upper right of every page**

- Motivate your answers and explain all assumptions

- Please do the following to speed up correction time:

    o Write only on one side of the paper

    o Start a new page for every question

    o Order your papers by question number

    o Mark all questions you have answered

    o Leave a margin for comments

Thanks and good luck, lycka till, viel Glück!

Exercise number	X	Points
1		
2		
3		
4		
5		
6		
7		
8		
TOTAL (40)		

Please mark exercises you have
worked on/solved in column X.

This page must be returned together with your answers.

Question 1 (4p)

What is an architectural style? Name and describe at least two architectural styles together with their respective advantages and disadvantages (you get credits for at most two).

Question 2 (1+1+2p)

(a) What is meant by the fundamental design principle of *coupling and cohesion*?

(b) Explain why a high degree of coupling in a software design can make maintenance very difficult.

(a) How could you measure coupling and cohesion? Propose at least one measure for coupling and one measure for cohesion and explain how these measures could be used to determine coupling and cohesion.

Question 3 (1+2+4p)

In the lecture, we have talked about design heuristics and design patterns.

(a) What are the similarities of and differences between design heuristics and design patterns?

(b) Give examples of a few design heuristics and describe their rationales (goals) briefly (you get credits for at most four).

(c) Name and describe some design patterns *in detail* (you get credits for at most two).

Question 4 (3+1+2p)

In the lecture, we have talked about quality factors for requirements (i.e. properties of "good" requirements and "good" requirements documents).

(a) Name and describe/explain some of these quality factors (you get credits for at most six).

(b) Define/describe the term *non-functional requirements*.

(c) Assume you are developing the requirements document for a web-based system for the booking cinema tickets. Suggest some *concrete* (i.e. specific, not generic) non-functional requirements of reasonable high quality (you get credits for at most four).

Question 5 (4p)

Model a state machine diagram for a microwave oven. Make sure the time can be changed without stopping the oven. Make also sure that the oven always can be opened safely during operation.

Question 6 (3p)

What are formal inspections? Explain what characterizes formal inspections and how they are carried out.

Umeå University, SE-901 87 Umeå, Sweden   EXAM 5DV087, VT10
Department of Computing Science/Jürgen Börstler
Phone: +46 90 786 67 35, Fax: +46 90 786 61 26   Date 2010-08-28
E-mail: jubo@cs.umu.se

Question 7 (3+2p)

Assume the following UML class diagram describing the relationships between cars and persons:
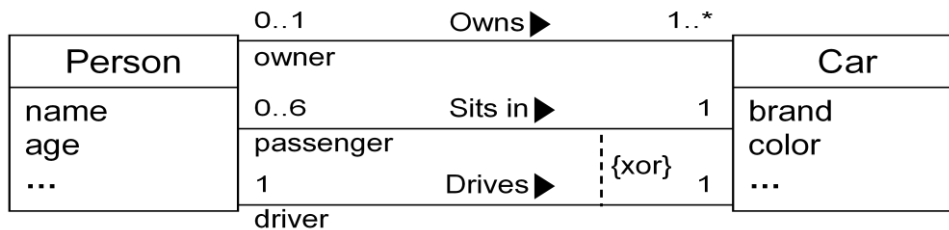
(a) Which of the following statements about particular instances can or cannot be correct with respect to the diagram above?

    i.   Jonny and Mary own a red Volvo together.

    ii.   John doesn't own a car.

    iii.   There is always exactly one driver in a car.

    iv.   This silver Saab must belong to somebody.

    v.   Owners are not allowed to drive a car.

    vi.   Passengers sit in exactly one car.

    vii.   Once Mary drove her Volvo and her Ferrari at the same time.

    viii.   I can take 8 passengers in my Opel.

    ix.   You are not allowed to drive alone.

    x.   You can either be the driver or the passenger of the same car but not both.

(b) How would you model the following "problems"?

    i.   The driver must have a driving license.

    ii.   Drivers can only drive their own cars.

Question 8 (1+1+5p)

(a) What is the main purpose of testing?

(b) What type(s) of testing would you argue you are doing in TDD?

(c) Develop test cases for reaching branch coverage for the following example:

```
function gcd( x, y: integer): integer;
(* Precondition: x,y > 0 *)
begin
      while x <> y do begin
            if x > y then
                  x := x-y
            else
                  y := y-x
      end;
      gcd := x
end;
```