

# INF5510 - Distribuerte Objekter

## Muntlig eksamen Juni 2013

Eric Jul  
Institutt for informatikk  
Universitetet i Oslo

15. mai 2013

Ved eksamen trekker eksaminanden et av spørsmålene nedenfor, når eksamen starter. Eksamenstiden er ca 30 min, hvorav den første delen skal være en fremleggelse av eksaminanden. Det er tillat å medbringe et a4-ark for hvert spørsmål. - Det er tillat å medbringe alt materiale fra pensumlisten. Det er tillat å bruke eksempler, figurer og annet materiale fra pensum under eksamen.

At the exam, each will draw one of the questions listed below when the exam starts. Each exam will take about 30 minutes. Expect to give a short presentation initially. You may bring ONE sheet of A4 sized paper for each question. You may bring any and all material mentioned in the PENSUM (the curriculum listed below). You may use any example, figure or other material from the PENSUM during the exam.

### Spørsmål

1. Gjør rede for Emeralds sprogmekanismer som understøttet mobility - herunder call-by-move.
2. Gjør rede for Emeralds måte at lave nye objekter på herunder object constructors, immutable objects og hvorfor og hvordan man laver klasser i Emerald.
3. Gjør rede for Emeralds typesystem - spesielt conformance, herunder hvorfor det er viktig i et distribueret system. Forklar pizza/junk eksemplet fra SP&E artikkelen. Forklar NIL, NOONE og ANY og hvordan det passer ind i typesystemet. Forklar view-as og restrict-to, gjerne via eksempel.
4. Gjør rede for remote invocations i Emerald og for hvordan de er implementeret i det oprindelige Emerald. Forklar samspillet med

mobility, call-by-move, call-by-visit og forklar break-even for at flytte et parameterobjekt med ifm et kald.

5. Gør rede for hvordan Emeralds comprehensive, robust, distributed garbage collector virker.
6. Gør rede for immutability og hvorfor det er vigtigt i et distribueret system.
7. Gør rede for hvordan object and process mobility er implementeret i det oprindelige Emerald.
8. Gør rede for Emeralds attach begreb, herunder motivation for det, hvordan det bruges og hvordan det er implementeret i det oprindelige Emerald.

## Questions in English

1. Explain Emerald's language mechanisms that support mobility - including call-by-move.
2. Explain how to create new objects in Emerald including the concept of object constructors, immutable objects and how and why to make classes in Emerald.
3. Explain Emerald's type system - specifically conformance and why it is important in a distributed system. Explain the pizza/junk example from the SP&E article. Explain NIL, NOONE and ANY and how they fit into the type system. Explain view-as and restrict-to; use examples, if possible.
4. Explain remote invocations in Emerald and how they are implemented in the original Emerald. Explain the interaction between the concepts of mobility, call-by-move, call-by-visit and explain break-even when moving a parameter object in connection with an invocation.
5. Explain how Emerald's comprehensive, robust, distributed garbage collector works.
6. Explain immutability and why it is important in an distributed system.
7. Explain how object and process mobility is implemented in the original Emerald.
8. Explain Emerald's concept of attachment, including the motivation for it, how it is used and how it is implemented in the original Emerald.

## **Pensum/Curriculum**

- Erics Ph.D. afhandling
- Niels Chr. Juuls Ph.D. afhandling
- Norm Hutchinsons Ph.D. afhandling
- 1992 ISMM artiklen om Distributed Garbage Collection
- 1991 SP&E artiklen om Emerald
- Emerald Language Report
- IEEE Software Engineering artiklen
- TOCS Emerald artiklen
- 2007 HOPL Emerald artikel