

Föreläsning 10: Ännu mer om att skriva egna klasser.

Christian.Soderberg@cs.lth.se

20 februari 2013



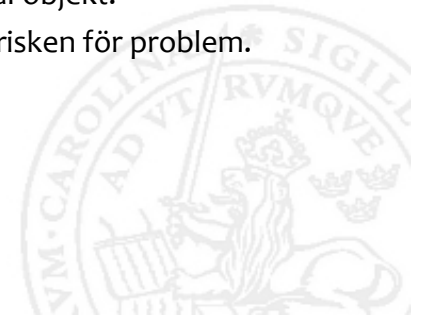
Inlämningsuppgift 3

Vanliga problem:

- ▶ Man får inte lämna ut den 'riktiga' listan med vänner!
- ▶ Lokal variabel i konstruktor.
- ▶ Registret ingen lista (det *innehåller* en lista).
- ▶ Jämförelser av strängar.
- ▶ Lokala variabler i switch-sats.



- ▶ Att ett referens-attribut är *private*-deklarerat betyder inte att man är helt säker – om vi släpper ut referensen så kan någon manipulera det objekt som attributet refererar till.
- ▶ Ibland måste vi skapa kopior på objekt för att vi inte skall riskera att någon ändrar i våra original-objekt.
- ▶ Objekt som är *oföränderliga* minskar risken för problem.



Mer om det objekt vi utför våra metoder på

- ▶ En klass beskriver ett objekt.
- ▶ När vi gör ett metod-anrop hamnar vi oss 'inne' i det objekt som vi gör anropet på.
- ▶ Vi kan använda namnet *this* för att referera till det objekt som vi befinner oss 'inuti'.



Överlagring och signatur

- ▶ Vi får i en klass ha flera metoder med samma namn, så länge de anropas på olika sätt – i kallar det för *överlagring*.
- ▶ *Signaturen* för en metod anger hur den anropas, den beskrivs av:
 - ▶ metodens namn, och
 - ▶ typerna på dess parametrar.
- ▶ Signaturerna på `distanceTo`-metoderna i klassen `Point` är alltså:
 - ▶ `distanceTo(Point)`
 - ▶ `distanceTo(double, double)`
- ▶ Alla metoder i en klass måste ha unika signaturer.

Överlagring av konstruktörer

- ▶ Även konstruktörer kan överlagras.
- ▶ I en överlagrad konstruktor kan vi anropa andra konstruktörer i samma klass, men måste göra det *innan vi gör något annat*.
- ▶ Vi använder det reserverade ordet `this` för att anropa andra konstruktörer.

Mer om inkapsling – tips

- ▶ Lämna inte utan vidare ut referenser till skyddade attribut.
- ▶ Överväg att använda oföränderliga objekt.