INF5110

Eksamensoppgave fra 2005

a)

Anta ta vi har et språk med klasser og subklasser. Alle metoder er virtuelle, slik at de kan redefineres i subklasser.
Gitt følgende klassedefinisjoner:

```
class A {
  int i;
  void P {... AP ...};
  void Q {... AQ ...};
}

class B extends A {
  int j;
  void Q {... BQ ...};
  void R {... BR ...};
}

class C1 extends B {
  void P {... C1P ...};
  void S {... C1S ...};
}

class C2 extends B {
  int k;
  void R {... C2R ...};
  void T {... C2T ...};
}
```

Vis hvordan objekter av klassene C1 og C2 vil være strukturert (layout) og tegn virtuell-tabellen for hvert av objektene. Bruk navnene i metodekroppene til å angi hvilken definisjon som gjelder for hvert objekt.

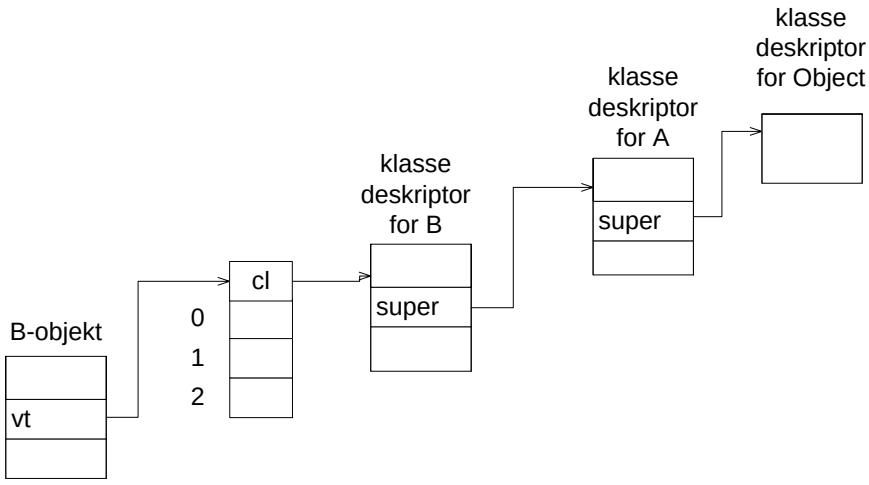
b)

Vi finner nå på å innføre i språket muligheten for å spesifisere en metode til å være **final**. Det skal bety at den ikke lenger er virtuell, dvs at den ikke kan redefineres i subklasser.
Anta at vi i klassen B spesifiserer metoden Q til å være final.
Må vi da endre på virtuell-tabellen for B-objekter? Begrund svaret.

c)

Vi innfører nå operatoren ’instanceof’: Det boolske uttrykket ’<refExpr> instanceof <class>’ er True hvis objektet som <refExpr> peker på har en klasse som er klassen <class> eller en subklasse av klassen <class>, ellers False.

For å kunne implementere denne operatoren utvider vi virtuell-tabellen med en peker til klasse-deskriptoren, som det er en av for hver klasse i programmet. Klassedeskriptoren har da en variabel ’super’, som peker til klasse-deskriptoren for superklassen. Klasser uten eksplisitt superklasse har den spesielle klasse Object som super. Eksemplet under viser dette for et objekt av klassen B:

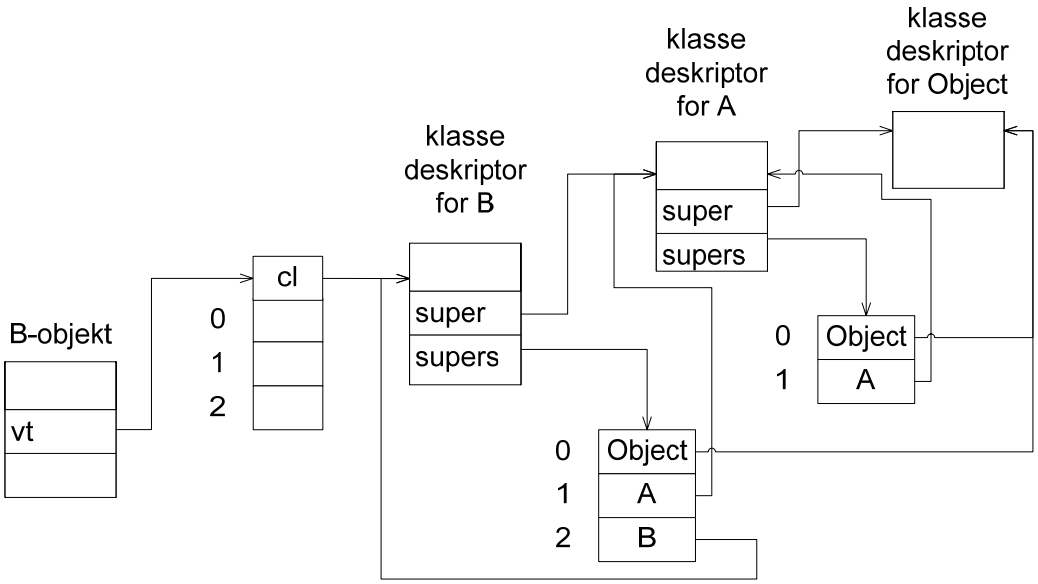


Skisser algoritmen som beregner verdien av ’<refExpr> instanceof <class>’.

d)

For å effektivisere testen på ’instanceof’ innfører vi (inspirert av display/kontekst vektor for nestede blokker) at en klassedeskriptor har en tabell ’supers’, som inneholder superklassene for klassen samt klassen selv. Denne tabellen har som indeks klassens ’subklassenivå’ startende med 0 for Object, 1 for rotklassen i et subklassehierarki, 2 for neste nivå, osv. I vårt eksempel har klassen A subklassenivå 1, B har 2, C1 og C2 har begge 3.

For klassene A og B ser klassedeskriptorene med supers-tabellene slik ut:



Forklar hvordan denne tabellen kan effektivisere instanceof-testen.

Eventuelt:

Til illustrere denne forklaringen kan du bruke følgende:

Vi innfører nå ytterligere to klasser:

```
class C11 extends C1 { ...}
class C21 extends C2 {...}
```

Lag supers-tabellene for klassedeskriptorene for C11 og C21, og vis hvordan følgende tester gjøres:

```
rC11 = new C11;
rC11 instanceof C1   (1)
rC11 instanceof C2   (2)
```