Explicaţie - Overriding

Petculescu Mihai-Silviu

Explicați rezultatele afișate prin executarea următorului program C++.

Rezultat

```
Start
B::v()
A::s()
B::w()
B::v()
A::s()
B::w()
A::v()
A::s()
A::w()
```

Programul propune două clase, clasa `A` superclasă şi clasa `B` subclasă, care suprascrie metoda `virtual void v()` moştenită de la `A`.

La execuţia programului creăm doi pointeri, unul `a` legat de superclasă, iar celălalt `b` legat de subclasă. Următoarea linie asociem pointerului `b` o instanţă a sublcasei `B`, la care, va pointa şi variabila `a` prin linia `a=b`.

Astfel la execuţia `a->v()` se va executa inplementarea subclasei `B` asupra metodei `v()`, rezultând afişarea B::v() ; A::s() ; B::w(), întrucât metoda `s()` este moşţenită de la superclasa `A`.

La următorul apel al funcţiei `v()`, după linia `a=(A*)b`, va păstra acelaşi comportament ca şi in precedenţă, întrucât forţărea la supratip păstrează implementarea metodelor subtipului, la metodele comune celor două entităţi. Metodele specifice doar subclasei se pierd, dar în cazul nostru acestea sunt inexistente (de precizat ca variabila `a`, pointând la sublcasă, poate execută, de asemenea, doar metodele comune celor două entităţi, deci, putem deduce ca liniile `a=b` şi `a=(A*)b` au acelaşi efect).

Ultima linie `((A)(*b)).v()` fortează instanţa clasei `B` să apeleze metodele, conform implementării din superclasa `A`, astfel afişându-se A::v() ; A::s() ; A::w().