

Explicație - Late binding vs. Early binding

Petculescu Mihai-Silviu

Explicație - Late binding vs. Early binding

Petculescu Mihai-Silviu

Ce se afișează prin executarea programului? Explicați mecanismul de legare a metodei la implementare pentru fiecare linie afișată.

Ce se afișează prin executarea programului? Explicați mecanismul de legare a metodei la implementare pentru fiecare linie afișată.

Despre legarea statică / early binding

Legarea statică se referă la evenimentele care au loc la momentul compilării. În principal, `early binding` se execută atunci când toate informațiile legate de apelul funcțiilor sunt cunoscute la momentul compilării. Datorită acestei particularități, aceste tipuri de funcții sunt caracterizate printr-o viteză de apelare mai mare decât în cazul `late binding`.

Despre legarea dinamică / late binding

Legarea dinamică se execută în momentul rurării programului, aspect implementat în C++ prin intermediul funcțiilor virtuale. Aceste funcții determină baza de cod ce va fi executată în funcție de tipul de date al pointerului de la care pornește apelul. Marele avantaj oferit de acestea ar fi flexibilitatea oferită în scrierea codului, care poate răspunde dinamic diferitelor cerințe apărute pe durata rurării. Legat de viteză, astăzi diferențele sunt suficient de mici pentru a trece ca insesizabile în marea majoritate a cazurilor cerute.

Rezultat

```
A::st()
A::vrt()
A::stafis()
A::st()
A::vrt()
B::vrtafis()
B::st()
A::vrt()
```

Clasa `A` implementează 4 metode, două statice `void st(); void stafis();` și două dinamice `void vrt(); void vrtafis();` prin declararea acestora cu keyword-ul `virtual`. Clasa `B` este subclasă a clasei `A` și alege să suprascrie 3 dintre cele 4 metode ale superclasei sale, cele 2 statice și funcția virtuală `void vrtafis()`.

La executarea programului se crează un pointer `p` de tipul superclasei `A`, care este făcut să poarte către o instanță a subclasei `B`.

La apelarea liniei `p->st()` se va executa funcția `st()` specifică clasei `A` întrucât aceasta a fost legată la compilare static. Se afișează `A::st()`.

Analog și pentru linia `p->stafis()` unde sunt luate în considerare doar implementările funcțiilor de tipul pointerului `p` (adică al superclasei). Se afișează `A::stafis()` `A::st()` `A::vrt()`.

Linia `p->vrt()` este un caz mai special, căci cu toate ca se apelează o funcție virtuală, aceasta nu are implementare în clasa `B` și deci, în consecință, se alege implementarea moștenită de la superclasă. Se afișează `A::vrt()`.

Ultima linie `p->vrtafis()` prezintă un caz clasic de `late binding` întrucât vor fi apelate în executare, în principal, doar interpretările din clasa `B` (subclasă) și în cazul în care lipsesc, la fel ca și în cazul anterior, se apelează interpretarea moștenită de la superclasă. Se afișează `B::vrtafis()` `B::st()` `A::vrt()`.