Explicaţie - Fragile Base Class Problem

Petculescu Mihai-Silviu

1. Executaţi programul şi explicaţi rezultatele.

Programul porneşte de la o clasă de bază `FragileBase` ce conţine o variabilă privată `counter`, două funcţii de manipulare a acesteia `inc_by_1()` şi `inc_by_2()` care incrementează valoarea variabilei `counter` cu 1 respectiv 2 unităţi, fără însă a se apela una pe cealaltă, şi o funcţie `display_counter()` care afişează valoarea variabilei.

Clasa `Derived` extinde clasa `FragileBase`, suprascriind funcţia `inc_by_2()` pentru a utiliza în proces funcţia moştenită `inc_by_1()`.

La apelarea funcţiei `main()` din clasa `Main` creăm un obiect de tipul `FragileBase` pe care îl instaţiem, pe rând, ca implementare a clasei `FragileBase` respectiv `Derived`. Pentru fiecare instanţa rurăm acelaşi set de instrucţiuni, apelând funcţiile de incrementare, apoi afişând valoarea variabilei `counter` specifică instanţei respective.

Rezultat

```
Operatii asupra unui obiect din clasa de baza
counter=3
Operatii asupra unui obiect din clasa derivata
counter=3
```

2. Executaţi aplicaţia schimbând prima versiune a clasei FragileBase cu versiunea nouă. Explicaţi rezltatul şi căutaţi explicaţia.

Rezultat

```
Exception in thread "main" java.lang.StackOverflowError
    at FragileBase.inc_by_1(FragileBase.java:27)
    at Derived.inc_by_2(Derived.java:5)
    at FragileBase.inc_by_1(FragileBase.java:27)
    at Derived.inc_by_2(Derived.java:5)
    (...)
```

Noua versiune a clasei `FragileBase` modifică functia `inc_by_1()` fără a-i schimba efectul. Astfel, în loc de a modifica direct contorul, ca în varianta originală, s-a optat pentru apelarea funcţiei `inc_by_2()` şi ajustarea rezultatului prin scaderea contorului cu o unitate. În final noua versiune a clasei va avea exact acelaşi efect ca şi în cazul celei precedente, lucru susţinut si de executarea funcţiei `main()` numai cu iniţializarea `o=new FragileBase();`

```
Operatii asupra unui obiect din clasa de baza
counter=3
```

În schimb, problema apare la iniţializarea variabilei cu ajutorul clasei derivate, `Derived`. Aceasta suprascrie metoda `inc_by_2()` pentru a apela funcţia `inc_by_1()` de două ori, dar din acest motiv, întrucât noua versiune a clasei `FragileBase` are metoda `inc_by_1()` gândită să apeleze la rândul ei metoda 2, se realizează o lupă infinită şi astfel programul se opreşte.

Problema poartă numele de "Fragile Base Class Problem" şi este una destul de comuna în domeniul moştenirii, care se aplica oricărui limbaj care permite această funcţionalitate. În principiu, clasa de baza, cea din care moşteneşţi, poartă numele de "fragile" deoarece orice modificare a acesteia poate conduce la rezultate neaşteptate în clasele care o moştenesc.

Există metode de a preveni acest comportament, dar nimic care să asigure evitare deplină a problemei. Una dintre soluţii, aplicabile în java, ar fi marcarea tuturor claselor care nu sunt concepute a fi moştenite ca şi `final`. Pentru restul claselor, recomandarea ar fi ca acestea să fie proiectate asemeni unui API, ascunzând toate detaliile de implementare, cu metode definite strict şi o documentaţie care sa explice, în detalii, comportamentul normal al acesteia.