

Java és .NET generics

Mind a Java 5.0 mind a .NET 2.0 támogatja a generikus szkítést, bár eltér módon. A Java a "code sharing" technikát használja, vagyis az alaptípus kerül használatra és a fordító oldja meg a típusvédelmet. .NET esetén a típus szkítése önálló típusként kerül a VM elé. A *code sharing* egy érdekes mellékhatást hoz létre, nézzük az alábbi kód futását:

```
List<Integer> list1 = new ArrayList<Integer>();
list1.add(1);
list1.add(2);
List list2 = list1;
list2.add("A");
for (int i : list1)
{
    System.out.println(i);
}
```

A lényeg ott van, hogy *list2* már nem tartalmaz szkítést a típusra vonatkozóan, s egy szimpla referencia lesz csak, amelyhez azonban hozzá tudunk adni egy *String* típusú értéket, amelyre aztán a *list1* példányon alapuló ciklus jól kiakad:

```
java.lang.ClassCastException: java.lang.String cannot be cast to java.lang.Integer
```

A .NET futásidej specializációja jobb teljesítményt is jelent, mivel nincs szükség classcast mveletre.