

Efficient hashCode() implementation

Ask Question

I often auto-generate an class's `hashCode()` method using IntelliJ IDEA and typically the method takes the form:

```
result = 31 * result + ...
```

My question is what is the purpose of multiplying by 31? I know this is a prime number but why pick 31 specifically? Also, if implementing a `hashCode()` for a particularly small / large dataset would people approach this problem differently?

[java](#) [algorithm](#) [data-structures](#) [hashmap](#) [hashcode](#)

asked Jul 2 '09 at 14:00



[Adamski](#)
44k 11 88 140

1 Answer

Multiplying by 31 is fast because the JIT can convert it to a shift left by 5 bits and a subtract:

```
x * 31 == (x << 5) - x
```

Without any particular extra information, I'd stick to this approach. It's reasonably fast and likely to end up with reasonably well-distributed hash codes, and it's also easy to get right :)

The size of the dataset doesn't really matter, but if you have particular extra information about the values you'll be work with (e.g. "it's always even") then you *may* be able to design a better hash function. I'd wait until it's an actual problem first though :)

answered Jul 2 '09 at 14:01



[Jon Skeet](#)



1035k 639 7654
8250

2 Then why not 7? That's a shift by 3 and a subtract. And it's a prime – [oxbow_lakes](#) Jul 2 '09 at 14:07

Thanks Jon. If this is the reason it's strange that IDEA doesn't just put $(x \ll 5) - x$ in the generated code. Is the JIT likely to actually spot this optimisation? – [Adamski](#) Jul 2 '09 at 14:08

6 7 allows strings that differ only in two adjacent characters to often end up with the same hash code. In fact pretty much any processor from the last decade or two should be able to manage a multiply by an eight bit number (so long as it is in a register) in a cycle. – [Tom Hawtin - tackline](#) Jul 2 '09 at 14:16

2 Last time I checked 31 was prime, too. – [Bombe](#) Jul 2 '09 at 15:19

1 [@dma_k](#): I'm afraid I don't know the details of it... only that it's meant to work well. (I thought Effective Java suggested 31 actually... maybe it's the second edition which does so?) – [Jon Skeet](#) Oct 5 '10 at 9:53
