

Re: what's it worth to write a short program for polynomial multiplication?

Re: what's it worth to write a short program for polynomial multiplication?

Source: <http://sci.tech-archive.net/Archive/sci.math.symbolic/2008-06/msg00037.html>

- *From:* Christopher Creutzig <christopher@xxxxxxxxxxxxx>
 - *Date:* Sat, 07 Jun 2008 14:05:29 +0200
-

rjf wrote:

Why should not the object-oriented choice of operator use the types of all the operands?

Good question. Most OO programming languages work that way. CLOS, the Common Lisp Object System is the one exception I am aware of, but perhaps there are more.

In C++, you can overload functions (including operators) on all arguments, sure:

```
template <typename coeffT>
SparsePoly<coeffT> operator*(SparsePoly<coeffT> &a const,
SparsePoly<coeffT> &b const);
```

```
template <typename coeffT>
SparsePoly<coeffT> operator*(DensePoly<coeffT> &a const,
SparsePoly<coeffT> &b const);
```

```
template <typename coeffT>
SparsePoly<coeffT> operator*(SparsePoly<coeffT> &a const,
DensePoly<coeffT> &b const);
```

```
template <typename coeffT>
DensePoly<coeffT> operator*(DensePoly<coeffT> &a const,
DensePoly<coeffT> &b const);
```

This is also possible for types only determined at runtime, but requires more code. The idiom is called a double dispatch pattern. Oh, and I should point out that the choice of return types above is of course not a good idea in general. Using the double dispatch pattern and determining whether to return a sparse or dense polynomial based on the actual data found is likely to be much better. But I haven't done any tests on this yet.

Re: what's it worth to write a short program for polynomial multiplication?

Re: what's it worth to write a short program for polynomial multiplication?

seit wann sind Vertragsinhalte für NewsGroup-Frager relevant?

Sie sind lebensnotwendig um sie sofort auf überraschende Inhalte abzuklopfen oder sonst in Frage zu stellen.
(Kurt Gunter und Konrad Wilhelm in dsrm)

.