

nanort::real3< vertex
_t >::operator*

nanort::real3< vertex
_t >::operator+

nanort::real3< vertex
_t >::operator-

nanort::real3< vertex
_t >::operator/

nanort::real3::real3

```
graph LR; A[nanort::real3< vertex_t >::operator*] --> D[nanort::real3::real3]; B[nanort::real3< vertex_t >::operator+] --> D; C[nanort::real3< vertex_t >::operator-] --> D; E[nanort::real3< vertex_t >::operator/] --> D;
```

The diagram illustrates the relationship between four overloaded operators and the `nanort::real3::real3` constructor. On the left, four white rectangular boxes are stacked vertically, each containing a C++ operator overload signature for `nanort::real3< vertex_t >`. Arrows from each of these boxes point to a single gray rectangular box on the right, which represents the `nanort::real3::real3` constructor. This visualizes that all four operators (multiplication, addition, subtraction, and division) ultimately rely on the same underlying `real3` object construction.