README

June 3, 2024

1 Strong typing and International System of Units

We generally pretend that we pratice object-oriented programming. However, when we have to choose the type of a new object, we still focus on its content rather than its behaviour, **especially when we deal with scalar numbers**. We choose `float` or `double`, and that's it. For an integer content, we choose `int`, rarely `short`, sometimes `long` if we suspect very big numbers, and that's it. Since **we ignore the meaning** of what we describe, we can **add meters with seconds**, or **confuse a table index with an object identifier**, without any complain from the compiler…

C++, especially nowadays, contains helpful functionnalities for the construction of **meaningful types**, for instance for the physical quantities : 1. Give units to scalar values 2. Handle multiples with user-defined literals 3. Combine them all 3. Centralize constants 4. BUT there are unsolved issues…

1.1 Main ressources and inspirations

Blogs * Strong typedefs, par Jonathan Muller * Strong types, par Jonathan Boccara * Opaque typedef, par Kyle Markley, 2016

Wikipedia * International System of Units

Propositions to standardization committee * Opaque Typedefs, P0109 * A C++ Approach to Physical Units, P1935