子曰、
知之者不如好之者、
好之者不如楽之者

僕のコードが不評なワケ

わんくま同盟

επιστημη

episteme@cppll.jp

Microsoft MVP for VC++ 2004-

僕のおしごと

- いわゆる「アプリケーション」は
  - めったに書かない
  - たまに火消し orz
- 目新しい開発環境/フレームワーク/ライブラリなどの
  - 調査・評価
  - チュートリアル・サンプル作り

「動かすコード」じゃなく「読ませるコード」が主。

「読ませるコード」ばっか書いてるから

- たまーに「動かすコード」を書くと…

不評!

- ライブラリの実装なんかがコレ。
  読んでもらうのが目的じゃないからねー

どこが不評かっちゅーと…

わからん

- 通常のコードにはお目にかからぬ記法
- なんでこれで動いちゃうの?
- コメント少ねー

たとえば…だ。

```
int main() {
  const int N = 1000;
  vector<int> p;
  primes.push_back(2);
  for ( int i = 3; i < N; i += 2 ) {
    if ( find_if(p.begin(), p.end(),
                 not1(bind1st(modulus<int>(),i)))
           == primes.end() ) {
      primes.push_back(i);
    }
  }
  copy(primes.begin(),primes.end(),
       ostream_iterator<int>(cout," "));
}
```

επιスタイル(1) ── 横に長く縦に短い

- TAB : 2-space
- 一行にみっちり詰め込む
  - ↑ 変数を介在させない
  - ↑ デバッグしんどい

スクリーンに表示される情報量を
できる限り多くしたい!

επιスタイル(2) ── 命名規則

- 省略型を嫌う
- スコープの広いものほど長い名前
- テンポラリ/使い捨て変数には短い名前
- で、変数/関数のスコープ/可視域は極力狭く

大事なモノにはエラソーな名前!

επιスタイル(3) ── コメント

- キホン「書かない」
- コメント不要のコードを善とする
  - 「何をするか」と「どうやってるか」は
    適切な名前付け と 適切な構文 とで
    あらかた表現できるっしょ。
- 書いておきたいコメントはコードで表現できないもの
  - 「なぜ、こうしたのか」
  - 「なぜ、そうしなかったのか」

ライブラリのジレンマ

- 汎用ライブラリの場合、使途を限定できない。
- 「堅さ」と「柔らかさ」を同時に求められる。

→ なので:

MEAN and LEAN が肝要

Interface と Generics をうまく使え!

※ オカズは拡張メソッドも悪くない