Conclusions

Conclusions

- **We hope you'll draw benefits from the design solutions we have presented.**
  - When facing a new problem, one may be inspired by documented, solved problems.
  - Immediate recognize the trade-offs for each solution.
  - Common vocabulary helps programmers exchange design ideas, as well as maintain each other's code.

Conclusion (cont.)

InterBit
Training & Consulting Ltd.

- **Try and isolate the minimum factor that changes.**

  - Example: strategy, bridge, observer ...
  - Benefits: avoiding code duplication; allowing others to later extend and configure your code.
  - Beware of: assumptions that are too narrow (not flexible) or too wide (hard to code; the common denominator problem).

Conclusions (cont.)

InterBit
Training & Consulting Ltd.

- **You may sometime consider using composition instead of inheritance.**
  - Example: bridge, decorator ...
  - Advantage: flexibility.
  - Tradeoffs:
    - requires more coding
    - possibly more allocations
    - having 2 instances (wrapper and encapsulated) may cause confusion with type-checks,
    - Synchronization

Conclusions (cont.)

InterBit
Training & Consulting Ltd.

- **There is still room for creativity!**
  - Recognizing patterns – may be non-trivial, as they may hide under very different facades.
  - Adapting a pattern to the problem in hand.
  - The art of combining patterns!

Conclusions (cont.)

InterBit
Training & Consulting Ltd.

- **The documented solutions may have inspired you to invite new patterns of your own… but be cautious!**
  - Could your new pattern be an existing one in disguise ?
  - Will it be easy for others to understand and maintain ?