

48. Possible parts of the book

Description

This page contains sections that may be included in the book, but not clear where they should be so they have preliminarily been placed here.

In other languages

For each chapter, we may have a section describing in general terms how this may be done in mainstream languages.

Exercises

We should add a section on exercises in each chapter.

Historical notes

We may have a section at the end of each chapter with a short mentioning of historical facts as in the Beta book.

References

We should have a chapter containing references – may be part of *Historical notes*, but perhaps it is more convenient to have all references at one point?

Supplementary properties

We have introduced access modifiers like `%private`.
We should also introduce the ones need for concurrency abstractions.

Exceptions

We should cover exceptions!
Beta use static exceptions as defined by Jørgen Lindskov in his PhD.
Java has checked and unchecked exceptions.
Checked are for situations where a user or client may supply wrong data or wrong arguments to a method or class. The programmer should be prepared for these. This is where `throws`, `throw`, and `try-catch` are used.
Unchecked covers situations where there may be errors in the program and these need not to be handled by the program since the error should be fixed.
Checked exceptions may correspond to static exceptions in Beta
JLK discuss fault tolerance and exceptions in a paper from 2001 where he augment the static exception model. He is close to checked and unchecked.
JLK: Fault Tolerance and Exception Handling in BETA, A. Romanovsky et al. (Eds.): Exception Handling, LNCS 2022, pp. 1-17, 2001.

Programming style

Vi bør overveje et afsnit om programming styles, herunder, comments, variable names, indentation, use of spaces, use of blank lines, ...

Se fx Stroustrup: <https://www.stroustrup.com/PPP-style.pdf>

Assertions

We may consider introducing assertions including invariants and pre- and post-conditions.

BlueJ has a package of classes be used for testing and this package includes assertion methods like `assertEqual`, `assertTrue`, etc.

We should also have such a package for qBeta.

It do, however, require some work!

Design techniques

We should perhaps have a section about identification of phenomena and concepts.

BlueJ has as section on finding verse and nouns. They say that nouns, in addition to define classes, also may define fields.

Very strange!?

They mention scenarios and use cases as a technique to identify nouns and verbs.

And they mention CRC cards, responsibility-driven-design and cohesion, coupling, etc.

Finally design patterns are described.

Og evt. også noget om valg af klasser og metoder ud fra et teknisk synspunkt – vi har jo fokus på modellering så det er den centrale måde at vælge klasser og metoder. Men dækker vi også hvad andre siger – altså fx at en procedure skal gøre én og kun én ting, mm?