```
-- FILE -----------------------------------------------------------------
-- name       : DevGuideLines.txt
-- project    : BoarderZone: Development Environment
-- created    : Leon Poyyayil - 2001-01-26
-- language   : English
-- environment: the human readers mind ... ;-)
-- copyright  : (c) 1990-2015 by Leon Poyyayil (private), Switzerland
-- license    : this is free software licensed under the GPL. see COPYING
-------------------------------------------------------------------------
```

Table of Contents line

1. Executive Summary

This document intends to provide a basis to coordinate the software development process for the BoarderZone.net namespace.

It covers the most basic issues to ensure the code developed by different people can be viewed, understood and enhanced by others. For this to work it is necessary that everybody sticks to some common rules which are outlined herein.

2. Organizational Issues

Documentation has to be kept in non-binary formats only.
-> usage of .txt, .htm, .xml, .rtf, ...
Advantages:
- independent of (possibly platform dependent) tools
- can be 'grepped' through ...
- can be compared with standard diff tools

Test programs go into a sub-package named "test" of the package of the module to be tested.

All identifiers, comments and documentation are to be kept in English
-> least common denominator of languages in IT (more folks are comfortable with English than with German)
-> supports a fluid reading of the code without necessity to switch between languages (keywords are English anyway)

3. Naming Conventions

Generally: according to the Java guidelines, see *[FLAN]* in short:
- Packages: lowercase names (e.g. "net.boarderzone.util")
- Classes: mixed case style, starting with a capital letter (e.g. "SuperDuperTool")
- Interfaces: same as for classes, but DON'T start with the letter 'I' to distinguish them (e.g. "IClientHandler"). Instead the implementing classes should append an "Impl" to their name. This encourages working with the interfaces instead of the implementations ...
- Methods/Variables: mixed case style, starting with a lower case letter (e.g. "getMemberValue()")
- Use self describing names rather than cryptic abreviations. The typing overhead for writing the code is worth the better readability!

Avoid underscores in identifiers (for better 'typeability')

If an identifier contains a commonly used abbreviation, don't write it in all capital letters (e.g. "MyHTTPHandler") but rather capitalize it as a normal word, e.g. MyHttpHandler. This increases readability.

Give all variable names a one- or two-letter lower case prefix to prevent confusion between variables of different scopes etc:
1. Scope prefix:

1

- 'c' for class statics
- 'm' for class members
- 'l' for method locals
- 'p' for method parameters

2. Type prefix:
- 'n' for numeric values of base types
- 'b' for boolean values

Constants are to be kept in all uppercase letters

Examples:
- A global numeric constant:
  ```
  public static final int MY_ID = 12345;
  ```
- A value accessible to all class instances:
  ```
  private static boolean cbInitialized = false;
  ```
- A member instance of some object
  ```
  private SomeClass mOutputFrame;
  ```
- An instance method of some class:
  ```
  private void myFunc(
    final int pnValueToSet // this is a comment for this parameter
  )
  {
    final int lnFactor = 10;            // a local constant
    mnValue = pnValueToSet * lnFactor; // set the member ...
  } // myFunc
  ```

4. Coding

All code has to be located under the package net.boarderzone

...

5. Style

No use of the TAB character, do all indenting with spaces

...

A. References

[FLAN]   Flanagan, David
         Java In A Nutshell, ISBN 1-56592-487-8
         Chapter 7 - Java Programming and Documentation Conventions

B. Revision History

2001-01-26 - lep - created
2001-02-12 - lep - pre-release for public comment ...
2001-11-14 - lep - changed prefix for variables in naming conventions
2002-01-28 - lep - changed naming convention for interfaces
2004-04-22 - lep - adapted to current style & requirements
2004-06-19 - lep - changed to require final keyword where appropriate
2004-06-19 - lep - changed table of contents to point to line numbers
2004-08-16 - lep - fixed some minor typos and added some clarifications
2007-09-20 - lep - updated to reflect new style and findings

C. Authors contact info

Leon Poyyayil (lep) : leon.poyyayil@boarderzone.net

```
-- EOF ----------------------------------------------------------------
```