



Software Engineering Laboratory
Online Bookstore System
Coding Standards

Group name: DEVAMKE

- **Muhammed Said Kaya 21627428**
- **Muhammet Eren Taşdemir 21727743**
- **İhsan Kürşad Ünal 21627758**
- **Emre Hancı 21604552**
- **Ali Kayadibi 21727432**

Prudentia Bookstore	
Coding Standards	Date: 17/05/2020

Prudentia Bookstore

Coding Standards

1 Introduction

This document serves as the complete definition of Prudentia Bookstore's coding standards. This document focuses primarily on the hard-and-fast rules that we follow.

2 Description

Using a coding standard is a widely accepted software development practice. The need for this practice takes on added importance in a highly collaborative environment. The team should have a standard way of naming and formatting things so they can understand the code quickly and know where to look at all times. This enables shared code ownership since any team member should be able to quickly understand the code written by others. Ideally, the coding standard should be the result of team consensus. Involving the team members will aid adoption of the standards.

Good software development organizations want their programmers to maintain to some well-defined and standard style of coding called coding standards. They usually make their own coding standards and guidelines depending on what suits their organization best and based on the types of software they develop. It is very important for the programmers to maintain the coding standards otherwise the code will be rejected during code review.

3 Coding Standards Specifications

3.1. Naming Standards

- Names of classes are pascal case.
- Names of functions and local variables start with a lowercase letter and camel case used and no underscores.
- The name of a class is usually a noun or a noun phrase explaining what the class. Ex: BucketIdentifier.java
- The name of a method is usually a verb or a verb phrase saying what the method does. Ex: getPaidBuckets()
- When a class has multiple constructors, or multiple methods with the same name, these appear sequentially, with no other code in between (not even private members).
- Class/Interface Declaration:

Prudentia Bookstore	
Coding Standards	Date: 17/05/2020

- Class/Interface comment (If present)
- class or interface statement
- Class/interface implementation comment (If present)
- Class/Instance variables
- Getter and setters
- Constructors
- Methods
- Annotations applying to a class, method or constructor appear immediately after the documentation block, and each annotation is listed on a line of its own (that is, one annotation per line)
- Constant name are all upper case letters.

3.2. File Organization

- The source file name consists of the case-sensitive name of the top-level class it contains (of which there is exactly one), plus the .java extension.
- Source file in order:
 - Beginning comments (If present)
 - Package statements
 - Import statements
 - Class and interface declarations (Each class resides in a source file of its own)
 - Exactly one blank line separates each section that is present.
- Names of packages are always lowercase without using underscores.
- Source files are encoded in UTF-8.

3.3. Comment Standards

- Block Comments: Block comments are used to provide descriptions of files, methods, data structures and algorithms. Block comments should be used at the beginning of each file and before each method.
- Single-Line Comments: Short comments can appear on a single line indented to the level of the code that follows. If a comment can't be written in a single line, it should follow the block comment format
- The // comment delimiter: it can be used in consecutive multiple lines for commenting out sections of code and very short comments on the same line as the code they describe, but should be shifted far enough to separate them from the statements.

Prudentia Bookstore	
Coding Standards	Date: 17/05/2020

3.4. Whitespaces

- Four spaces should be used as the unit of indentation.
- One blank line should always be used in the following circumstances:
 - Between methods.
 - Between the local variables in a method and its first statement.
 - Before a block or single-line comment.
 - A single blank line may also appear anywhere it improves readability
- Horizontal Whitespace
 - On both sides of any binary or ternary operator.
 - After ,; or the closing parenthesis ()) of a cast
 - Between the type and variable of a declaration: List<String> list
 - Separating any reserved word, such as if, for or catch, from an open parenthesis (() that follows it on that line
 - Separating any reserved word, such as else or catch, from a closing curly brace (}) that precedes it on that line
 - Before any open curly brace ({}))
 - No space between a method name and the parenthesis.

3.5. Other

- Line lengths are 120 characters by default. A "character" means any Unicode code point.
- When an expression will not fit on a single line, break after comma and before an operator.
- Braces are used with if, else, for, do and while statements.
- One declaration per line is used generally.
- Put declarations only at the beginning of blocks except loops.