| 資料番号 | ECE2300-D |
|---|---|

ソフトウェア開発技法

# 良い設計例と悪い設計例比較

【 オブジェクト指向 】

株式会社　OfficeGoody

COS システム（受注サブシステム）

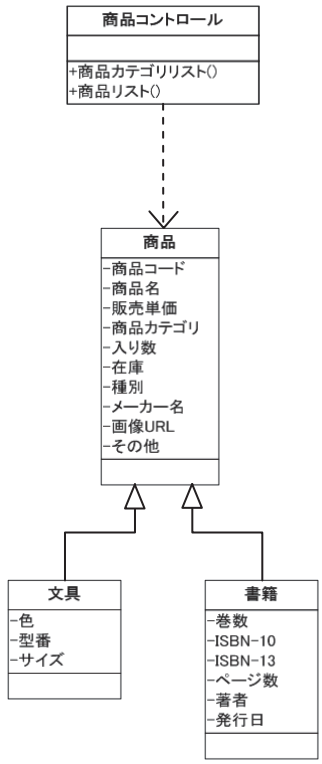
## １．良い設計例と悪い設計例の評価ポイントと特徴

評価ポイント

| | 良い設計例 | 悪い設計例 |
|---|---|---|
| 全体 | ○カプセル化により、「保守性」、「再利用性」を確保する。さらに将来性を考慮し、「拡張性」を意識した設計。<br>・単一責任の原則（SRP）<br>・オープン・クローズドの原則（OCP） | ×カプセル化により、「保守性」、「再利用性」を確保する。しかし将来性を考慮せず、現状のまま設計。そのため「拡張性」、「再利用性」、「保守性」を十分に確保できない。 |
| アクタの設計 | ○アクタの汎化を導出して設計<br>・継承可能なクラスを導出しやすい | ×アクタの汎化を導出せずに設計<br>・継承可能なクラスを導出しにくい |
| クラスの設計 | ○継承が可能なクラスを導出して設計<br>・共通項目を汎化することで、差分コーディングを実現し、生産性の向上を図る<br>・共通項目の修正（仕様変更）に対する保守性を最小限に抑える<br>・汎化を利用することで、他オブジェクトとの関連が単純化され、仕様変更や拡張に対する影響度を最小限に抑える | ×継承が可能なクラスを導出せずに設計<br>・クラスに必要なものを全てコーディングするため、共通項目が複数個所に発生<br>・共通項目の修正（仕様変更）は、クラスが増えるとともに増加する<br>・他オブジェクトとの関連が複雑になり、仕様変更や拡張に対する影響が膨大となる |

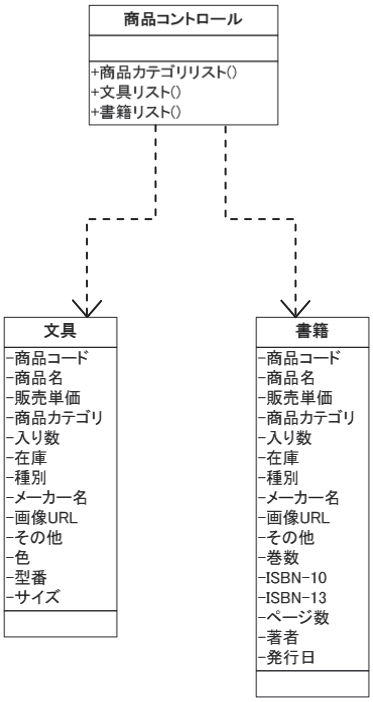
## 1．継承が可能なクラスの導出

良い例




※参照資料：
ECD2122-D　p30
ソフトウェア方式設計書＜良い例＞
サブシステム機能設計書　クラス構成図

悪い例




※参照資料：
ECD2112-D　p30
ソフトウェア方式設計書＜解答例：悪い例＞
サブシステム機能設計書　クラス構成図

【良い例】

・共通項目を汎化クラスとして定義し、各クラスの差分のみを定義する。

例）文具と書籍には、商品コードや販売単価など商品カテゴリに関係なく共通する項目及び商品カテゴリごとに特有な項目の 2 種類が存在する。
共通する項目を「商品」クラスとして定義する。（汎化）
「文具」クラスと「書籍」クラスは「商品」クラスを継承することで、差分だけの定義をする。（特化）

【悪い例】

・共通項目をそのまま各クラスに定義する。そのため同じ内容のコーディングが複数個所で必要となる。

例）文具と書籍には、商品コードや販売単価など商品カテゴリに関係なく共通する項目及び商品カテゴリごとに特有な項目の 2 種類が存在する。
「文具」クラスと「書籍」クラスにはどちらも、共通項目を定義する。