

シラバス（I）

科目名：ソフトウェア開発技法の実践的演習

授業の概要と目的

ソフトウェア開発技法である「構造化技法」と「オブジェクト指向」の2つの技法を取り上げ、実践に近い事例を基に、それぞれの技法が持つ特長を講義と演習を通して理解し、設計原則に基づいた実践的な設計ができるスキルを養う。

ただし、以下の点を踏まえた科目とする。

- (a) 「構造化技法」と「オブジェクト指向」の2コースに大別し、それぞれの技法ごとに「適用事例に基づく技法の特長を捉える」と「課題による実践演習により検証／評価スキルを高める」の2つのフェーズに分割して実施する構成としている。
- (b) 講義で学んだ内容をベースに演習／討議を通して自己の設計スキルアップを狙いとしている。また、第1フェーズは最大2コマとした事例に基づいた講義スタイルで、第2フェーズは「構造化技法」が2コマ、「オブジェクト指向」3コマを最大として、課題演習／討議の構成となっている。

この授業の目的は、

- (a) ソフトウェア開発技法の特長と各技法の実開発における必要性と有効性を理解した上で、ソフトウェアの品質を客観的に評価できる知識を習得する。
 - ・「構造化技法」と「オブジェクト指向」を利用するに当たり、各技法の特長が活かせる領域や対象を客観的に評価できる。
 - ・「構造化技法」と「オブジェクト指向」の企業での実開発における必要性と有効性を理解する。
- (b) 「構造化技法」および「オブジェクト指向」の各技法を具体的に用いて設計することで実践力の強化を図る。
 - ・「構造化技法」を具体的に用いたシステムの構造設計が行なえる。
 - ・「オブジェクト指向」の抽象化の概念を理解し、オブジェクト指向設計が行なえる。

とし、「学生へのソフトウェア開発実践力を醸成」を促すことを目標にしている。

授業形態

事例に基づいた講義と課題演習で構成され、課題演習をホームワークとして課すこともある。

他授業との関連及び履修上の注意

ソフトウェア工学、アルゴリズムを履修していることが前提であり、「構造化技法」ではCプログラミング及びDFD、「オブジェクト指向」ではJavaプログラミング及びUMLを履修していることが前提となる。

授業計画

() 内はコマ数

1. 構造化技法コース (4)

1.1 事例研究：良い例／悪い例による技法の目的と特長 フェーズ	
テーマ	学習内容
・構造化分析／構造化設計 (1)	・構造化技法によるシステム開発の利点と欠点 ・プログラム及びモジュール分割指針や評価尺度（結合度・凝集度）による独立性の高い設計
・構造化技法の実践的適用事例によるケーススタディ (1)	・良い例／悪い例の事例により、適用領域や対象を理解し、品質面(性能、保守性等)からの検証と評価
1.2 実践設計演習と検証／評価 フェーズ	
テーマ	学習内容
・構造化技法の実践演習 1 (1)	改修要件を基に、悪い例による機能追加／変更への対応 ・ソフトウェア詳細設計書の作成及び仕様変更への対応を体験
・構造化技法の実践演習 2 と「構造化技法」の検証／評価 (1)	改修要件を基に、良い例による機能追加／変更への対応 ・ソフトウェア詳細設計書の作成及び仕様変更への対応を体験

2. オブジェクト指向技法コース (5)

2.1 事例研究：良い例／悪い例による技法の目的と特長 フェーズ	
テーマ	学習内容
・オブジェクト指向分析と設計 (1)	・オブジェクト指向の特長である、継承、多態性、インタフェース及び低結合度かつ高凝集度などの設計原則を意識した設計
・オブジェクト指向の実践的適用事例によるケーススタディ (1)	・良い例／悪い例の事例により、適用領域や対象を理解し、品質面(性能、拡張性等)からの検証と評価
2.2 実践設計演習と検証／評価 フェーズ	
テーマ	学習内容
・オブジェクト指向の実践演習 1 (1)	改修要件を基に、悪い例による機能追加／変更への対応 ・UMLを使ったソフトウェア詳細設計書の作成及び仕様変更への対応を体験
・オブジェクト指向の実践演習 2 とオブジェクト指向の検証／評価 (1)	改修要件を基に、良い例による機能追加／変更への対応 ・UMLを使ったソフトウェア詳細設計書の作成及び仕様変更への対応を体験
・開発技法のまとめ (1)	「構造化技法」と「オブジェクト指向」の比較による設計実務での実践的ポイントを学習する

成績評価の方法及び基準

各コースのレポートと課題演習の結果を基に総合的に評価する。