

## ソフトウェア工学(Software Engineering)

担当教員名	奥村 紀之		
学科・専攻、科目詳細	電気情報工学科 情報工学コース 5年 後期 1単位 講義		
学科のカリキュラム表	専門科目 必修科目		
共生システム工学の科目構成表	専門工学科目 専門応用系		
学習・教育目標	共生システム工学	F-1(55%)	G-2(25%) H-1(20%)
	JABEE基準1(1)	(d)(e)(h)	
科目的概要	<p>ソフトウェアの需要が年々飛躍的に高まっているのに対して、ソフトウェアの生産性を急に改善するのは困難である。いわゆる「ソフトウェア危機」である。この「危機」を回避するためには、高品質のソフトウェアを効率的に開発する手法が求められている。この科目では、高品質のソフトウェアを効率的かつ組織的に開発するために発達してきたソフトウェア工学の、種々の技法・手法を講義する。</p>		
テキスト(参考文献)	<p>(教科書) Tucker!著：「憂鬱なプログラマのためのオブジェクト指向開発講座」, 翔泳社          (参考文献) 河村一樹著：「改訂新版ソフトウェア工学入門」, 近代科学社</p>		
履修上の注意	受講に当たっては、プログラミングII・III、データ構造とアルゴリズム、またはそれらに相当する科目を修得しておくことが望ましい。		
科目的達成目標	<p>この科目では、一般的なウォーターフォールモデルによる段階的なソフトウェア開発のプロセスに沿って、各段階における目標と構造化技法について学習する(H-1, F-1)。その上で、近年急激に発展している、オブジェクト指向パラダイムに基づくソフトウェア開発技法についても学習する(G-2)。具体的な達成目標は以下の通りである。</p> <ul style="list-style-type: none"> <li>[1] 一般的なソフトウェア開発工程を理解すること。</li> <li>[2] 要求定義・プログラム設計・プログラミングの構造化技法を理解・習得すること。</li> <li>[3] オブジェクト指向のソフトウェア開発技法について理解すること。</li> </ul> <p>これらのソフトウェア開発技法の学習を通じて、実践的なソフトウェア開発の能力を身につけ、実践的な問題解決能力を醸成する。</p>		
自己学習	<p>目標を達成するためには、授業以外に次の自己学習が必要である。</p> <ol style="list-style-type: none"> <li>1) 適当な例を用いて、構造化分析を行ってみること。</li> <li>2) 適当な例を用いて、構造化設計を行ってみること。</li> <li>3) 適当な例を用いて、構造化チャートを記述してみること。</li> <li>4) 適当な例を用いて、UMLのクラス図等を記述してみること。</li> <li>5) ソフトウェア開発技法の最新の動向を調査し、概要を把握すること。</li> </ol>		
目標達成度(成績)の評価方法と基準	<p>合格の対象としない欠席条件(割合)</p>		1/3以上の欠課
	<p>後期中間試験(50%)、後期期末レポート(50%)の合計100点満点で評価し、60%以上取得したものを合格とする。</p> <p>評価基準として達成目標の各々で修得すべき内容を以下に示す。</p> <ul style="list-style-type: none"> <li>[1] ソフトウェアの一般的な特性、ソフトウェア開発のプロセスモデル</li> <li>[2] 構造化分析、構造化設計、構造化プログラミング</li> <li>[3] オブジェクト指向パラダイム、オブジェクト指向分析・設計、UML以上の内容を中間試験、期末レポート(いずれも100点満点)で出題する。</li> </ul>		
連絡先	okumura@akashi.ac.jp		

授業の計画・内容	
<b>第1週 ソフトウェアの本質</b>	ソフトウェアの一般的定義・一般的構造・一般的特性について解説し、よいソフトウェアの条件を考察する。
<b>第2週 ソフトウェア工学の概要</b>	ソフトウェア工学の発展経緯を概観し、その目標・意義・成果について簡単にまとめ、ソフトウェア工学の全体像を把握する。
<b>第3週 ソフトウェアのプロセスモデルと要求モデル</b>	ソフトウェア開発工程のモデルであるプロセスモデルと、開発するべきソフトウェアのモデルである要求モデルについて解説し、それぞれいくつかのモデルを紹介する。
<b>第4週 構造化分析</b>	デマルコの構造化分析について解説する。また、構造化分析の問題点について触れ、構造化分析を実際に用いる際の手順と具体例を示す。
<b>第5週 構造化分析</b>	プログラムのモジュール構造と、各モジュール間のインターフェースを設計するための構造化技法の一つである構造化設計について解説する。
<b>第6週 構造化プログラミング</b>	プログラムモジュール内の制御構造を詳細に設計する際の構造化技法として、構造化プログラミングについて解説する。
<b>第7週 オブジェクト指向とは何か</b>	オブジェクト指向に対する考え方、オブジェクト指向ソフトウェア開発について説明し、基本概念であるクラスに関して解説する。
<b>第8週 中間試験</b>	
<b>第9週 オブジェクト指向開発のメリット</b>	オブジェクト指向開発が開発者にとってどのようなメリットをもたらすかに触れ、オブジェクト指向分析・設計について概略を説明する。
<b>第10週 静的分析(クラス・関連)</b>	オブジェクト指向分析における静的分析の中で、クラスと関連に関する分析法について解説する。
<b>第11週 静的分析(継承・多重継承)</b>	オブジェクト指向の概念の中で重要な継承について解説する。また、継承を利用することでプログラムを作らないための考え方について学ぶ。
<b>第12週 動的分析(オブジェクトの状態管理)</b>	状態遷移図を利用したオブジェクトの状態変化を管理するための手法について解説する。
<b>第13週 動的分析(メッセージシーケンス)</b>	オブジェクト間のメッセージのやりとりについて、シーケンス図を利用した分析手法について解説する。
<b>第14週 設計・実装</b>	実際に動作するプログラムを作成するための設計方法、実装方法について解説する。また、豊富なクラスライブラリを紹介する。
<b>第15週 実習</b>	課題を与え、オブジェクト指向分析・設計に則った開発を行い、プログラムを作成する。レポート提出有り。
<b>期末試験実施せず</b>	