久留米工業高等専門学校	開講年度	平成31年度 (2019年度)	授業科目	コンピュータサイエンス	
科目基礎情報					
科目番号	7E11	科目区分	専門 / 選択		
授業形態	講義	単位の種別と単位数	学修単位: 2		
開設学科	機械・電気システム工学専攻 (電気電子工学コース)	対象学年	専2		
開設期	前期	週時間数	2		
教科書/教材	Java言語で学ぶ デザインパターン入門, 結城 浩, SBクリエイティブ				
担当教員	加藤 直孝				
到達目標					
<p>オブジェクト指向言語であるJavaを用いて, the Gangs of Four (Erich Gamma, Richard Helm, Ralph Johnson, John Vissides) によるデザインパターンを学ぶ。23個のすべてを学ぶのではなく, その3分の1程度を学ぶ。オブジェクト指向言語では, インターフェースを考えるときにデザインパターンを学んでおくと, いろんな場面でそのパターンを適応できることが多い。そこで, 本科目では, 実際に動くデザインパターンのプログラムを書くことにより, 再利用しやすく拡張しやすいプログラムを作成するコツをつかむことを目指す。オブジェクト指向プログラミングとは何かが体験できることを目標とする。</p>					
ルーブリック					
	理想的な到達レベルの目安	標準的な到達レベルの目安	未到達レベルの目安		
評価項目1	Javaによる基本的なプログラミングが自由にできる。	Javaによるプログラミングができる。	Javaによるプログラミングができない。		
評価項目2	オブジェクト指向のプログラミングが自由にできる。	オブジェクト指向のプログラミングが理解できる。	オブジェクト指向のプログラミングが理解できない。		
評価項目3	デザインパターンを適用したプログラミングができる。	プログラム中のデザインパターンを見抜くことができる。	デザインパターンが使われているかどうか理解できない。		
学科の到達目標項目との関係					
JABEE B-1					
教育方法等					
概要	<p>オブジェクト指向言語であるJavaを用いて, Gangs of Four (Erich Gamma, Richard Helm, Ralph Johnson, John Vissides) によるデザインパターンを学ぶ。23個のすべてを学ぶのではなく, その3分の1程度を学ぶ。デザインパターンは概念であるがゆえに, 本を読むだけでは理解できないことが多い。プログラムのパーツがどのように組み立てられているか, あるいはどのように関連しているかということを, デザインパターンの視点から学習する。オブジェクト指向言語では, インターフェースを考えるときにデザインパターンを学んでおくと, いろんな場面でそのパターンを適応でき役立つ。オブジェクト指向のプログラミングをしたい学生に有益である。実務経験のある教員による授業科目: 加藤は企業 (IBM) でソフトウェアの開発を担当していた。その経験を活かし, オブジェクト指向におけるソフトウェア・インターフェースについて授業を行う。</p>				
授業の進め方・方法	<p>この授業では, Gangs of Four (Erich Gamma, Richard Helm, Ralph Johnson, John Vissides) によるデザインパターンを学ぶが, 彼らの原著は使わない。最終的には原著を読むことは大切であるが, 原著はかなりわかりにくい。この授業では, 日本人による教科書を使用して授業を進める。この授業では実際に手を動かしプログラムを書いて, パターンを含んだプログラムを経験することに重点を置く。実際には, Javaでプログラミングしながら, デザインパターンを1つ1つ理解して, 練習問題を解きながら進めていく。将来プログラム開発に携わる学生にとって, 非常に有益な時間となるはずである。授業の最後に, チームでデザインパターンを使って, 好きなプログラムを開発する。評価は, そのプログラムとその発表で行う。Javaを初めて学ぶ学生がいた場合は, 下記注意点に挙げた教科書を用いて, Javaの授業も少し行う。はじめてJavaを学びたい学生にも役立つ授業とする。</p>				
注意点	Javaを初めて学ぶ学生のため, 上記の教科書と同じ著者の「Java言語 プログラミングレッスン 上・下 (SBクリエイティブ)」を使用する。				
授業計画					
	週	授業内容	週ごとの到達目標		
前期	1stQ	1週	オリエンテーション Javaの導入 (Eclipse等)	Javaのプログラミング環境を作る	
		2週	Javaに関する説明	Javaでプログラミングできる	
		3週	UMLについて学ぶ	最低限のUMLが理解できる	
		4週	Javaである程度プログラミングができるようになる。 Iterator Design Pattern (1)を学ぶ	Javaでプログラミングができる。	
		5週	Javaに関するTutorialを見る Iterator Design Pattern (2)を学ぶ	独力でJavaを学習できる Iterator Design Patternを理解できる	
		6週	Adapter Design Patternを学ぶ	Adapter Design Patternを理解する Adapter Design Pattern の練習問題を解くことができる	
		7週	デザインパターンを使ったチームのプログラムを作る (1)	デザインパターンを使ったチームのプログラムを作る (1)	
		8週	Template Method Design Pattern を学ぶ	Template Method Design Pattern を理解できる	
	2ndQ	9週	Factory Method Design Pattern を学ぶ	Factory Method Design Pattern を理解できる	
		10週	Singleton Design Patternを学ぶ	Singleton Design Patternを理解できる	
		11週	Prototype Design Patternを学ぶ	Prototype Design Patternを理解できる	
		12週	Builder Design Patternを学ぶ	Builder Design Patternを理解できる	
		13週	チームのプログラムの発表を行う (1)	開発したプログラムをPowerPointで説明し, プログラムをデモできる。	
		14週	チームのプログラムの発表を行う (2)	開発したプログラムをPowerPointで説明し, プログラムをデモできる。	
		15週	予備	予備	
		16週			
モデルコアカリキュラムの学習内容及到達目標					
分類	分野	学習内容	学習内容の到達目標	到達レベル	授業週

専門的能力	分野別の専門工学	情報系分野	プログラミング	代入や演算子の概念を理解し、式を記述できる。	3	
				プロシージャ(または、関数、サブルーチンなど)の概念を理解し、これらを含むプログラムを記述できる。	3	
				与えられた問題に対して、それを解決するためのソースプログラムを記述できる。	3	
				ソフトウェア生成に必要なツールを使い、ソースプログラムをロードモジュールに変換して実行できる。	3	
				要求仕様に従って、標準的な手法により実行効率を考慮したプログラムを設計できる。	3	

評価割合							
	試験	発表	相互評価	態度	ポートフォリオ	その他	合計
総合評価割合	0	30	0	0	0	70	100
基礎的能力	0	10	0	0	0	30	40
専門的能力	0	0	0	0	0	30	30
分野横断的能力	0	20	0	0	0	10	30