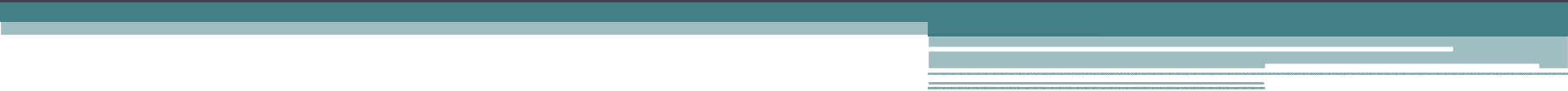


実務者にも役立つ ソフトウェア工学入門



教育目標

大規模なソフトウェア製品を系統的に開発するために、ソフトウェアのライフサイクルモデルが提案されている。

本講義(ソフトウェア工学)ではライフサイクルにおける各種の技術(段階的詳細化, データ抽象化, 構造化プログラミング, プログラミングスタイル, テスト技法等)を系統的に身につけることを目的としている。

これらの技術は、ソフトウェア開発者にとっての必須技術である。また、ソフトウェアを効果的に委託開発するための重要技術でもある。

教育の対象者

- プログラミング教育を受けた情報系の学部学生
- IT企業等の新人ソフトウェア開発技術者
- 情報システムユーザー企業等のIT調達担当者
 - 本講義において、ITベンダー企業の技術者を科目等履修生として受け入れた実績あり.
 - 各種の学外向けセミナーで本講義のコンテンツを活用
 - ソフトウェア企業の技術者を対象としたソフトウェア開発技術セミナーの実施実績あり.

授業計画(その1)

第1回演習

第1回 ソフトウェア工学とは

- ソフトウェア工学の目的
- ITスキル標準, 情報処理技術者試験, 技術士

第2回 ソフトウェアのモデル化(1)

- プロセスモデル : ソフトウェア開発の基本手順

第3回 仕様分析(1)

- 仕様分析の目的と重要性
- データフロー図(DFD)と段階的詳細化
- 正規表現を用いたデータ構造の表現

第4回 仕様分析(2)

- 入出力設計
- ユースケースを用いた仕様記述
- ユースケース作成事例

第5回「DFDの作成」演習

- 概略仕様をDFDの全体文脈図で表す.
- 全体文脈図を詳細化する.
- 全体文脈図の各データフローを正規表現で表す.

第6回 ソフトウェアのモデル化(2)

- コストモデル : Function Point法, Cocomo

第7回 ソフトウェアのモデル化(3)

- 計算モデル : 構造化モデルとオブジェクト指向モデル

第8回 モジュール設計(1)

- モジュール設計の目的
- 段階的詳細化
- モジュール間結合度

授業計画(その2)

第9回 モジュール設計(2)

- モジュール強度
- モジュールの抽象化
- データ抽象化と抽象データ型

第2回演習

第10回「モジュール設計」演習

- DFD中のファイルに対応するデータ構造とそれをアクセスするモジュールを設計する.
- DFD中の機能に対応するモジュールとルーチンを設計する.

第11回 詳細設計

- 詳細設計とその目的
- 構造化プログラミング
- Jackson法

第3回演習

第12回「Jackson法」演習

- Jackson法を用いてアルゴリズム設計を行う.

第13回 コーディング

- ソフトウェアの品質
- プログラミングスタイル
- プログラミングの落とし穴

第14回 ソフトウェア・テスト(1)

- ソフトウェアの検証とテスト
- ソフトウェア・テストの目的
- ホワイトボックステスト

第15回 ソフトウェア・テスト(2)

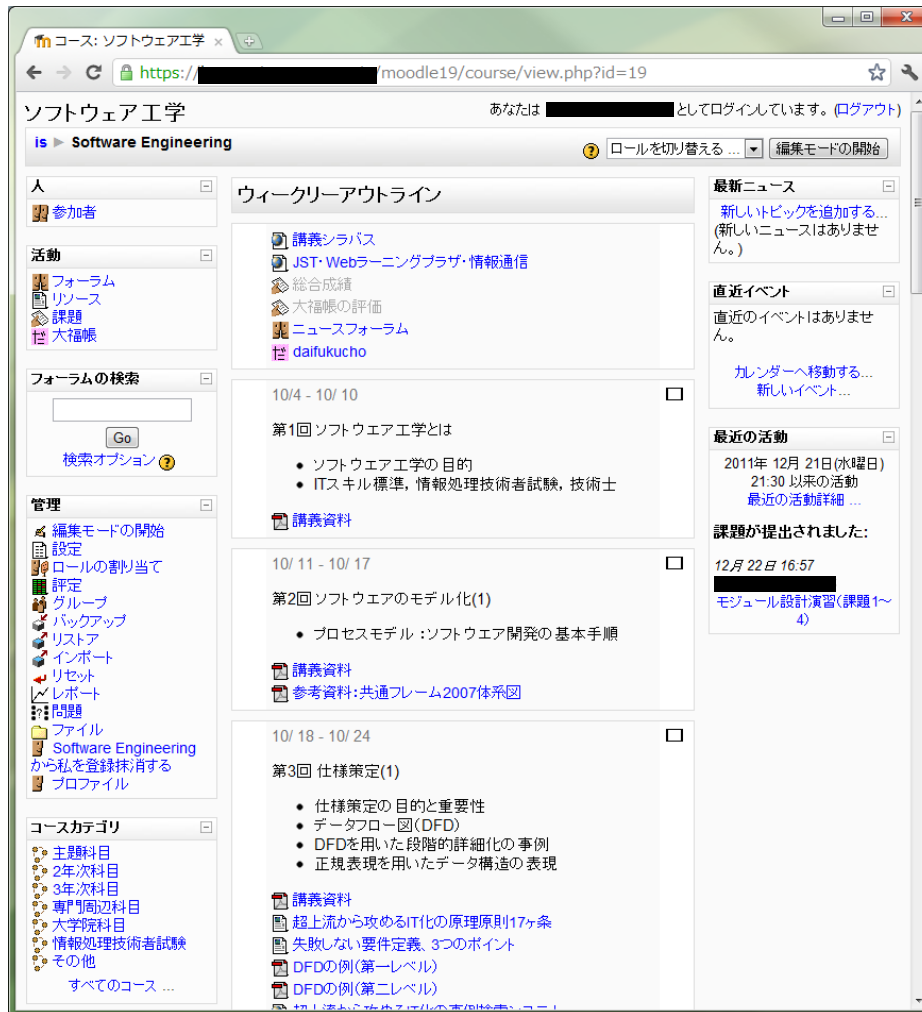
- ブラックボックステスト
- 統合テスト

第16回 定期試験

特徴

- ソフトウェア開発における現実的事例を踏まえた授業設計
 - 各種の業務システム開発経験を通じて蓄積した経験を活用
 - 例: 大学入試合否判定プログラム開発, 学生総合情報システム開発プロジェクトリーダー, 病院総合情報システム導入アドバイザー, 商用パッケージソフト企画・設計, Webを用いた汎用アンケートシステムの企画・仕様策定
- Moodleの徹底活用
 - 講義資料および参考資料の公開
 - レポートのオンライン提出およびフィードバック
 - Moodle版大福帳を活用した学生と教員のコミュニケーション促進(毎回の授業後の学生の意見・質問収集および次回授業前までの教員の回答)
 - 成績通知, 各種連絡
- 3回のPC演習を通じた系統的なスキル育成
 - DFD作成
 - モジュール設計
 - Jackson法を用いた構造化プログラミング
- 設計ガイドラインやプログラミングガイドラインを活用した成績評価基準の明示, 成果物レビュー, および成績評価の実施
- 独自開発したソフトウェア設計支援ツールPerseusを活用したPC演習
- 関連する授業と連携した系統的な授業設計
 - プログラミング概論・演習IおよびII
 - オブジェクト指向開発
 - 情報システム実験
- JABEE認定基準(情報および情報関連分野)への適合

Moodleの徹底活用



- 講義シラバスの公開
- 講義コンテンツの公開
 - 講義資料
 - 演習用ファイル
 - ソフトウェアダウンロード
- 大福帳
- レポート提出の受付
- レポート評価およびコメントのフィードバック
- 成績通知
- アクセスログによる学生の修学状況確認
- メッセージ送受信
- 小テスト(本講義では未実施)
- 各種連絡

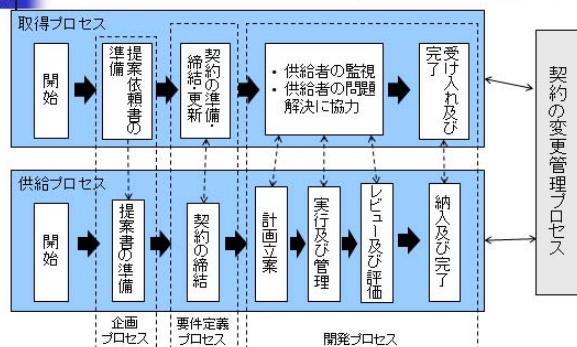
講義資料(その1)

IT人材のレベルとその判定

出典: 高度IT人材の育成をめざして(2007-7-20)

高度IT人材	レベル	レベル	成果(実績)ベース ↓ 業務経験や面談等	試験+業務経験	スキル(能力)ベース ↓ 試験の可否	各企業で判断	情報処理技術者試験
スーパーハイ	レベル7	国内のハイエンドプレイヤーかつ世界で通用するプレイヤー	レベル6	国内のハイエンドプレイヤー	レベル5	企業内のハイエンドプレイヤー	高度試験
	レベル6	国内のハイエンドプレイヤー		企業内のハイエンドプレイヤー		高度な知識・技能	
ハイ	レベル4	高度な知識・技能	レベル3	応用的知識・技能	レベル2	基本的知識・技能	ミドル試験
	レベル3	応用的知識・技能		基本的知識・技能		最低限求められる基礎知識	
ミドル	レベル2	基本的知識・技能	レベル1	最低限求められる基礎知識		試験の可否	エントリ試験
エントリ	レベル1	最低限求められる基礎知識		試験の可否			

共通フレーム2007・主ライフサイクルプロセス 情報システムの取得と供給



15

ソフトウェア工学資料

Function Point(FP)法

FP = [入力FP + 出力FP + 照会FP + 編集ファイルFP + インタフェースFP] × 補正係数

1つの外部入力に対するFP値

開発ファ イル型	データ要素型	1-4	5-15	16以上
0-1	3	3	4	5
2	3	4	5	6
3以上	4	5	6	7

1つの外部出力に対するFP値

開発ファ イル型	データ要素型	1-5	6-15	20以上
0-1	4	4	5	6
2-3	4	5	6	7
4以上	5	6	7	8

1つの照会入力に対するFP値

開発ファ イル型	データ要素型	1-4	5-15	16以上
0-1	3	3	4	5
2	3	4	5	6
3以上	4	5	6	7

1つの照会出力に対するFP値

開発ファ イル型	データ要素型	1-5	6-15	20以上
0-1	3	3	4	5
2-3	3	4	5	6
4以上	4	5	6	7

1つの内部編集ファイルに対するFP値

レコード 要素型	データ要素型	1-19	20-50	51以上
0-1	7	7	10	15
2-5	7	10	15	20
6以上	10	15	20	25

1つの外部インタフェースファイルに対するFP値

レコード 要素型	データ要素型	1-19	20-50	51以上
0-1	5	5	7	10
2-5	5	7	10	15
6以上	7	10	15	20

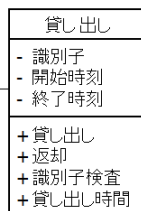
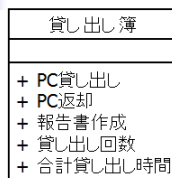
補正係数 = 下記のスコア合計 / 100 + 0.65

データ通信	0: バッチアプリケーション 1: 遠隔からの印刷またはデータ入力 2: 遠隔からの印刷とデータ入力 3: アプリケーションへの遠隔地端フロントエンド 4, 5: 遠隔地端を行う
機能の分散度	0: 単一アプリケーション 1: 他コンポーネントのためにデータを調整するアプリケーション 2, 3: 複数のコンポーネントに分散 4: 非常に多くのコンポーネント上で分散的に実行する
性能目標	0: 特になし 1: 実現のために特異な工数が必要
利用傾向	0: 特になし 1: 実現のために特異な工数が必要
トランザクション量	0: 処理量が同量にならない 1: 実現のために特異な工数が必要
オンラインデータ入力	0: 余剰的な処理が15%以下 1: 余剰的な処理が20%以上
エンドユーザ参加	0: 特になし 1: 実現のために特異な工数が必要
オンライン更新	0: オンライン更新なし 1: データのバックアップと復元が必要で、実現が困難
複雑な処理	0: 特になし 1: 非常に複雑
再利用可能性	0: 再利用しない 1: 多くの部分を他のアプリケーションで広範囲に再利用
インストールの容易性	0: 特になし 1: 実現のために特異な工数が必要
操作の容易性	0: 非常に簡単でない 1: 実現のために特異な工数が必要
複数サイト	0: 1箇所のみで利用 1: 多くの離れた地で利用
変更の容易性	0: 変更の必要なし 1: 新データやテーブルを迅速に変更できるような実装アプリケーションを開発

講義資料(その2)

オブジェクト指向プログラムの構造

クラス図



インスタンス図



最長貸し出し時間を求める

- 貸し出し簿にメソッド「最長貸し出し時間」を追加

午前と午後の報告書を分離する

- メソッド「報告書作成」を修正
- 午前/午後の区別を引数とする

未返却分の除去

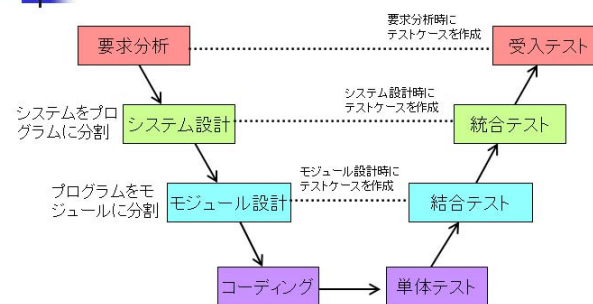
- メソッド「報告書作成」を修正

7種類のモジュール強度

		他モジュールからの独立性	誤りの発生しやすさ	再利用性	拡張性
望ましい強度	機能的強度	高	低	高	高
	情動的強度	高	中	高	高
例外的に認められる強度	連絡的強度	中	低	中	中
	手順的強度	中	低	低-中	中
要改善	時間的強度	低	中	低	中
	論理的強度	中	高	中	低
	暗号的強度	低	非常高	低	低

ソフトウェアのテストはシステム開発工程の全体で行なう

教科書
p. 166



増加テスト技法の比較

	トップダウンテスト	ボトムアップテスト	サンドイッチテスト
上位モジュールのテスト	何度も行なう(好ましい)	最後に行なう	何度も行なえる(比較的好ましい)
スタブ・ドライバの作成	モジュール毎にスタブが必要(比較的容易)	モジュール毎にドライバが必要(複雑)	スタブおよびドライバが必要(複雑)
モジュール開発順序の制約	上位モジュールほど早く完成させる必要がある(制約強)	下位モジュールほど早く完成させる必要がある(制約中)	開発順序は特に問わない(制約弱)
結合テストの順序	順次行なう必要がある	並行して行なえる(好ましい)	

「DFDの作成」演習

目的

- 与えられた企画を分析して、あいまいさおよび矛盾のないDFDを作成できる。

課題

- 与えられた企画を分析してDFDを作成し、DFDを構成する各ファイルおよびデータフローを正規表現で定義する。

レポート評価基準

- コンテキストダイアグラムおよび各レベルのDFDは互いに一貫しているか。
- 各機能の入出力は正しいか。
- 各正規表現は正しいか。
- 提出期限を守っているか。

企画例: 企業情報分析システム

システムが管理するテーブル

- 企業概要, 株価, 単独決算

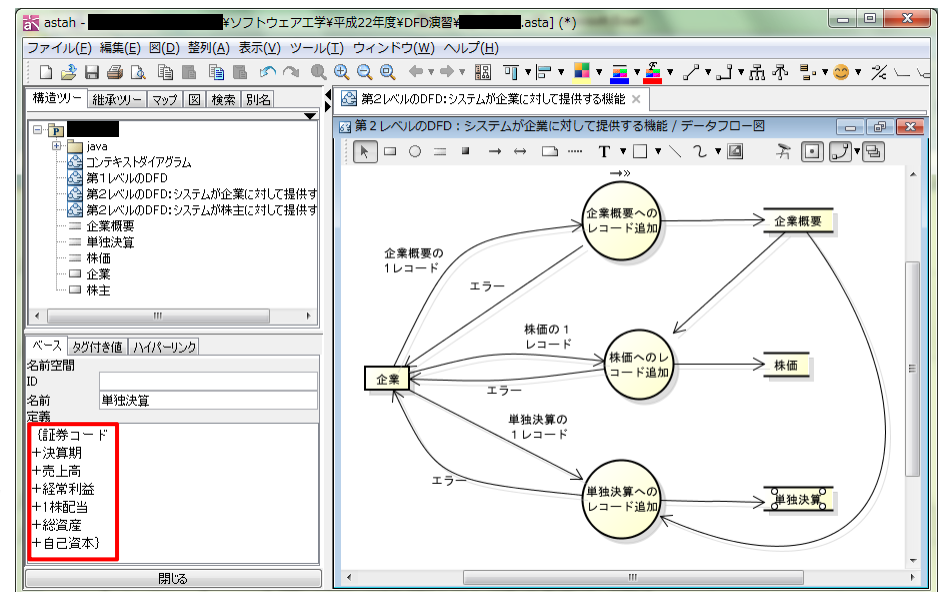
企業に提供する機能

- 各テーブルへのレコード追加(データチェックを含む)

株主に提供する機能

- 指定した人数以上の従業員がいる企業の一覧を表示する。
- 指定した証券コードの企業の株価一覧を表示する。
- その他, 各種のデータ分析機能を提供する。

正規表現
の記述



「モジュール設計」演習

目的

- 仕様書に基づいて段階的詳細化を行い、独立性の高いルーチンやモジュールを設計できる。
- DFD作成, モジュール設計, アルゴリズム設計に至るワークフローを理解する。

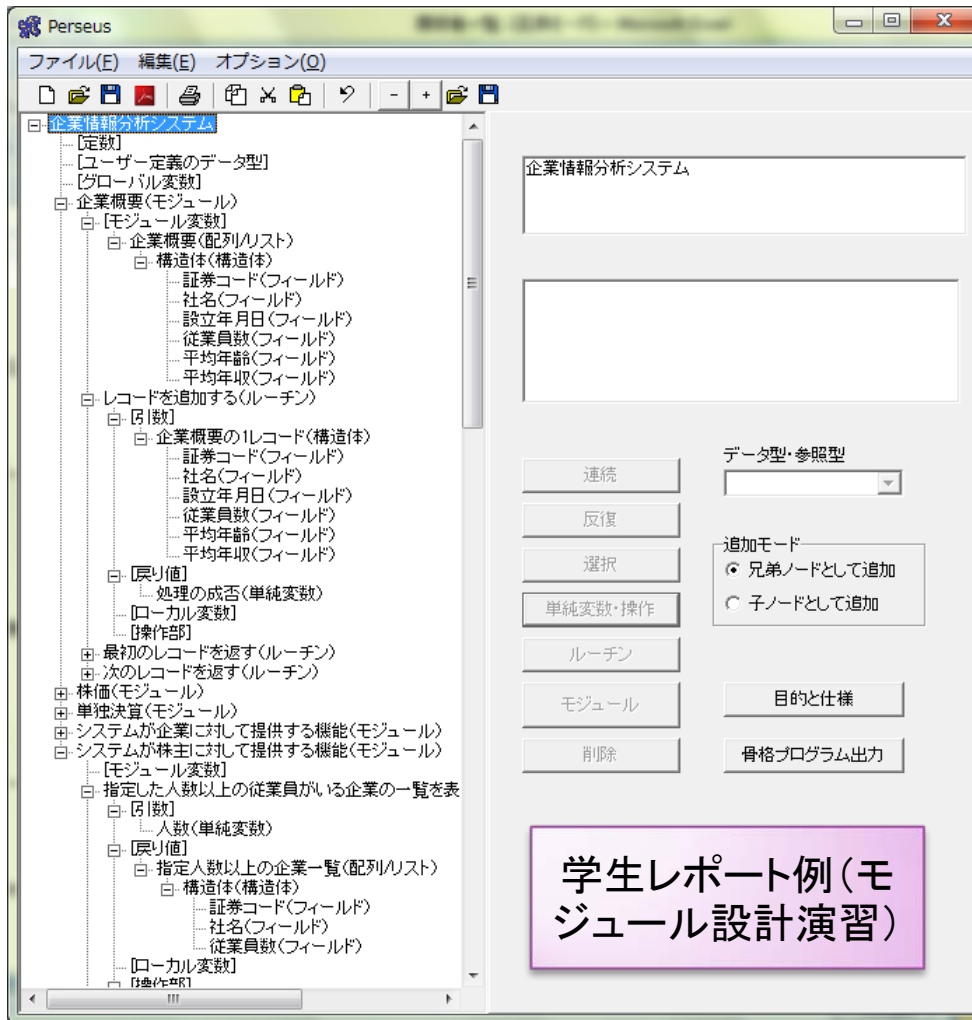
課題

- DFD作成演習で作成したDFDに基づいてモジュールおよびルーチンを設計する。
- 設計したモジュールについて、モジュール間結合度およびモジュール強度を判定する。

レポート評価基準

- 「1ルーチン=1機能」の原則を守っているか。
- 各モジュールが保持するデータ構造が正しく定義されているか。
- モジュールが保持するデータ構造を操作するためのアクセスルーチンが正しく定義されており、情動的強度のモジュールになっているか。
- 各ルーチンの引数および戻り値はDFDと一致しているか。
- ルーチン間の呼び出し関係はDFDと一致しているか。
- モジュール間結合度とモジュール強度の判定結果および判定理由は妥当か。
- 提出期限を守っているか。

ソフトウェア設計支援ツールPerseus

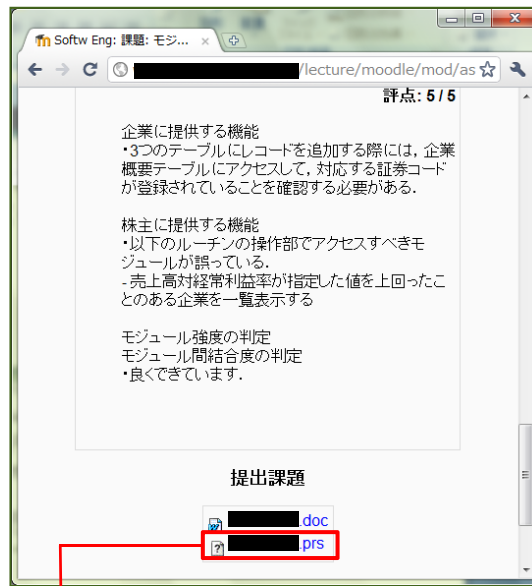


- 我々が研究開発したソフトウェア設計支援ツール
- 設計データをツリーで表現し、編集機能を提供
- モジュール設計, ルーチン設計, データ構造設計, アルゴリズム設計を幅広くカバー
- 構造化設計・オブジェクト指向設計の両方に対応
- 設計結果に基づきC++/Javaプログラムの骨格を生成
- 設計結果の自動チェック機能およびレビュー支援機能
- Javaプログラムのリバースエンジニアリング機能

モジュール設計演習およびJackson法演習で活用

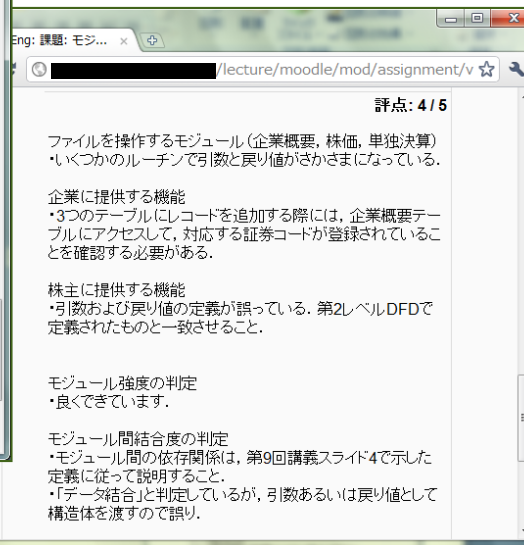
レポートのオンライン提出およびフィードバック

モジュール設計演習におけるフィードバック例

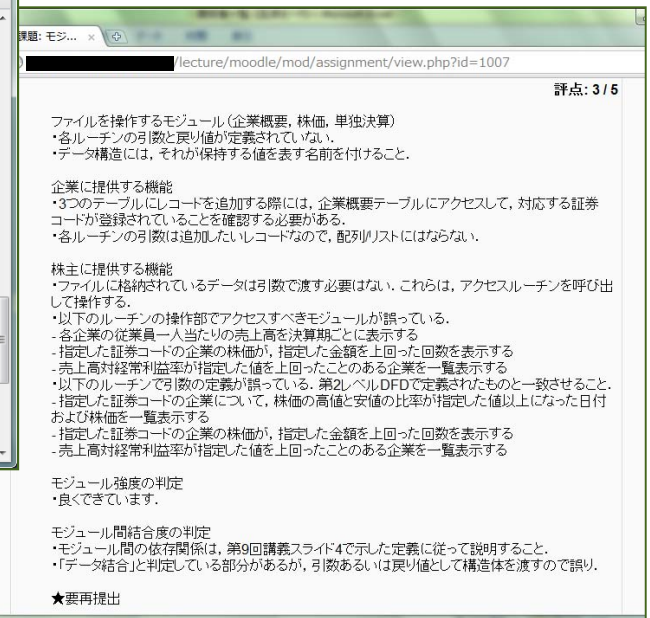


評価：優

Perseusで作成
したファイル



評価：良



評価：可

「Jackson法」演習

目的

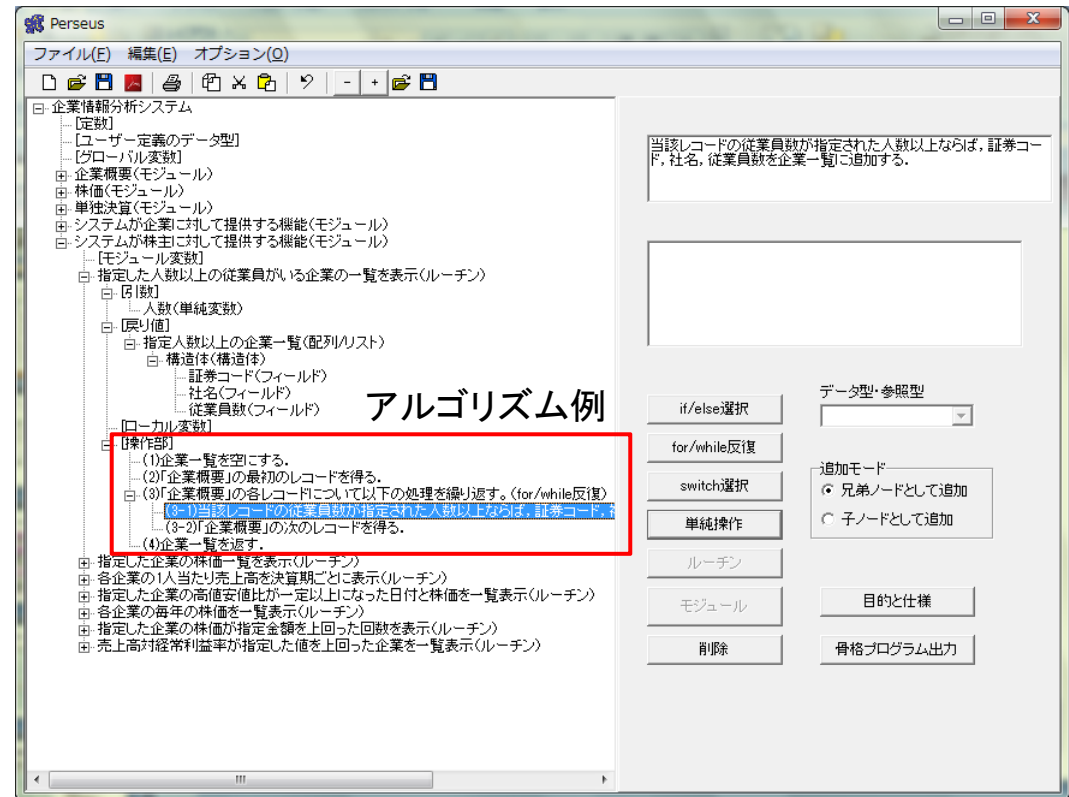
- Jackson法を用いた構造化プログラミングができる。

課題

- モジュール設計演習の結果に基づき、割り当てられたルーチンのアルゴリズムをJackson法を用いて作成する。

レポート評価基準

- Jackson法を正しく適用しているか。
- 各操作は具体化されているか。
- 不要な操作を除去しているか。
- 提出期限を守っているか。



各学生は、モジュール設計演習で自らがPerseusで作成した成果物を基にJackson法演習を行う。

設計ガイドライン

- アルゴリズム記述の基本

- アルゴリズムは、有限回の操作で終了しなければならない。
- アルゴリズムの各操作は、明確に定義されており、誰が実行しても同一の結果が得られなければならない。
- アルゴリズムの各ステップは、仕様の中で使われている概念や用語を使って記述すること。
- アルゴリズムは、記述の洩れや余分な記述を含んではならない。

- データ構造

- 変数名には当該変数が保持する情報を説明する具体的な名詞を用いること。
- 配列を用いる場合、要素数と要素のデータ型を明示すること。
- 構造体を用いる場合、各フィールドについて、フィールド名とフィールドのデータ型を明示すること。

- 制御構造

- 場合分けの条件がすべての場合を網羅していることを確認すること。
- 場合分けは、系統的な順序で配列すること。（例：出現頻度順，年代順）
- 繰り返し処理を記述する場合には、各回の繰り返しで処理されるデータ（または処理の内容）ができるだけ明確になるように工夫すること。

- ルーチン

- 各ルーチンの機能は明確かつ単純に表現すること。従って、複数の機能を持つルーチンを定義してはならない。
- 各ルーチンのアルゴリズムは20ステップ程度以下にまとめること。それ以上の長さのものはルーチンを使用して詳細化すること。
- ルーチン名には、当該ルーチンが提供する機能を表す動詞句を用いること。

成績評価

成績評価基準

- 情報システム開発に関する基礎的知識を理解している。
 - 複数分野における情報システムの活用事例を知っている.
 - 情報システムの開発者としての視点からの知識を持っている.
 - 実社会で発生している問題について関心および知識を持っている.
 - 実社会で発生している問題に対して, 情報システム技術者としての観点から意見を述べることができる.
- 3回の演習レポートおよび定期試験を通じて評価
- 与えられた企画を分析して, あいまいさおよび矛盾のないDFDを作成できる.
- 「DFDの作成」演習を通じて評価

- 仕様書にもとづいて段階的詳細化を行い, 独立性の高いルーチンやモジュールを設計できる.

→ 「モジュール設計」演習を通じて評価

- Jackson法を用いた構造化プログラミングができる.

→ 「Jackson法」演習を通じて評価

評価方法

- 「DFDの作成」演習: 20%
- 「モジュール設計」演習: 25%
- 「Jackson法」演習: 10%
- 定期試験: 45%
- 上記の他, 大福帳で学生の理解度を高める質問や教育改善に役立つコメントを行った学生には加点する.

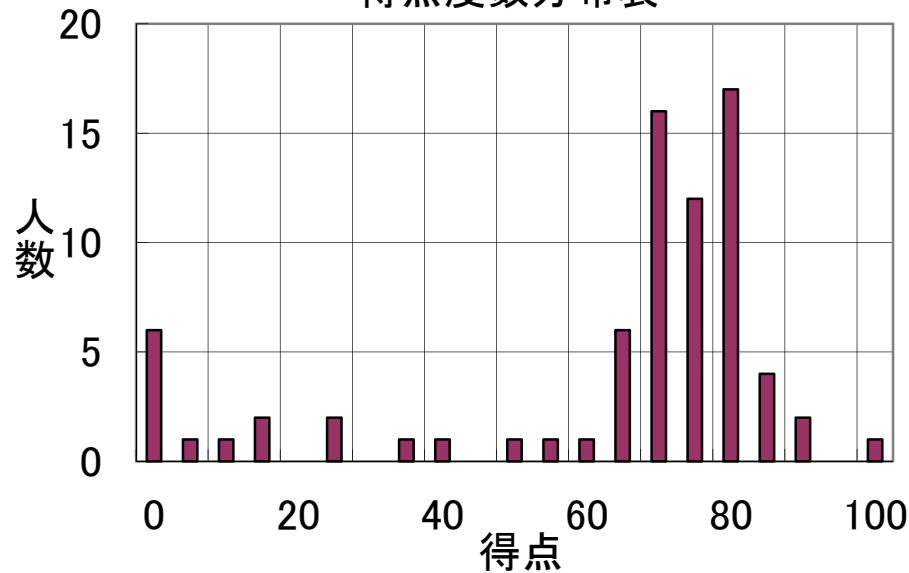
成績評価結果

平成22年度成績 [互換モード] - Microsoft Excel

	A	B	C	D	G	H	I	K	L
	学籍番号	氏名	大福帳	DFD演習	モジュール 設計演習	Jackson 法演習	定期試 験	総合評価	備考
1									
2			45	68	75	61	66		
3			1.60	3.87	3.03	3.58	73.65		
22			2	4	4	4	76	80	
23			1	3	4	3	73	71	
24			1	4	4	4	68	75	
25			2	4	4	3	68	74	
26			2	4	4	5	89	88	4位
27			2	4	4	3	91	84	
28			4	4	4	4	64	72	
29			2	5	4	4	79	85	
			1	4	3	5	75	75	
			2	4	4	3	75	77	
			1	4	3	2	79	71	
				3	4	4	75	73	
				3	3	4	77	69	
			2	4	4	4	82	82	
				4	0		25	27	
				4	4	3	91	82	
			1	5	4	3	73	79	
			1	3	3	2	69	63	
			1	4	4	3	64	71	
			1	3	4	4	59	67	
			2	3	3	2	76	67	
			2	4	4	3	86	82	
			2	4	3	5	75	76	
			1	5	4	5	72	83	
			2	3	4	4	81	78	
			1	3	3	3	84	71	
			1	4	3	3	71	69	
			1	5	3	4	85	82	

不合格者等 定期試験 Moodle 成績

得点度数分布表



- # 授業日毎の大福帳
- ## 第9回 モジュール設計(2)における例

Softw Eng: daifukucho x

https://lecture/moodle/mod/daifukucho/read_id.php?course=12&studentid=

ソフトウェア工学

3G講義 ▶ Softw Eng ▶ 大福帳 ▶ daifukucho

この大福帳を更新する

の大福帳 (ソフトウェア工学)

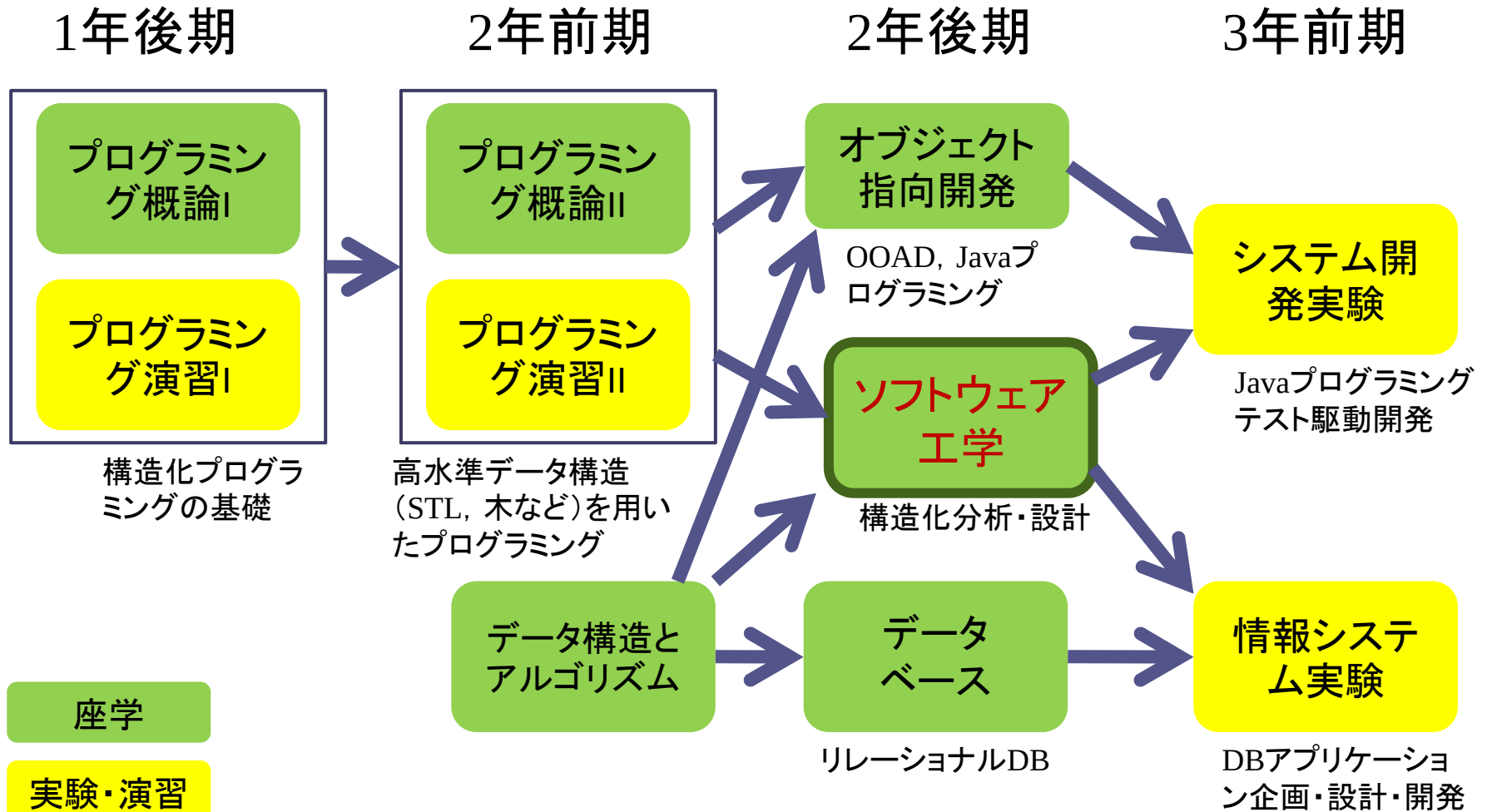
- 大福帳は、学生と先生間のコミュニケーション促進を通じて、勉学や学生生活等を支援するとともに、授業を改善するためにも活用すること、両者に「福」をもたらすことを目的としています。
- 大福帳には、なるべく当日の授業に関係すること(疑問、質問、コメント、意見など)を書いて下さい。
- 他の学生の理解や授業改善を促進するよう良い質問やコメントを書いた学生には、成績評価の際にボーナス点を加算します。

日付	学生のメッセージ	先生からの返事	回答作成
10/05	ソフトウェア工学といっても細分化すれば様々な種類の職業にわかれている事をすることができてよかったです。その中でどれが自分に合っているのか探していきたいと思えます。	IT関係の職業は様々なので、進路を決定する際に良く考えること、IPAのITスキル標準のページを勉強すると良いでしょう。	回答
10/12	体の具合は大丈夫ですか？早く良くなるといいですね。	そうですね、なかなか咳と痰が止まりませんが、なるべく早く治したいと思います。	回答
10/21	一昨日出し忘れました、すいません。集中して内容を理解することができました。	理解したことを来週のDFD演習でしっかり活かしてください。書き込み遅れ	回答
10/26	DFDの演習は難しかったです。レポートは大変そうなので早めに終わらせていきたいです。	授業で学んだことを理解するために演習を行っています。指示に従って取り組めば、ちゃんとできます。ただし、作業量はそれなりにあります。	回答
11/02	確認事項をしっかりと書いていないと、漏れが生じたりするので気ををつけたいと思いました。	そうですね。配布したケースの確認事項には、概略仕様の中で何か曖昧な点を見抜くためのノウハウが詰まっているので、しっかり勉強することを勧めます。	回答
11/09	費用、工期の関係がわかりました。	かなり規模の大きな話ですが、そういう規模のソフトウェアはたくさんあります。	回答
11/16	オブジェクトという言葉には、インスタンスとクラスを含んでいる時と、単にインスタンスだけを示す時の2パターンあるということを知りました。	ITの世界では良くあることですが、用語に惑わされずに内容を理解することが重要です。	回答
11/30	オセロゲームを作るだけでもあんなにたくさんの機能に段階的に分けていらないといけないからRPGのゲームなどはかなり大変なんだなって思いました。	RPGの場合、オセロの例の100倍ぐらいの規模にはなるでしょうね。そのぐらいの規模になると、モジュール設計をしないと実現不可能です。	回答

学生毎の大福帳

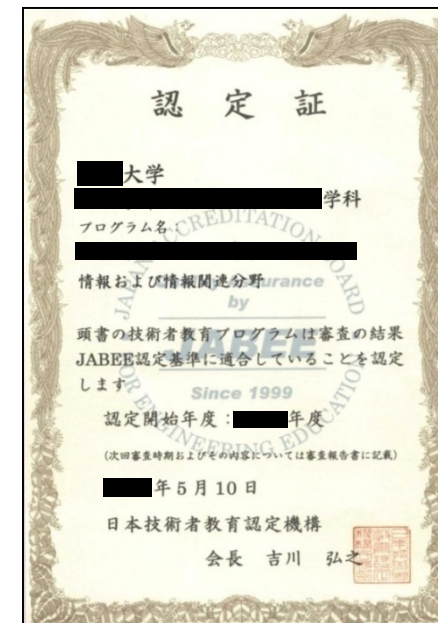
各学生は自らの大福帳のみ参照可

授業間の関連



教育効果

- 情報システム開発に関する基礎的知識を理解している.
- 与えられた企画を分析して, あいまいさおよび矛盾のないDFDを作成できる.
- 仕様書にもとづいて段階的詳細化を行い, 独立性の高いルーチンやモジュールを設計できる.
- Jackson法を用いた構造化プログラミングができる.



JABEEによるアクレディテーション認定を取得することにより, 学習・教育目標およびシラバスに従った授業の実施および成績評価が実施されていることが, 第三者の専門家により確認されている.