

17j. オブジェクト指向の本質に基づいたアプリケーション設計 ～品質、コスト、改修を意識した設計～

1. 研修要領

・募集定員	16名
・研修会場	オフィスメーション(確定)
・講師	富士通九州システムズ(FJQS)講師:井上 龍也氏
・開催月日	平成28年10月12日(水)～14日(金)
・実施時間・日数	●10月12日(水) 9:30～17:30(7時間/日) ●10月13日(木) 9:30～17:30(7時間/日) ●10月14日(金) 9:00～17:00(7時間/日) 計3日間(21時間)
・受講料(税別)	78,800円
・教材料(税別)	5,000円

2. 対象者

オブジェクト指向言語を用いて設計作業、開発作業を行っている方。また、オブジェクト指向開発を行ったが、QCDに結びつかなかった経験をお持ちの方。

3. カリキュラムの概要

Java言語やVisual Basic、C#などのオブジェクト指向言語で開発しているのに、開発効率が上がらない、メンテナンス効率が悪いと言うプロジェクトはありませんか。オブジェクト指向言語を使用しているも、オブジェクト指向の本質が分からないままに、何となくコードを慣習で記述しているプロジェクトが多いようです。本コースでは、オブジェクト指向の本質を理解して頂き、コードを記述する前の設計に焦点を当て、正しいオブジェクト指向設計とは何かを習得して頂きます。本コースを受講することによって、正しいオブジェクト指向設計を習得して頂き、同じコードを記述させないテクニック、部品化のノウハウ、メンテナンスしやすいアプリケーション構造を設計できるようになります。オブジェクト指向開発の品質、コスト、納期でお悩みの方にお奨めのコースです。

4. カリキュラムの詳細

3日間(21時間)

科目	時間	科目の内容
1. 問題のあるクラス設計の例	3.0 Hr	<ul style="list-style-type: none"> ・クラス設計の失敗例: 巨大な何でもクラス ・クラス設計の失敗例: 深すぎる継承 ・クラス設計の失敗例: 似て非なるクラス ・クラス設計の失敗例: メンバ変数の乱用 ・クラス設計の失敗例: ロジックの漏れ ・クラス設計の失敗例: インタフェースの不揃い 【演習】グループディスカッション
2. クラス設計に必要なオブジェクト指向の基礎知識	3.0 Hr	<ul style="list-style-type: none"> ・カプセル化とデータ隠蔽 ・継承の正しい使い方 ・継承と抽象化 ・継承とポリモフィズム ・クラス間の関係性(継承、関連、集約、依存) ・継承と関連、集約の使い分け ・変更に強いメソッド(操作)の外部仕様 ・正しいクラス設計のメリット (品質向上、コスト削減、再利用性、拡張性、保守性の向上)
3. クラス設計の原則	4.0 Hr	<ul style="list-style-type: none"> ・1 fact in 1 place (1つの事実は1つの場所に) ・システム構成要素を元にしたクラス分割 ・クラス分割の方針(多重度、継承、集約、システム境界) ・コールドポイントとホットポイントの分離 ・GoFのデザインパターンを使用したコード記述量削減法 ・Open-Close Principal ・拡張に対しオープンなクラス設計とは ・修正に対しクローズな設計とは 【演習】

4. 総合演習2	3.0 Hr	【問題のあるクラス設計の例の修正方法を検討する】 ・クラス設計の失敗例：巨大な何でもクラス ・クラス設計の失敗例：深すぎる継承 ・クラス設計の失敗例：似て非なるクラス ・クラス設計の失敗例：メンバ変数の乱用 ・クラス設計の失敗例：ロジックの漏れ ・クラス設計の失敗例：インタフェースの不揃い
5. 正しいクラスの分析・設計技法	3.0 Hr	・クラスを業務機能(ユースケース)から見つけるテクニック ・クラスに属性や操作を入れる(初期クラス図作成) ・クラスの役割を決める ・クラスの状態の有無、あれば状態遷移を分析する (ステートチャート図作成) ・クラスにどの業務ロジックを持たせるか (シーケンス図作成) ・クラスの属性、操作、クラス間の関係を決定する (クラス図作成)
6. 総合演習2	5.0 Hr	【あたえられた業務機能からオブジェクトを設計する】 あたえられた業務機能からオブジェクトを抽出し、その属性、操作、関係を決定する。その過程でシーケンス図、ステートチャート図、クラス図を作成する。
計	21.0 Hr	

5. 使用教材

オブジェクト指向を活用したクラス設計技法(富士通九州システムズ)
 オブジェクト設計演習(富士通九州システムズ)

6. 到達目標

本コース修了後、次の事項ができることを目標としています。

1. 間違ったクラス設計例の問題点を理解できる。
2. 正しいクラス設計により品質、再利用性、拡張性、保守性が向上、コスト削減が可能なことを理解できる。
3. クラス設計の正しい技法や思想を用いて間違ったクラス設計例を修正できる。
4. クラス設計の正しい手順(分析～詳細設計)を理解する。
5. 与えられた業務機能からクラスを抽出し、詳細化してクラス設計ができる。

7. レベル

ITSS:ITスペシャリスト育成 - [*]テクノロジー【レベル:2-3】

ITSS:アプリケーションスペシャリスト育成 - [*]テクノロジー【レベル:2-3】

[*] ITスキル標準研修ロードマップにおけるコース群名