

冬期インターン課題 - アプリ編 -



株式会社オリエンタルインフォメーションサービス

はじめに プロとアマチュアの違いとは



インターンの目的

- I. システム開発業務における
「プロ」と
「アマチュア」のちがいを理解する
- II. 「設計」「コーディング」の
基礎技術を修得する



プロとアマチュアの違いとは

1. 品質の追求
2. 時間のかけどころ
3. 可読性
4. 期待しない条件下での動作



プロとアマチュアの違いとは

1. 品質の追求

- **バグ**
 - ・ 正常に動作するのは当たり前
 - ・ 異常発生時の対応によって品質が決まる
- **パフォーマンス**
 - ・ 省メモリ
 - ・ 処理速度
- **拡張性**
 - ・ 仕様変更などにより、追加（削除）が発生
 - ・ ロジックを変えずに変更



プロとアマチュアの違いとは

2. 時間のかけどころ

- **設計**
 - ・ フローチャート、関数の構成・処理
- **実装**
 - ・ コーディング

まずは、設計で品質を高める

- 未然のバグ解消
- 仕様変更などに対して柔軟に対応が可能



プロとアマチュアの違いとは

2. 時間のかけどころ (状況に応じた方法が大切)

- **スマートフォンアプリ**
 - 早くリリース
 - 頻繁に改修・機能追加など
- **車・金融・医療システム**
 - バグ発生 = 大問題
 - 生命に関わる, 社会問題



プロとアマチュアの違いとは

3. 可読性

- **複数人が関わる**
 - レビュー・担当者の変更
 - 誰が読んでも理解出来る
- **再現性**
 - バグ対応・実装変更
 - 数ヶ月、数年前に実装したものの変更
 - (自分で実装したものでも、
期間が空くと別人が書いたものと同じに)



プロとアマチュアのの違いとは

3. 可読性の上昇(一例)

- 関数・変数の命名規則

- 誰が呼んでも理解できるように

- コーディング規約

- コメント
- プレフィックス
- インデント



プロとアマチュアのの違いとは

3. 可読性

- 綺麗なコードを書くように意識する

```
for(i=0; i<5; i++) {  
  if(i<3) {  
    for(j=0; j<5; j++) {  
      if(j==0) {  
      }  
    }  
  }  
  else if(j!=1) {  
  }  
}  
else {  
}  
}
```

```
for(i=0; i<5; i++) {  
  if(i<3) {  
    for(j=0; j<5; j++) {  
      if(j==0) {  
      }  
    }  
  }  
  else if(j!=1) {  
  }  
}  
else {  
}  
}
```



プロとアマチュアのの違いとは

4. 期待しない条件下での動作

- フォールセーフ

```
switch(pedal) {  
  /* クラッチ */  
  case Clutch:  
    doClutching();  
    break;  
  
  /* ブレーキ */  
  case Brake:  
    speedDown();  
    break;  
  
  /* アクセル */  
  default:  
    speedUp();  
    break;  
}
```

```
switch(pedal) {  
  /* クラッチ */  
  case Clutch:  
    doClutching();  
    break;  
  
  /* アクセル */  
  case Accel:  
    speedUp();  
    break;  
  
  /* ブレーキ */  
  default:  
    speedDown();  
    break;  
}
```



インターン課題
アプリ概要



概要

【今回作成するアプリについて】

1. ECサイトのネイティブアプリを製作する。
2. ウォーターフォールモデルでの、内部設計・プログラミング・テストを実施する。



【補足】ウォーターフォールモデル

I. 要件定義

- 開発するアプリケーションの機能と目的、対象範囲のまとめる工程。

II. 外部設計（基本・機能設計）

- 製作するアプリのシステム要件・業務要件をまとめる工程。

III. 内部設計（詳細設計）

- システムの構造や仕様をプログラム単位に分割し、動作を定義していく工程。

IV. 開発（プログラミング）

- 設計書をもとに実装（プログラミング）を行う工程。

V. テスト

- 実装されたプログラムの動作確認を行う工程。



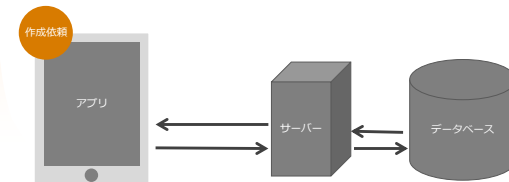
インターネット課題 要件定義



I. 要件定義

● アプリの概要

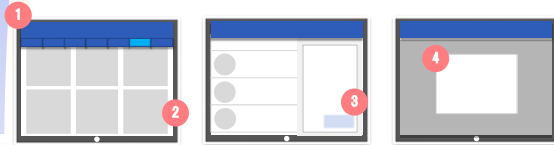
ECサイトのネイティブアプリを開発する。



I. 要件定義

● 商品の購入の流れ

1. 商品を検索。
2. 欲しい商品をカートに入れる。
3. カートに入っている商品を確認する。
4. 購入の確認を行い、購入完了。



I. 要件定義

● 清算の条件

- 消費税は8%
- 税込3000円以上で送料無料
- 3000円未満は送料として350円を税込価格に加算
- 税抜合計の100円毎にポイントが1ポイントずつ貯まる。
- ポイントは1000ポイントで税抜き価格から1000円分の割引として使える。
- 毎月20,30日は税抜き価格から5%OFF



インターネット課題

外部設計（画面設計）



II. 外部設計（画面設計）

【開発環境】

● 対象端末

- Android 5.0/5.1 Lollipop 以上

● 実装

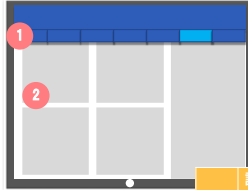
- 使用言語 : Java
- 開発環境 : Android Studio



Ⅱ. 外部設計（画面設計）

● 商品一覧画面

商品一覧



商品枠内

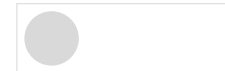
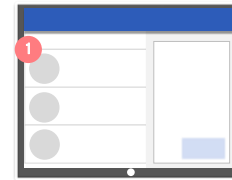


	論理名
1	タブ
2	商品枠
3	商品写真
4	商品名
5	購入ボタン



Ⅱ. 外部設計（画面設計）

● 購入確認画面

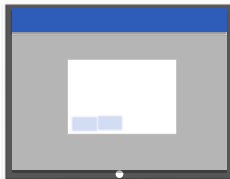


	論理名
1	タブ
2	商品枠
3	商品写真
4	商品名
5	購入ボタン



Ⅱ. 外部設計（画面設計）

● 購入確認ダイアログ



	論理名
1	キャンセルボタン
2	確定ボタン



インターネット課題
内部設計



II. 設計・コーディング技術

設計するにあたり必要な知識

- フローチャート書き方ルール
- MVCモデルによる開発



課題

【課題】

1. 清算のフローチャートを作成せよ
2. フローチャートに基づき、与えられたプログラムを変更せよ。



I. 要件定義

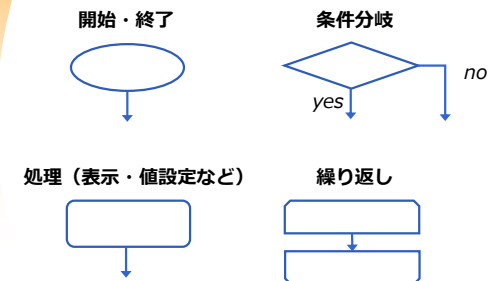
● 清算の条件

- 消費税は8%
- 税込3000円以上で送料無料
- 3000円未満は送料として350円を税込価格に加算（ポイント使用時は加算なし）
- 税抜合計の100円毎にポイントが1ポイントずつ貯まる。
- ポイントは1000ポイントで税抜き価格から1000円分の割引として使える。
- 毎月20,30日は税抜き価格から5%OFF



II. 設計・コーディング技術

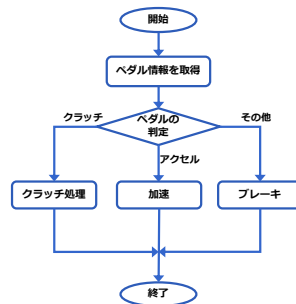
● フローチャート



II. 設計・コーディング技術

【フローチャートの描き方】

● 車のペダルのシステム



I. 要件定義

● MVCモデルによる開発

本アプリは、フレームワークとして MVCモデルを採用している。

● Model

データを定義する。(変数の集まり)

● View

ボタンや文字表示などを管理する。

● Controller

データの受け渡しを管理する。



インターン課題 実装 (プログラミング)



I. 要件定義

● 無料会員の清算ロジックの変更

パッケージ

ois.internship.controller.logic.billCalc;

クラス

FreeMemberLogic

コンストラクタ

FreeMemberLogic()



インターン課題 テスト



課題（仕様変更）

【課題（仕様追加）】

仕様変更で従来の無料会員に追加して、
有料会員の登録区分が追加された。

新たに追加する有料会員の清算は
次のスライドの内容とする。



I. 要件定義

● 有料会員の清算の条件

- 消費税は8%
- 送料は常時無料。
- 税抜合計の100円毎にポイントが
2ポイントずつ貯まる。
- ポイントは1000ポイントで1000円分の
割引として使える。
- 毎月20,30日は税抜き価格から10%OFF



課題（仕様変更）

【課題（仕様変更）】

1. 清算のフローチャートを作成せよ
2. フローチャートに基づき、
与えられたプログラムを変更せよ。



インターン課題 実装 (プログラミング)



I. 要件定義

- 有料会員の清算ロジックの変更

パッケージ

ois.internship.controller.logic.billCalc;

クラス

PayMemberLogic

コンストラクタ

PayMemberLogic()



インターン課題 テスト

