

プログラム設計

<http://bit.ly/design4d>

ユースケース図

後期 第11週

2019/12/9

今回学ぶダイアグラム

ユースケース図

- ▶ システムを利用する**ユーザ**がシステムに**要求する機能**を表現する
- ▶ システム開発の**初期段階**で利用し、開発するシステムの機能を明確にするために作成する

アクタ

システムと**相互に作用**（情報をやり取り）する要素

【例】 システムの利用者、外部のハードウェア、
既存のシステム、など



通常は人型で表記する

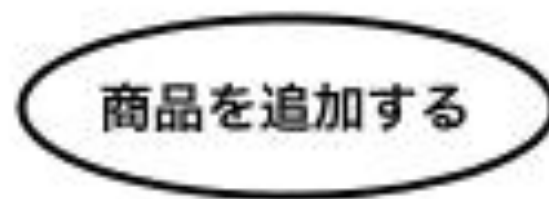


ステレオタイプ（クラスの性質などにより分類するための機能）で「<<actor>>」を描けば、クラスとしても表記できる

ユースケース

利用者から見たシステムの機能を表現する

（「システムをどのように実現するのか？」というシステム内部の視点ではなく
「システムをどのように利用したいのか？」というシステム外部の利用者の視点
で表現する）



楕円の中（または楕円の下）に
ユースケース名を書いて表記する



ユーザがシステムに対して期待している機能 1つをユースケース1つとする。
1つのユースケースで利用者の目的が達成される程度の機能を1つとする。

ユースケース記述

ユースケースの詳細な**処理手順**（シナリオ）を記述する。書き方に標準は用意されていないが、文章や箇条書きを用いて、流れが分かるように表記する。

ユースケース記述でよく使われる項目

- ▶ 目的（概要）・・・ユースケースで実現される機能の説明
- ▶ 事前条件・・・ユースケースが開始できる状態についての条件
- ▶ 事後条件（保証）・・・ユースケースが終了した状態についての条件
- ▶ 基本系列（シナリオ）・・・ユースケースの主な処理の流れ
- ▶ 例外系列（拡張）・・・シナリオにおいて、正常に終了しない場合や、別の処理として拡張する場合の処理の流れ

ユースケース記述の例

項目	内容
ユースケース	商品を追加する
概要	追加処理
アクター	店員
事前条件	追加画面が表示されている
事後条件	データベースに商品が追加されている
基本系列	商品情報を入力する 追加ボタンを押す (E-1) 商品一覧を確認する 終了ボタンを押す
例外系列	E-1: 追加失敗、エラーメッセージを表示する

ユースケースの関係：関連

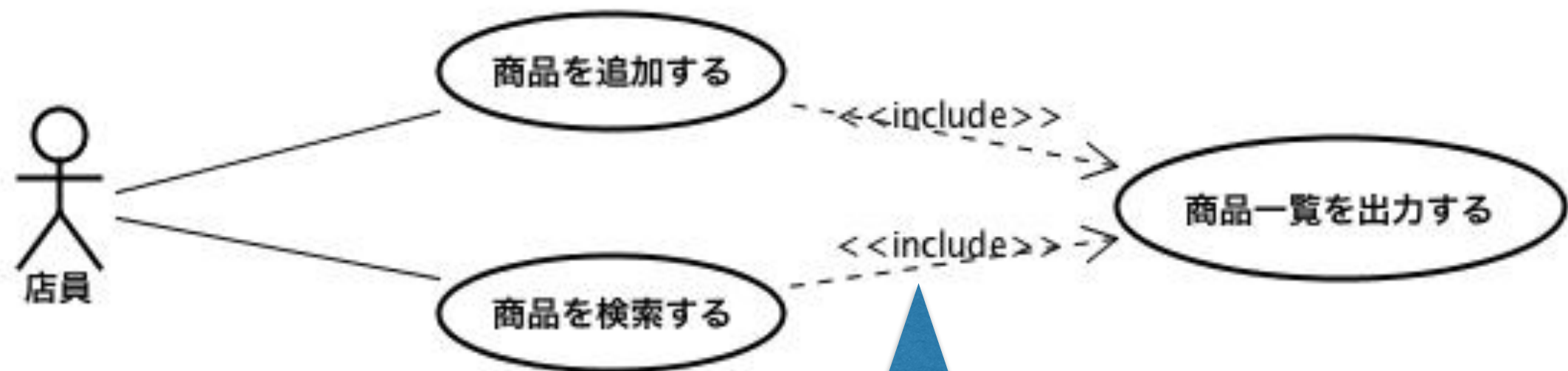
アクタが**利用するユースケース**との対応付けを表す



アクタと対応するユースケースを実線でつなぐ

ユースケースの関係：包含

ユースケース同士で共通部分がある場合、**共通部分**を新たなユースケースとして作成し、ユースケース同士を**包含関係**で表現する。



ユースケース同士を依存関係（点線の矢印）でつなぎ、ラベル「<<include>>」を付ける

※ユースケース同士の関係には、「包含」の他に「拡張」「凡化」などがある。

【課題の準備】

演習室で作業する前に、以下のコマンドを
入れるだけで準備が完了する

```
$ mygitclone4d 「自分のGitHubユーザ名」  
$ cd prog4d-(ユーザ名)  
$ ./myconf
```

※本体をシャットダウンするまでは、
上記「mygitclone」と「myconf」の設定は有効です

【課題の準備】

以下の流れで、課題のプログラムを作るためのフォルダを準備しましょう。

1. 端末を起動して、以下のコマンドを実行して後期第11週のフォルダを作る
\$ cd prog4d-(ユーザ名) (←既に移動しているなら不要)
\$ mkdir week211
\$ cd week211

※課題で作るファイル名は各自で決めて構いません。

【練習11-1】

astahを使って、スライドで示したユースケース図の
関連と包含の例を描いてみましょう。

【練習11-2】

astahを使って、スライドで示したユースケース記述の例を入力してみましょう。

【課題11-1】

「携帯電話の移行に関する」システムについて、
以下のような機能を考えます。

- ▶利用者は、電話番号と FeliCaデータを移行できる
- ▶店員は、電話番号を移行できる
- ▶店員は、電話番号を変更または削除できる

アクタを「利用者」「店員」とし、ユースケースを「電話番号を移行する」「FeliCa を移行する」「電話番号を変更する」「電話番号を削除する」とした場合のユースケース図をastahで作成して下さい。

【課題の提出】

以下の流れで、GitHubにプッシュしてWebサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題11-1提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

【課題11-2】

課題11-1のユースケース「電話番号を変更する」に対して、以下のユースケース記述を追加して下さい。

項目	内容
ユースケース	商品を追加する
概要	変更処理
アクター	店員
事前条件	変更画面が表示されている
事後条件	番号が変更されている
基本系列	利用者情報を入力する (E-1) 変更前と変更後の番号を入力する (E-2) 変更ボタンを押す 変更結果を確認する 終了ボタンを押す
例外系列	E-1: 情報が一致しない、再入力する E-2: 変更失敗、エラーメッセージを表示する

【課題の提出】

以下の流れで、GitHubにプッシュしてWebサイトで確認してみましょう。

1. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m “課題11-2提出”
```

```
$ git push origin master
```

2. 自分のリポジトリを開いて、提出したファイルがプッシュされているか確認する

[https://github.com/nit-ibaraki-prog4d-2019/prog4d-\(ユーザー名\)](https://github.com/nit-ibaraki-prog4d-2019/prog4d-(ユーザー名))

今回の小テスト

第10週の内容で小テストを実施します。