PlantUMLを通じてアクティビティ図の書き方を学ぶ

はじめに

ソフトウェアの仕様書、設計書の作成や管理を効率化するために、Markdown + PlantUMLによる作成方法を日々模索しています。
しかしながら、そもそもUML図の正式な書き方というものをちゃんと分かっていないというのが正直なところなので、PlantUMLを通じてUMLの各種図の書き方を勉強していきます。

- 本稿のテーマは、「UMLによるアクティビティ図の書き方」です。
- 本稿におけるUML図の作成は、Markdown + PlantUMLがベースであることを前提とします。

目次

- PlantUMLを通じてアクティビティ図の書き方を学ぶ
  - はじめに
  - 目次
  - アクティビティ図とは
  - シーケンス図との違い
  - アクティビティ図の構成要素
    - 基本的な描き方
    - ラベル
    - デシジョン、マージノード(分岐)
    - フォーク、ジョイント
    - パーティション
  - 参考資料

アクティビティ図とは

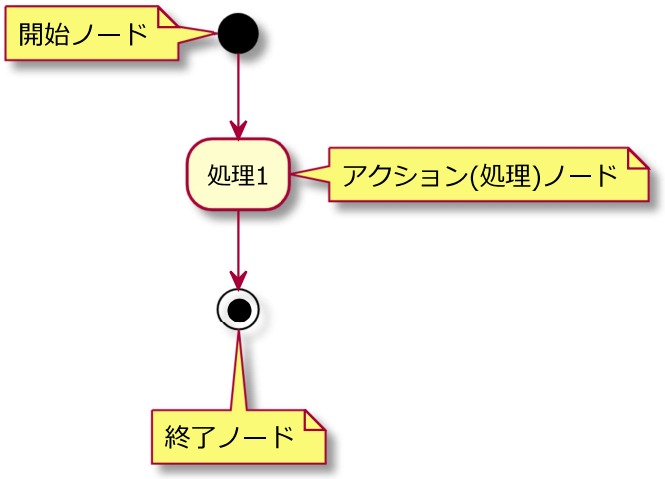
- システムや業務のアクティビティ、データの流れ、アクティビティ実施の条件分岐などを表現するもの。
- どのプロセスとどのプロセスが同時に走るか、走らせることが出来るかを見るのに良い。
- お客さんはプロセスレベルで考える。お客さんと業務の流れを話すときは、アクティビティ図を「業務の流れ」といって図示するのが良い。

シーケンス図との違い

- シーケンス図はメッセージが何処まで生きているかを書き込める。
- あるメッセージを投げているとき、他のリソースが同時並行して動くかどうかが表現できる。
- アクティビティ図よりも細かい情報を描くことが出来る。
- 細かいやり取りが必要な開発者間ではシーケンス図がいい。

アクティビティ図の構成要素

基本的な描き方

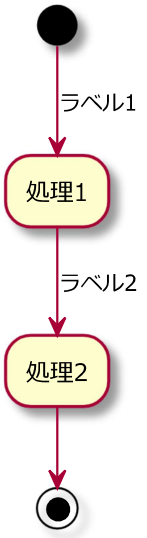


PlantUMLでの記述例

```
note left : 開始ノード

(*) --> "処理1"
note right
    アクション(処理)ノード
end note

"処理1" --> (*)
note bottom : 終了ノード
```

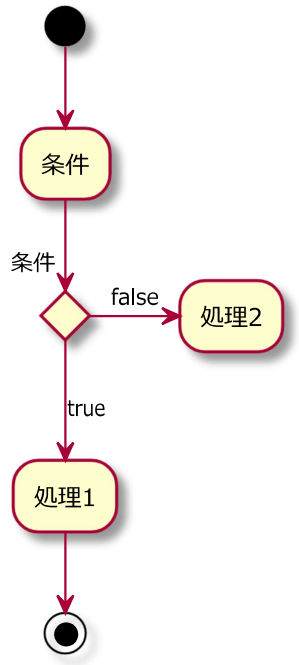
ラベル

```
(*) --> [ラベル1] "処理1"
"処理1" --> [ラベル2] "処理2"
"処理2" --> (*)
```



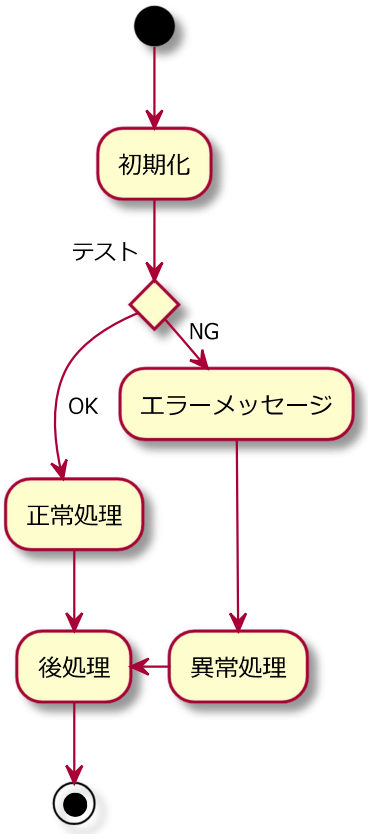
デシジョン、マージノード(分岐)

```
(*) --> 条件
if 条件 then
    -right-> [false] 処理2
else
    --> [true] 処理1
endif
処理1 --> (*)
```



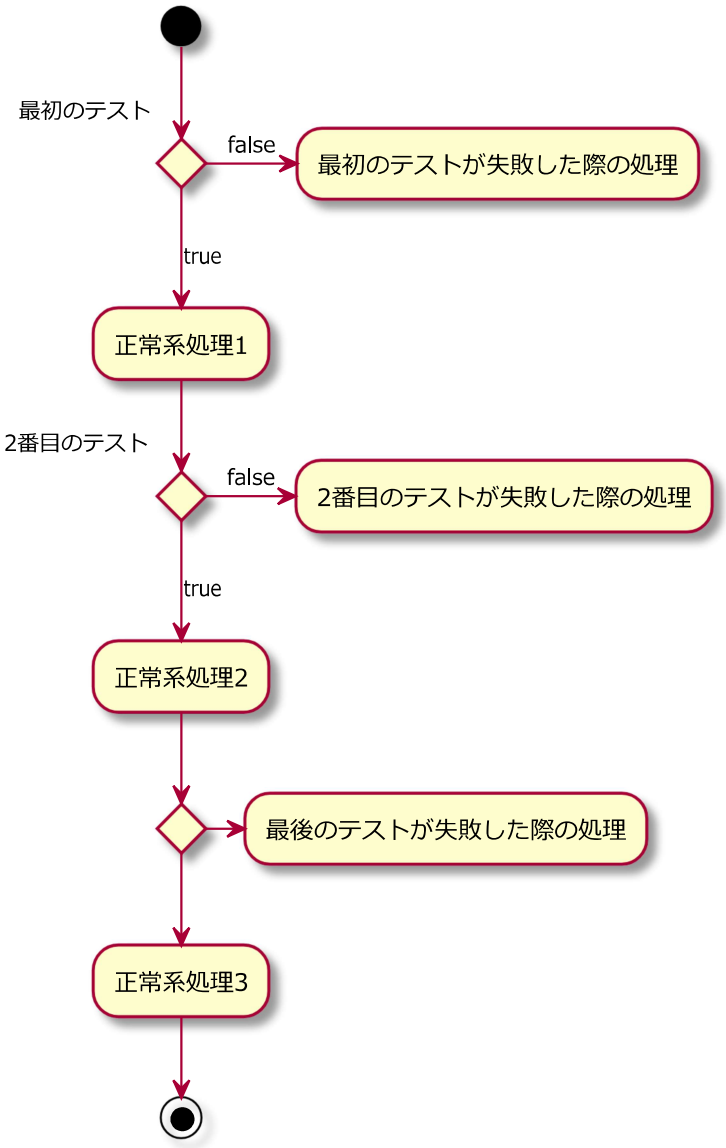
矢印の - の数が2の場合は下向き、1の場合は横向きの矢印となる。

```
(*) --> 初期化
if テスト then
    --> [OK] 正常処理
    --> 後処理
else
    -right-> [NG] エラーメッセージ
    --> 異常処理
    -left-> 後処理
endif
--> (*)
```



if文は入れ子にすることもできる。

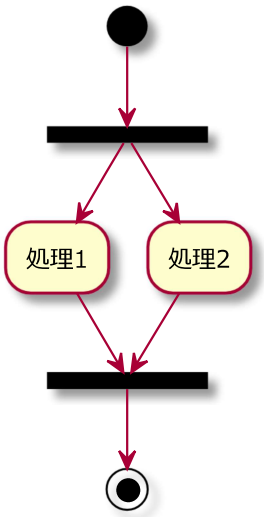
```
(*) --> if 最初のテスト then
    --> [true] 正常系処理1

    if 2番目のテスト then
        --> [true] 正常系処理2
    else
        -> [false] 2番目のテストが失敗した際の処理
    endif

else
    -> [false] 最初のテストが失敗した際の処理
endif

正常系処理2 --> if "" then
    --> 正常系処理3
    --> (*)
else
    -> 最後のテストが失敗した際の処理
endif
```



フォーク、ジョイント

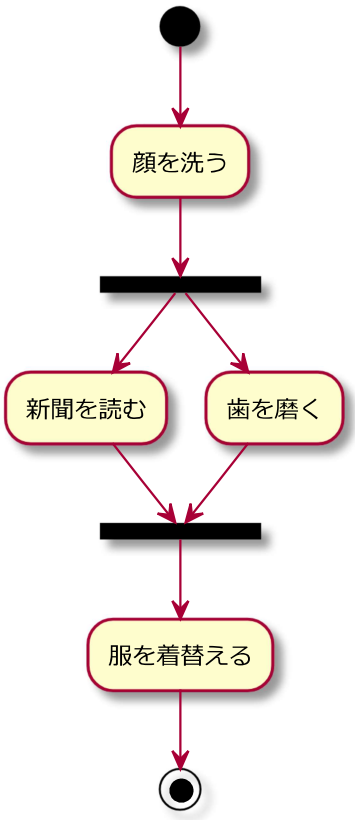
非同期に行う並列処理を表す。

```
(*) --> ===フォーク===
===フォーク=== --> 処理1
処理1 --> ===ジョイント===
===フォーク=== --> 処理2
処理2 --> ===ジョイント===
===ジョイント=== --> (*)
```



朝起きてから服を着替えるまでの流れを表現した例。

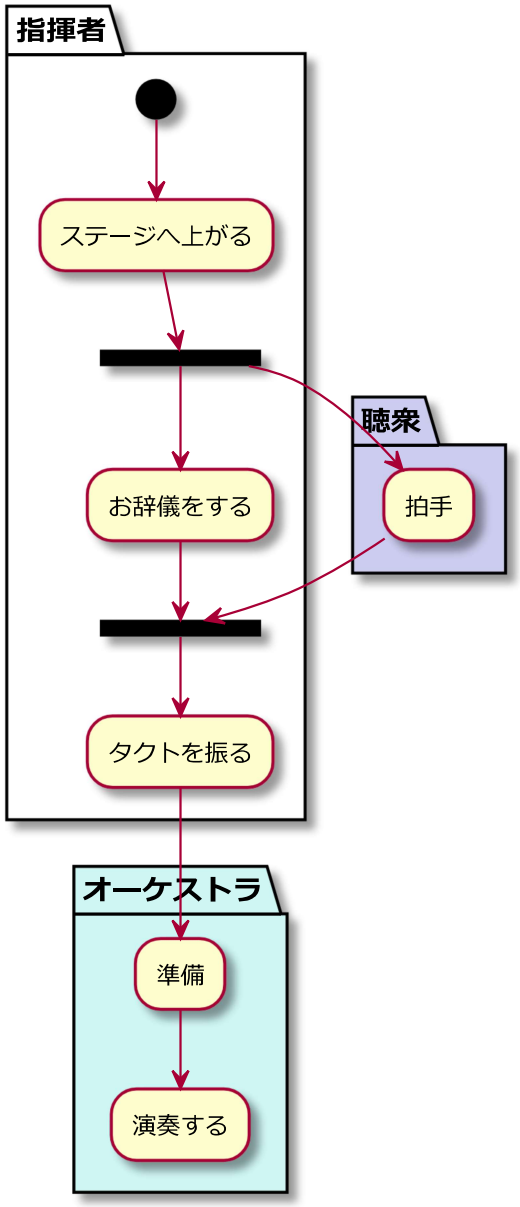
```
(*) --> 顔を洗う

--> ===フォーク===
--> 新聞を読む
--> ===ジョイント===

===フォーク=== --> 歯を磨く
--> ===ジョイント===

--> 服を着替える
--> (*)
```



パーティション

処理が行われる区画分けのために、パーティション名をラベル表記して四角で囲む。

```
partition 指揮者 {
    (*) --> ステージへ上がる
    --> ===Sf===
    --> お辞儀をする
}

partition 聴衆 #CCCCEE {
    ===Sf=== --> 拍手
}

partition 指揮者 {
    お辞儀をする --> ===Sj===
    --> タクトを振る
    拍手 --> ===Sj===
}

partition オーケストラ #CEF6F5 {
    タクトを振る --> 準備
    --> 演奏する
}
```



参考資料

- 【新人教育 資料】第7章 UMLまでの道 ～アクティビティ図の説明&書いてみよう編～
- プログラマーズ雑記帳 PlantUML - アクティビティ図
- ウィリアムのいたずらの開発日記