

# オブジェクト指向技術 第15回

## — 例題 —

立命館大学 情報理工学部  
丸山 勝久

[maru@cs.ritsumeai.ac.jp](mailto:maru@cs.ritsumeai.ac.jp)

# オブジェクト指向モデル

---

## ■ 概念モデル

- ドメイン分析(業務分析)により作成
- 対象業務の世界を構成する概念と概念間の関係を表すモデル

## ■ 要求モデル

- 顧客がシステムに望む事柄を表すモデル

## ■ 分析モデル

- 実装方法に関知せずに、対象業務についてシステム化する方法を表すモデル

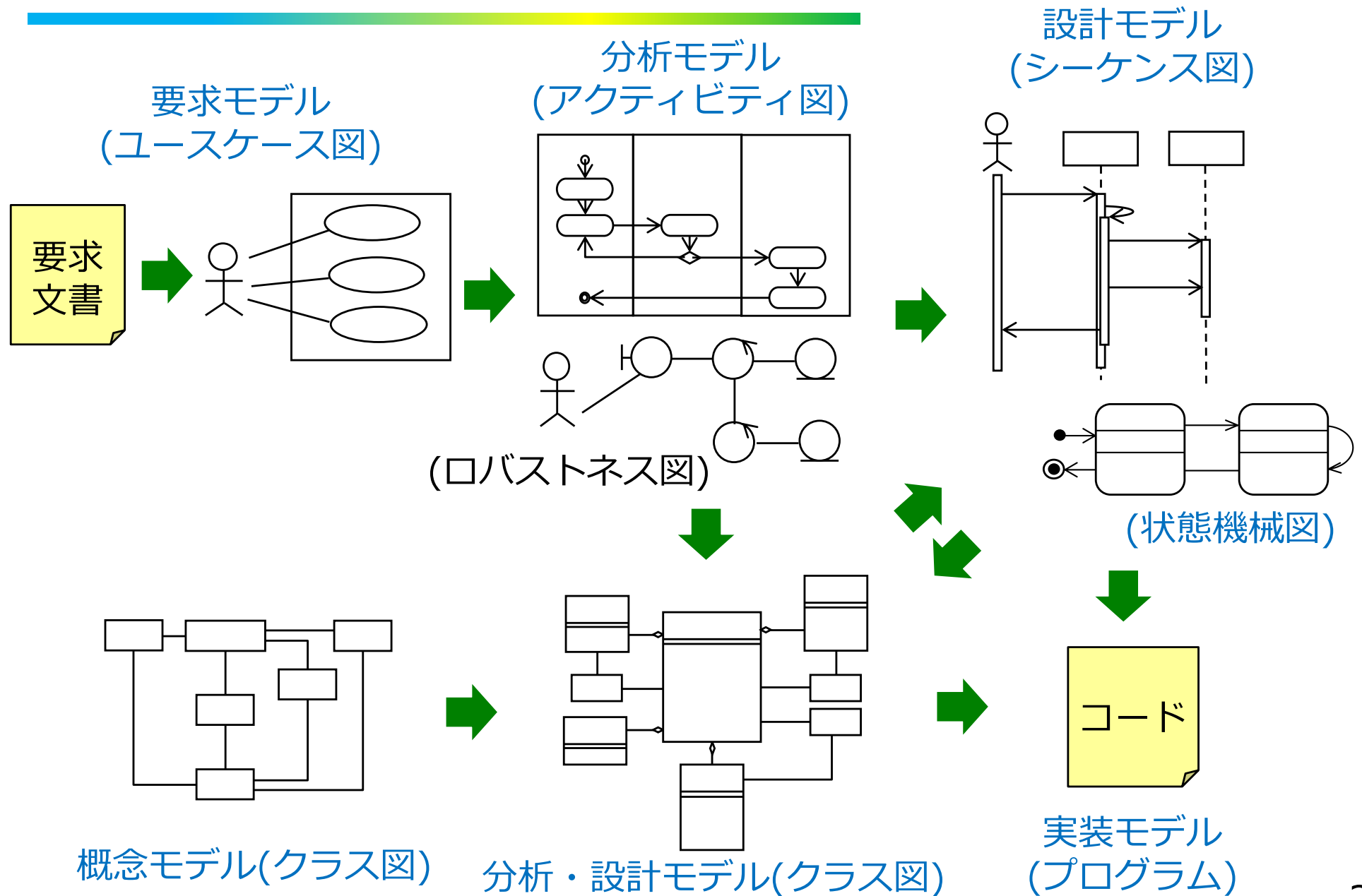
## ■ 設計モデル

- システム化する事柄と、その実装方法を詳細に表すモデル

## ■ 実装モデル

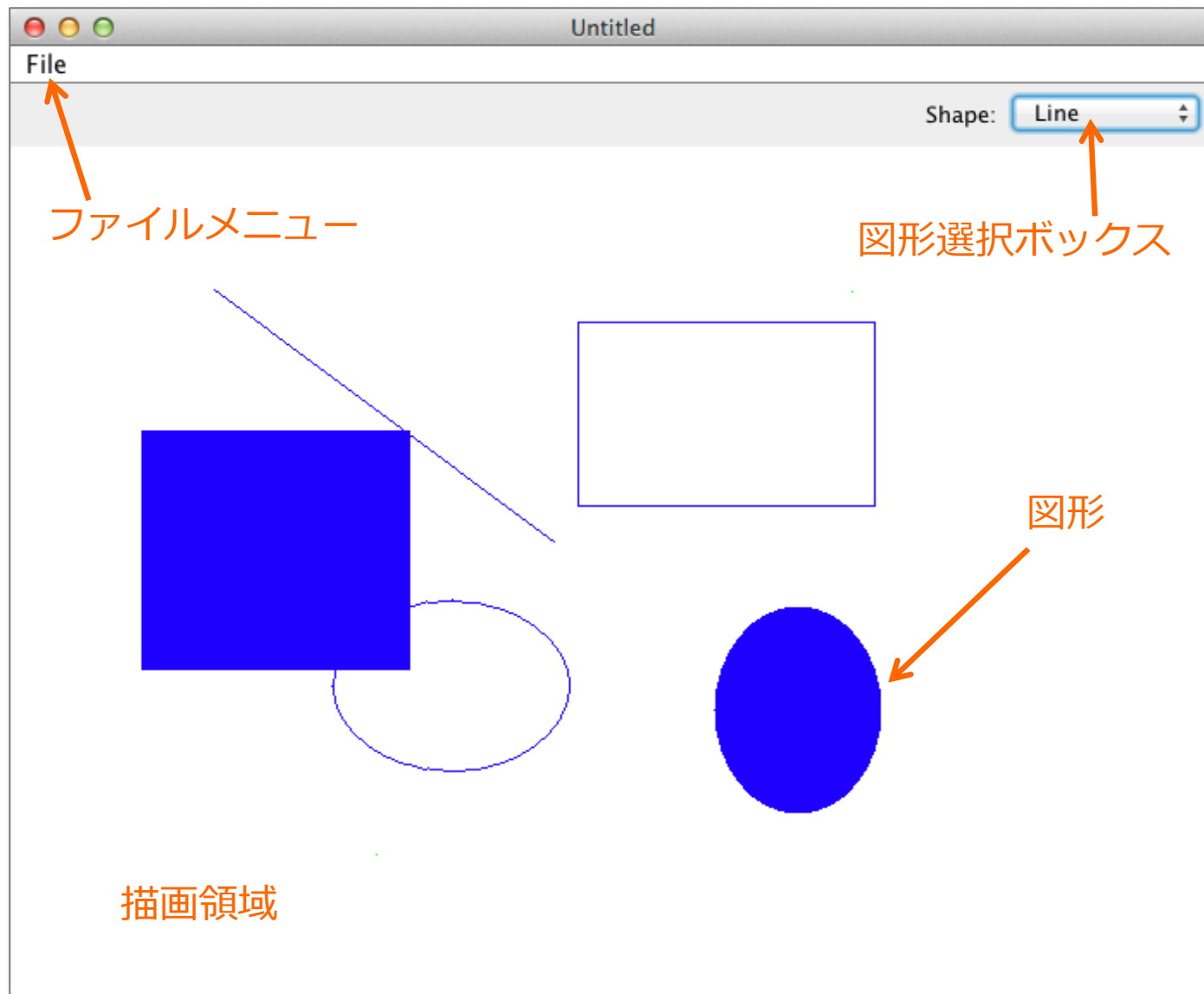
- プログラムソースコード, バイナリプログラム
- プログラムの計算機上の配置

# オブジェクト指向モデリング



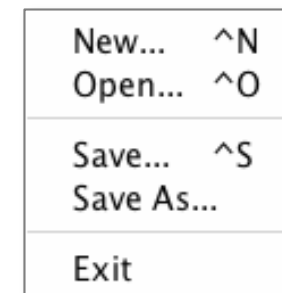
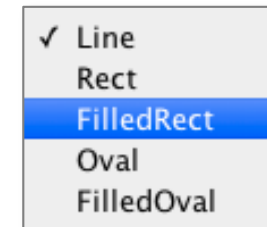
# 例題1：図形描画ツール

# 図形描画ツール(画面イメージ)

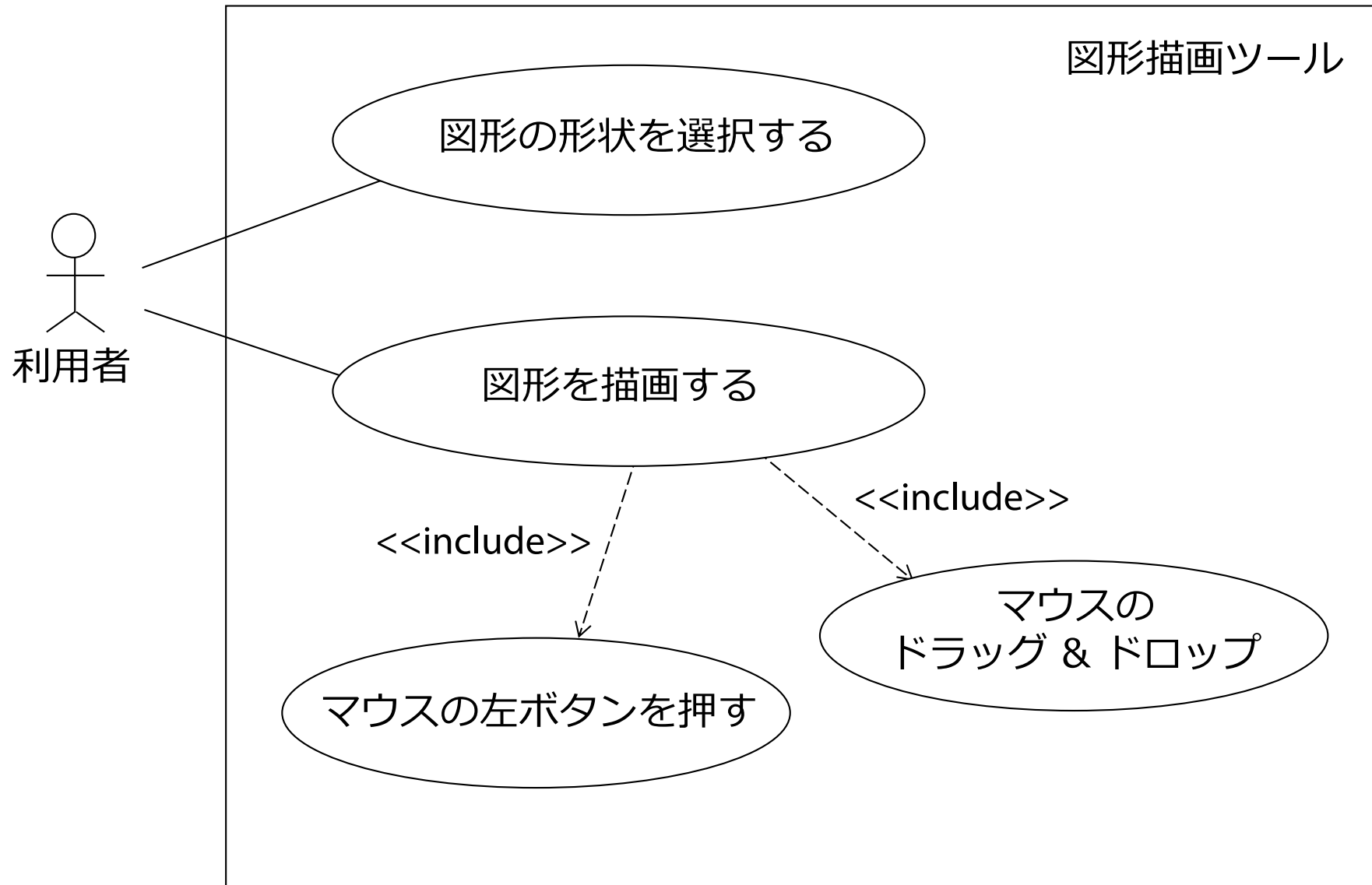


# 図形描画ツール(要求文書)

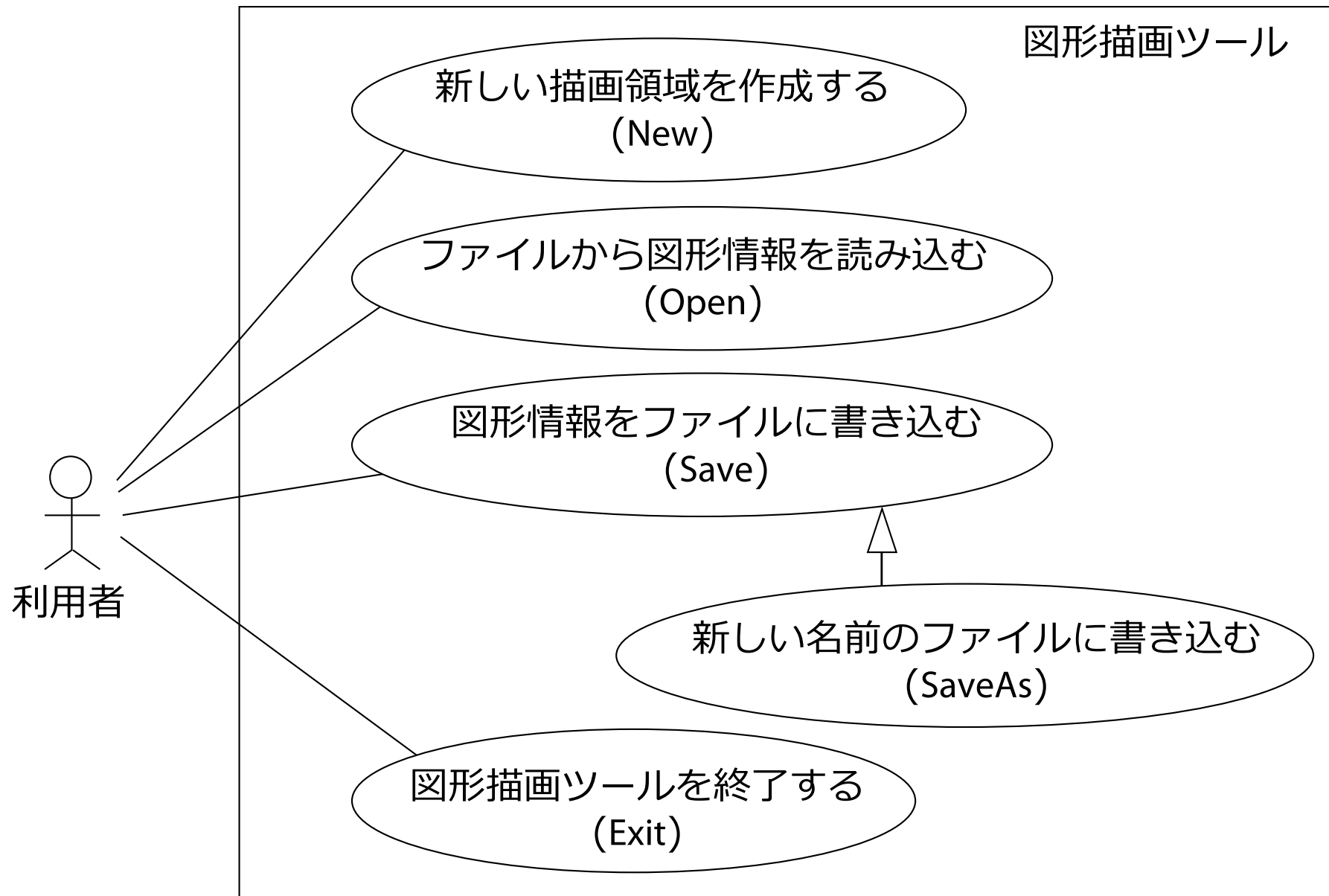
- 図形描画ツール(Drawing Tool)は描画領域(Canvas)を持つ
- 利用者(User)は, マウス(Mouse)を用いて描画領域に図形(Figure)を描くことができる
  - ◆ マウスの左ボタンを押すことで図形の始点を指定し, ドラッグ(マウスを移動) & ドロップ(左ボタンを放す)することで図形の終点を指定する. 描画中の図形はラバーバンドで表現される
- 描画する図形の形状は, 図形選択ボックス(Figure Selector)で選択する
  - ◆ 描画可能な図形は, 線, 長方形, 楕円である
- 利用者は, ファイルメニュー(File Menu)から選択することで描画した図形をファイル(File)から読み込んだり, ファイルに書き込んだりすることができる
  - ◆ 新しい描画領域を作成する(New)
  - ◆ ファイルから図形情報を読み込む(Open)
  - ◆ 図形情報をファイルに書き込む(Save)
  - ◆ 新しい名前のファイルに書き込む(SaveAs)
  - ◆ 図形描画ツールを終了する(Exit)



# 要求モデル(ユースケース図 1/2)



# 要求モデル(ユースケース図 2/2)





# 要求モデル(ユースケース記述 1/2)

<b>名前</b>	図形を描画する
<b>アクタ</b>	利用者
<b>事前条件</b>	描画する図形の形状が選択されている
<b>事後条件</b>	描画された図形が描画領域に配置されている 描画された図形が記録されている

## 基本フロー

1. 利用者が、描画領域の上でマウスの左ボタンを押すことで、図形の始点を指定する
2. 図形描画ツールが、図形を作成する
3. 利用者が、マウスをドラッグ(マウスを移動) & ドロップ(左ボタンを放す)することで、図形の終点を指定する。描画中の図形はラバーバンドで表現される
4. 図形描画ツールが、描画された図形を描画領域に配置する
5. 図形描画ツールが、描画された図形を記録する

## 代替フロー

3. ドラッグせずにドロップした場合、図形の描画を中止する

# 要求モデル(ユースケース記述 2/2)

<b>名前</b>	ファイルから図形情報を読み込む
<b>アクタ</b>	利用者
<b>事前条件</b>	なし
<b>事後条件</b>	読み込んだ図形が描画領域に配置されている 読み込んだ図形が記録されている

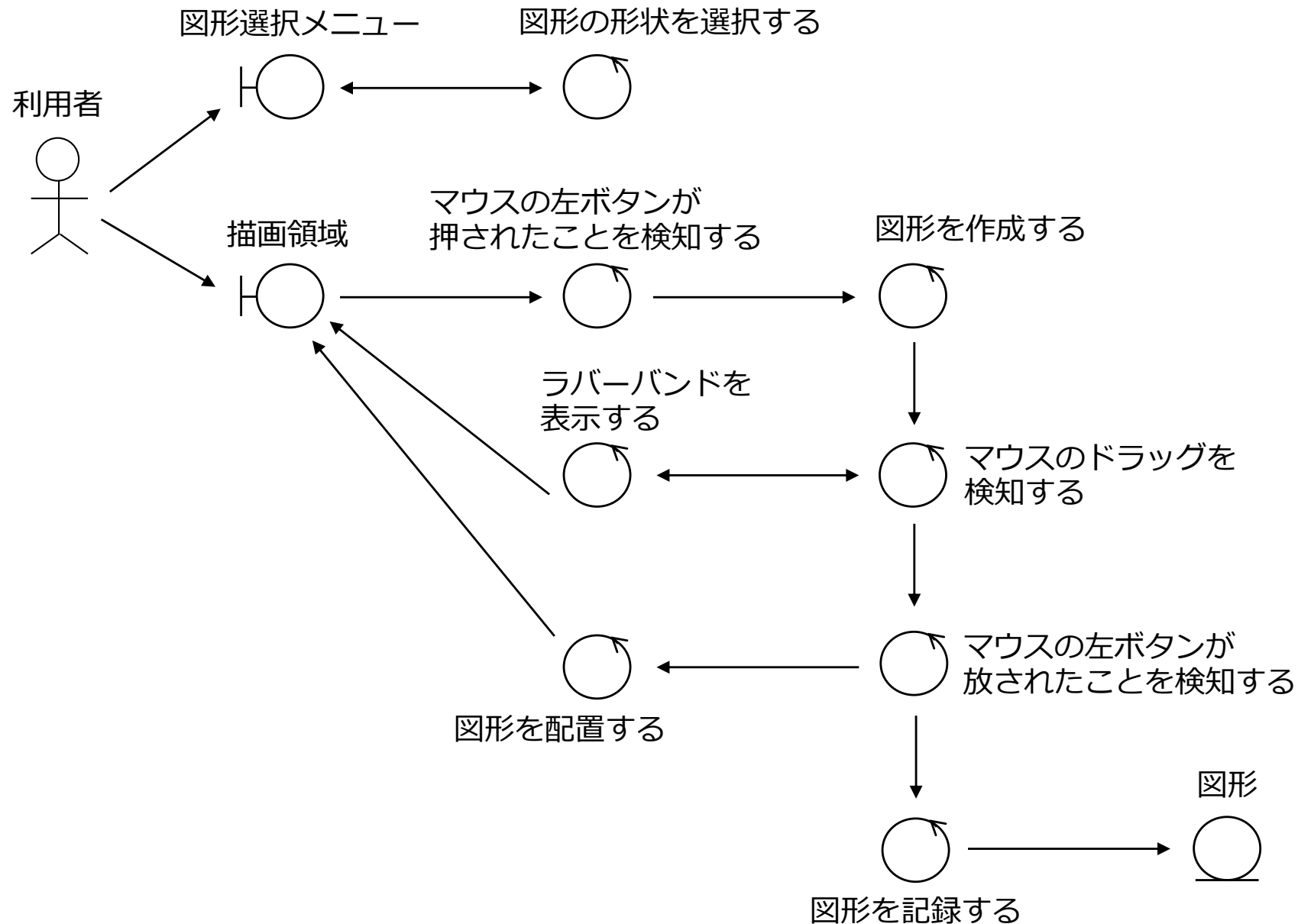
## 基本フロー

1. 利用者が、ファイルメニューで“Open”を指定する
2. 図形描画ツールが、ファイル選択ダイアログを表示する
3. 利用者が、ファイルを指定する
4. 図形描画ツールが、ファイルから図形情報を読み込む
5. 図形描画ツールが、読み込んだ図形を描画領域に配置する
6. 図形描画ツールが、読み込んだ図形を記録する

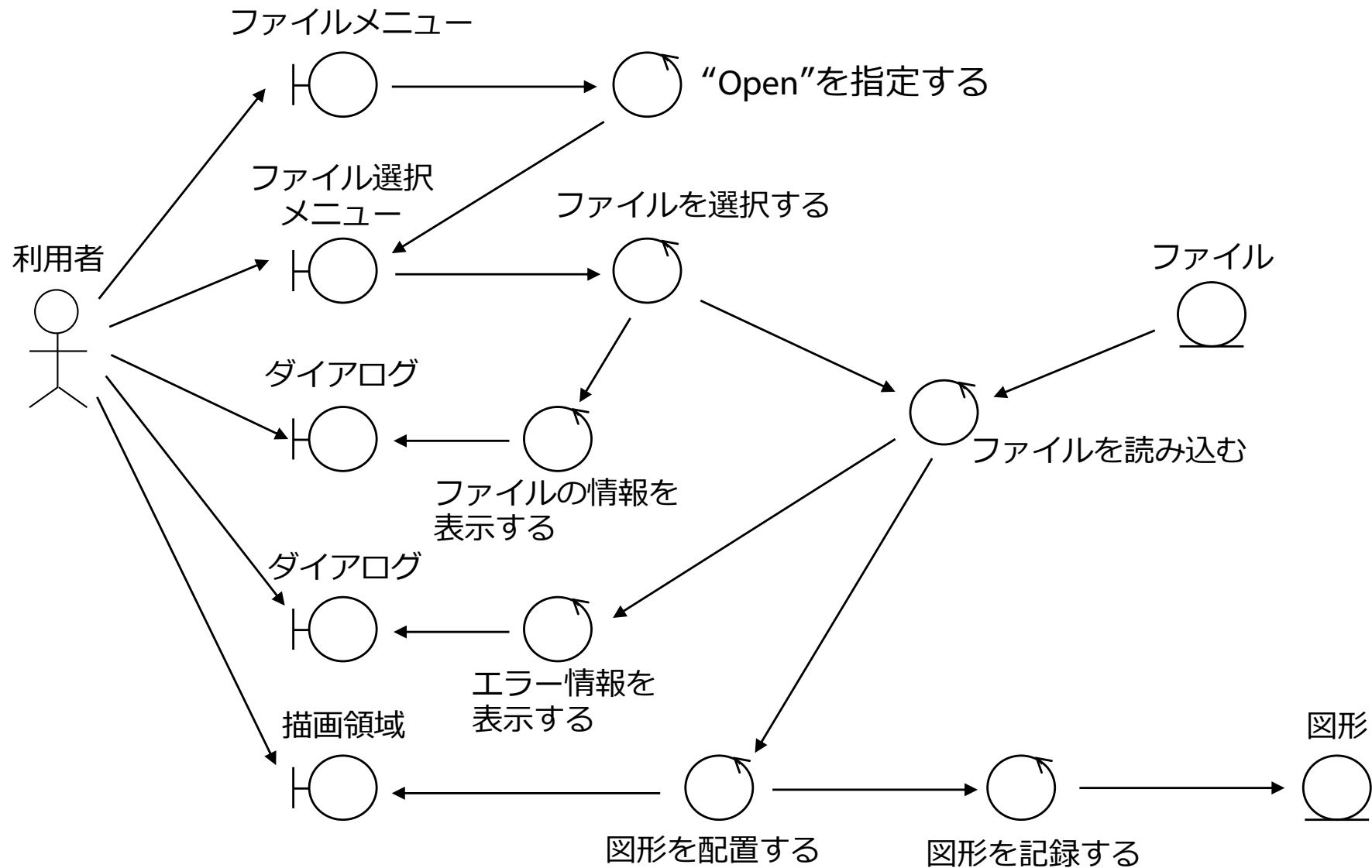
## 代替フロー

3. 描画領域を保存するファイルと同じ名前のファイルを指定した場合, ...
4. 読み込みが失敗した場合, ...

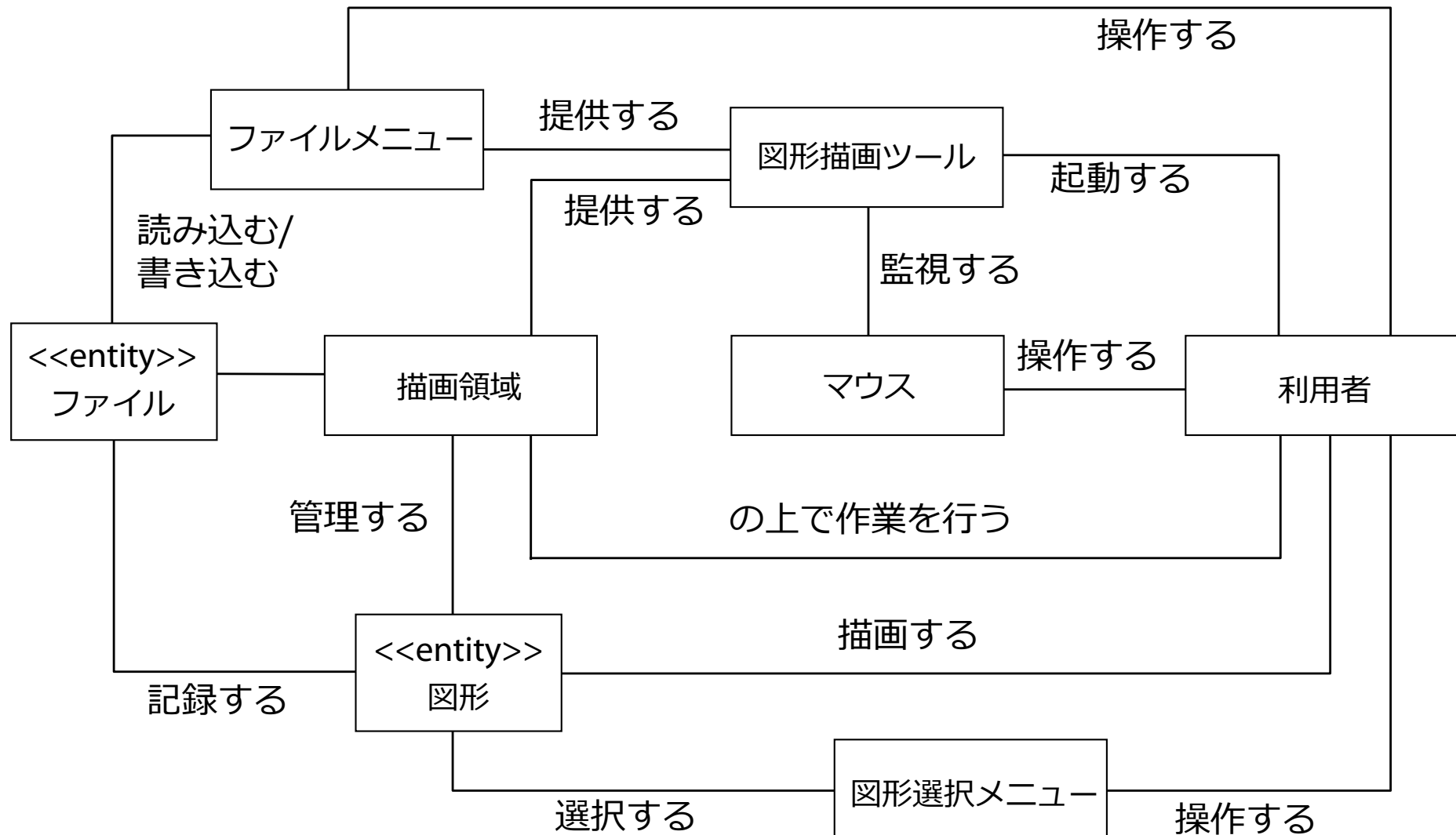
# 分析モデル(ロバストネス図 1/2)



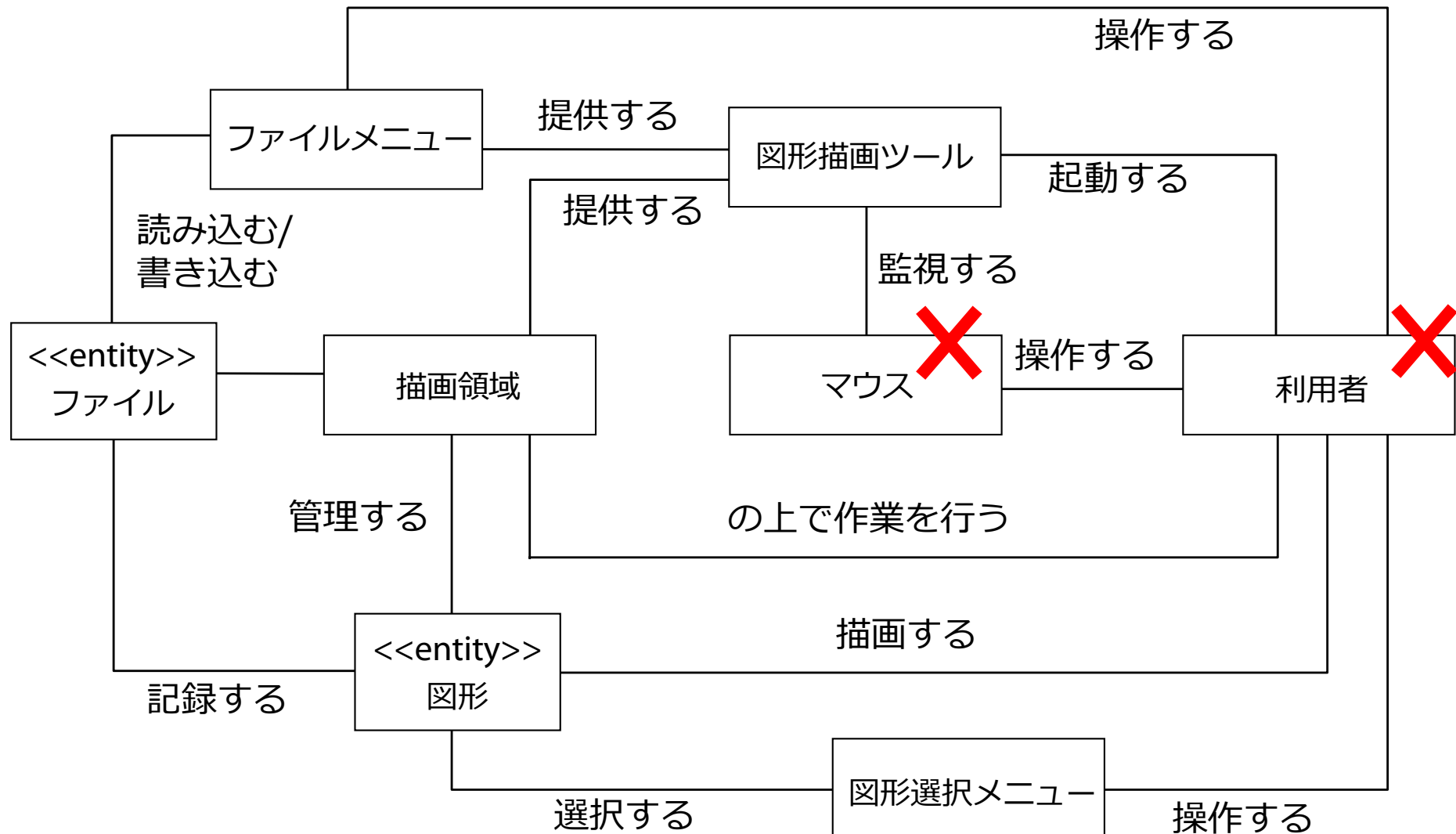
# 分析モデル(ロバストネス図 2/2)



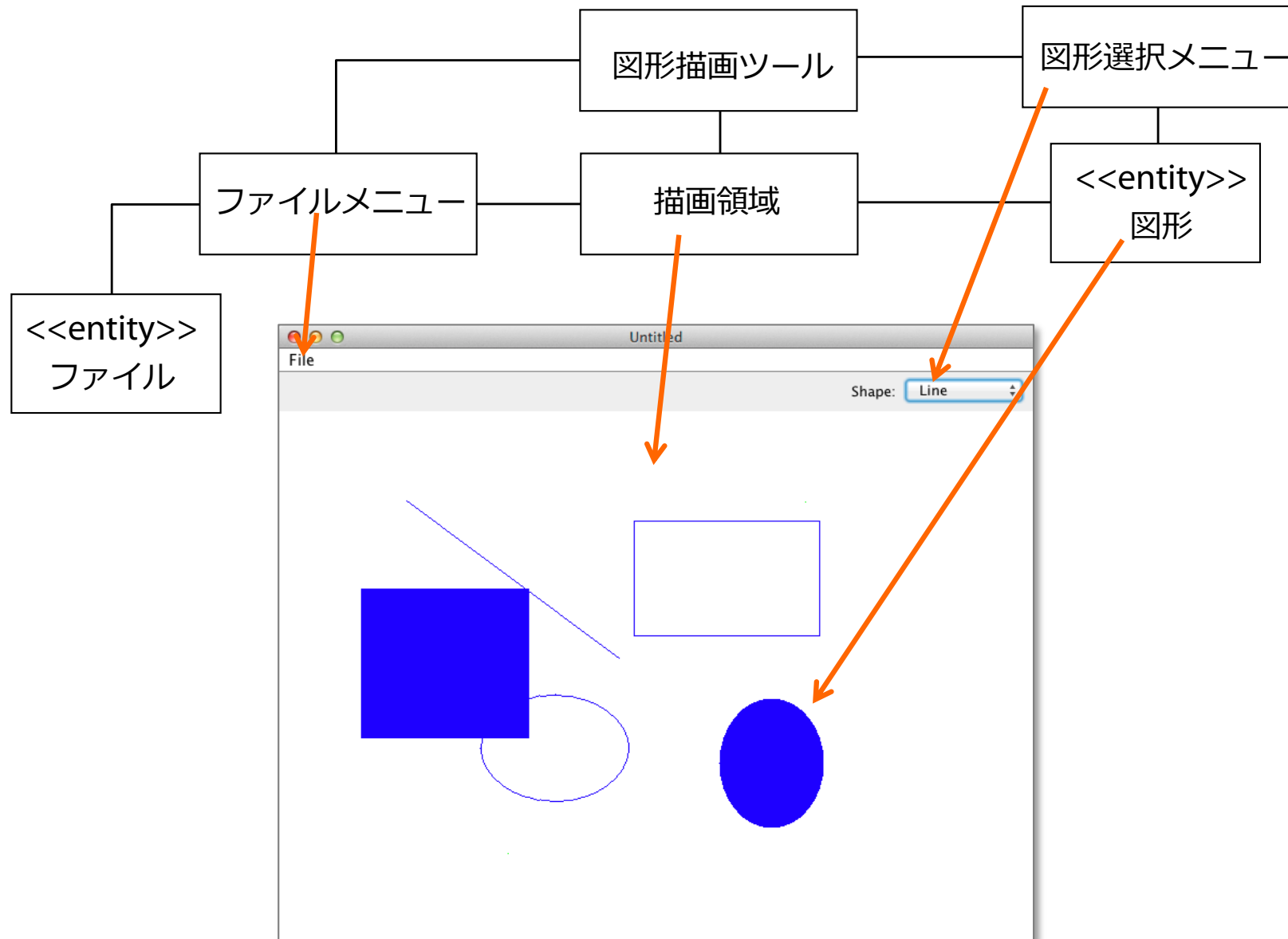
# 概念モデル(クラス図)



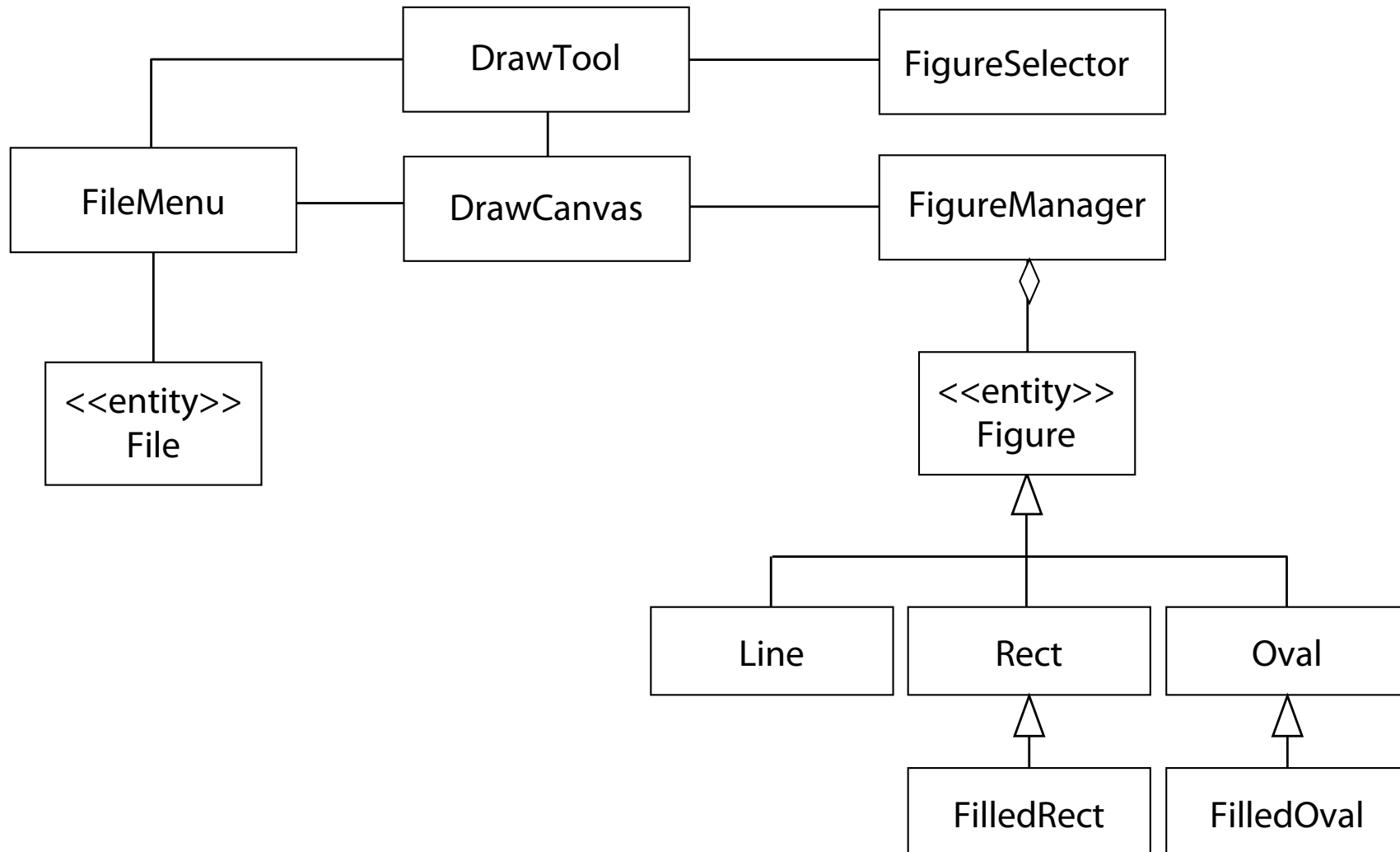
# 概念モデル(クラス図)



# 分析モデル(クラス図)

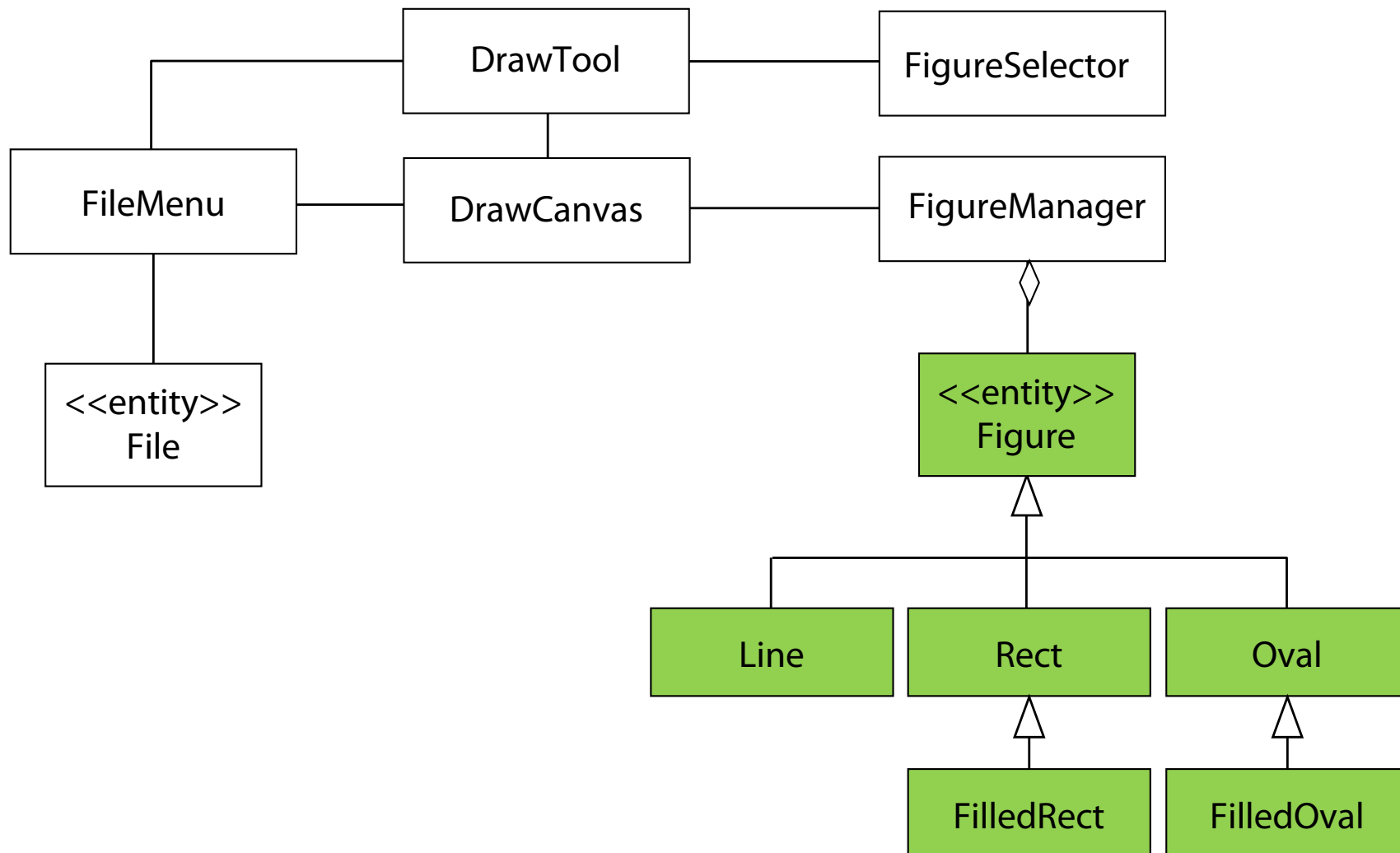


# 分析モデル(クラス図)

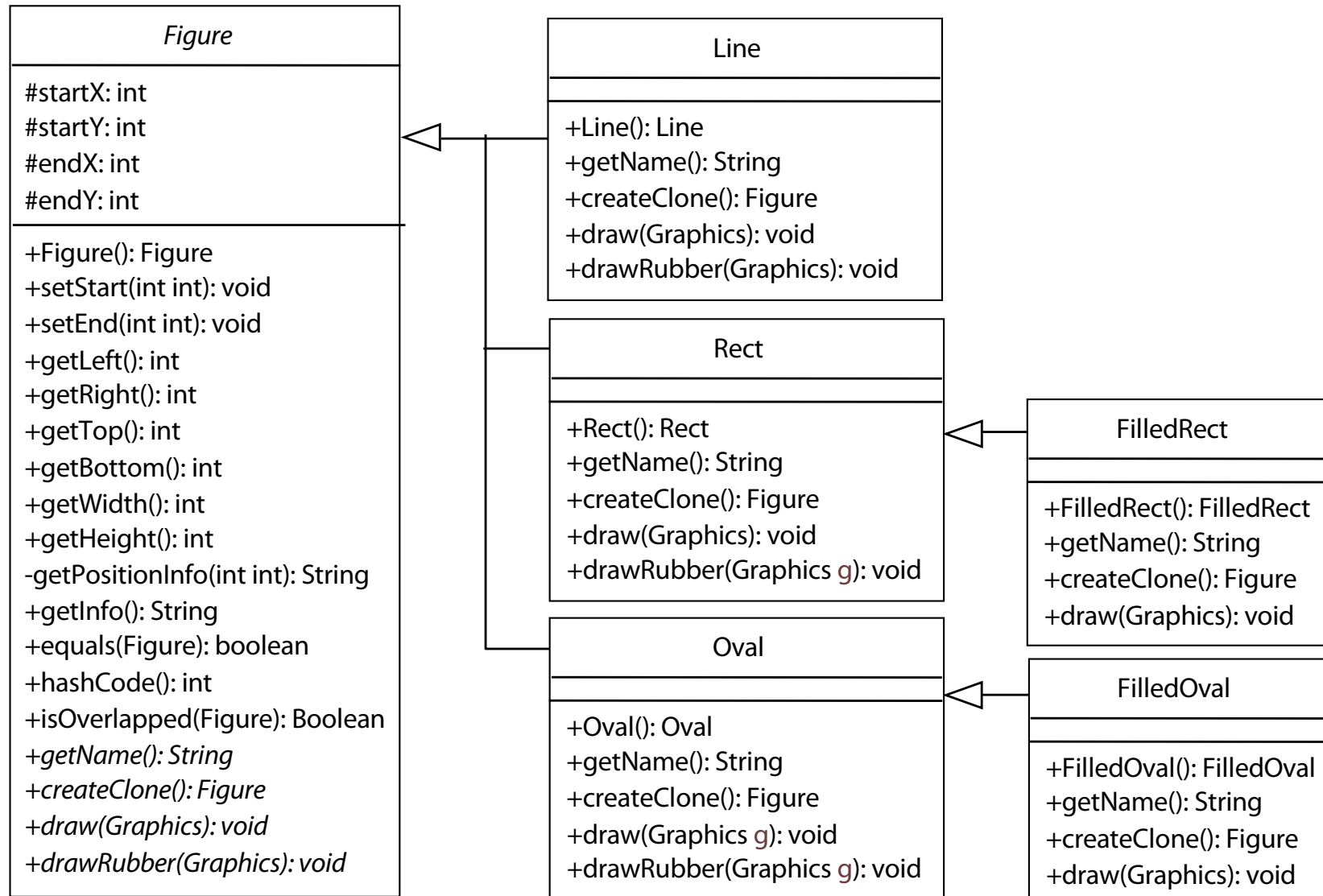




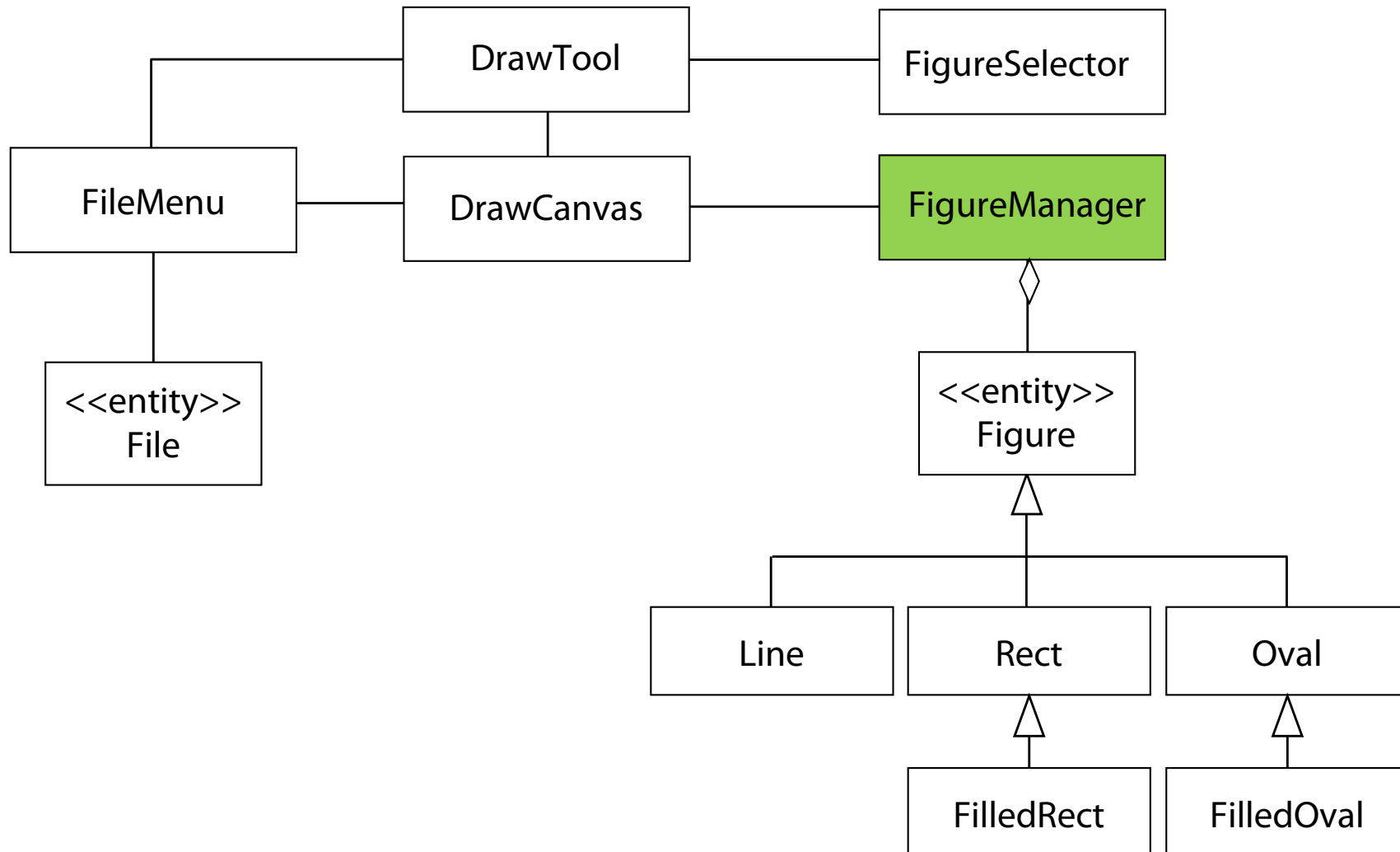
# 分析モデル(クラス図)



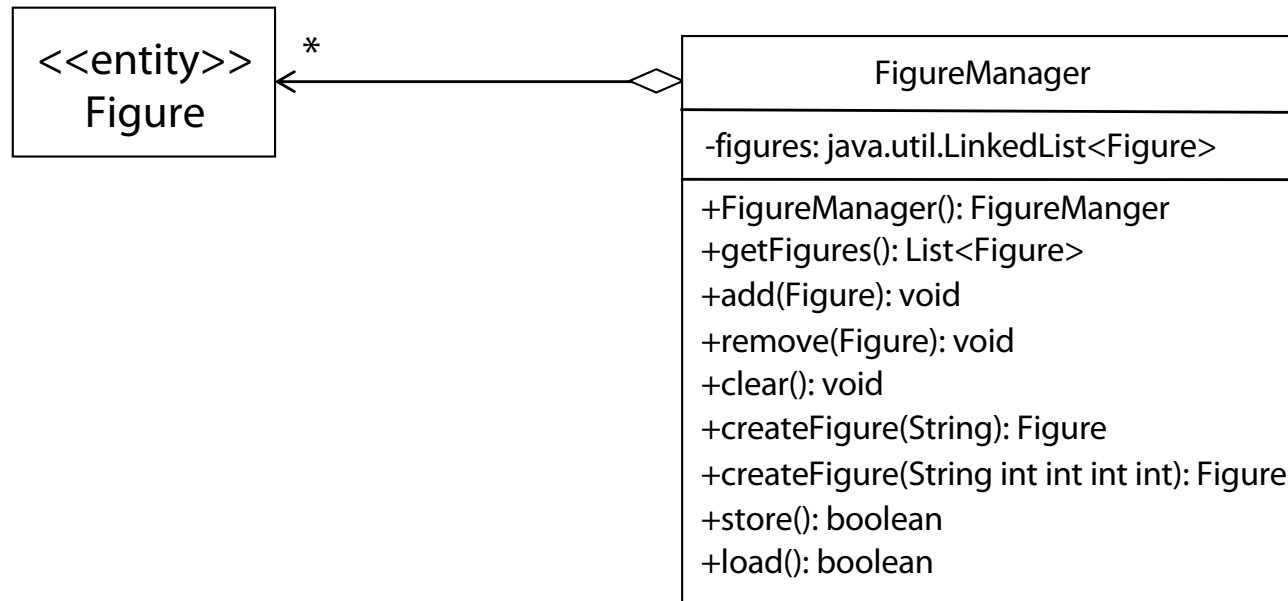
# 設計モデル(クラス図)



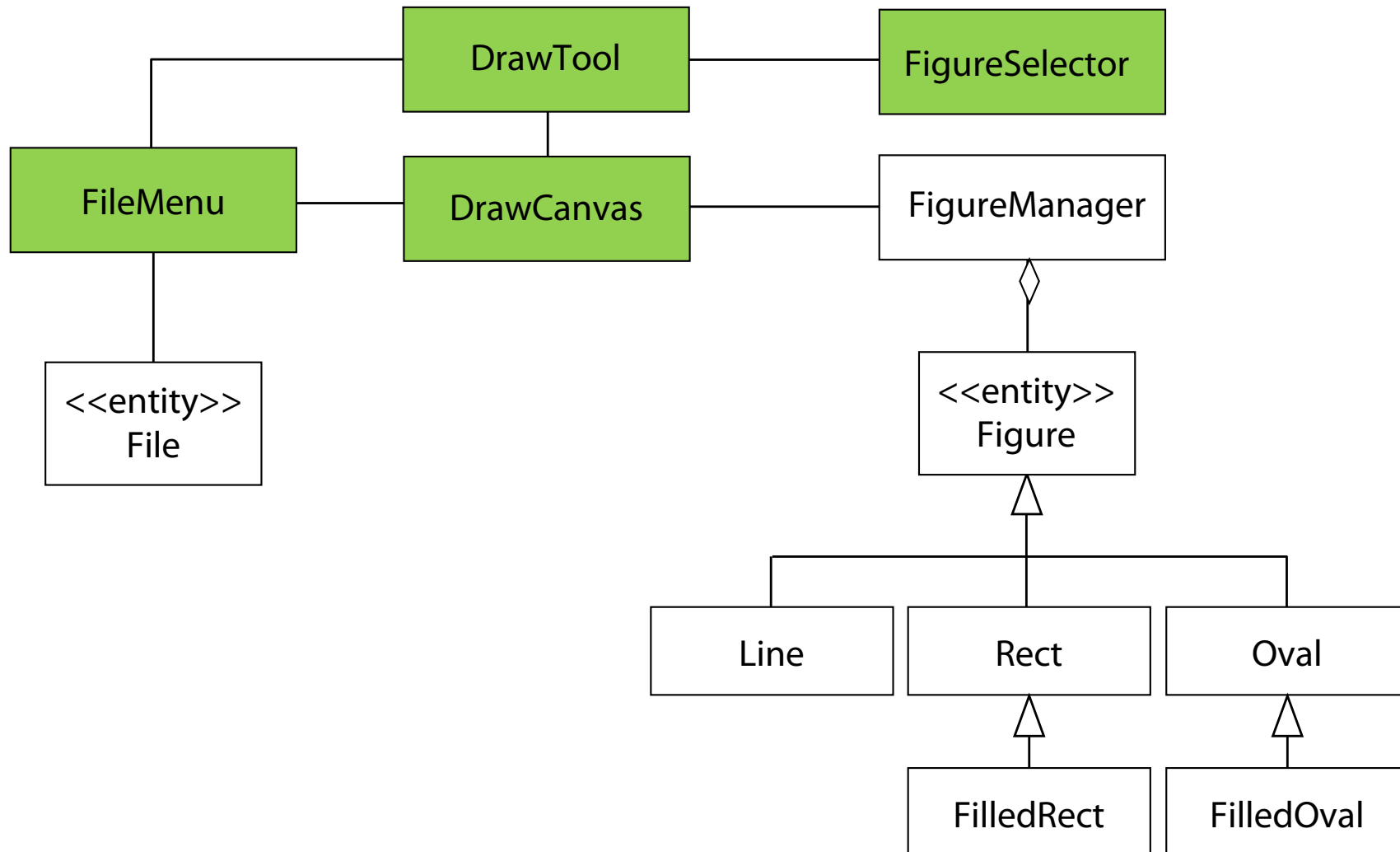
# 分析モデル(クラス図)



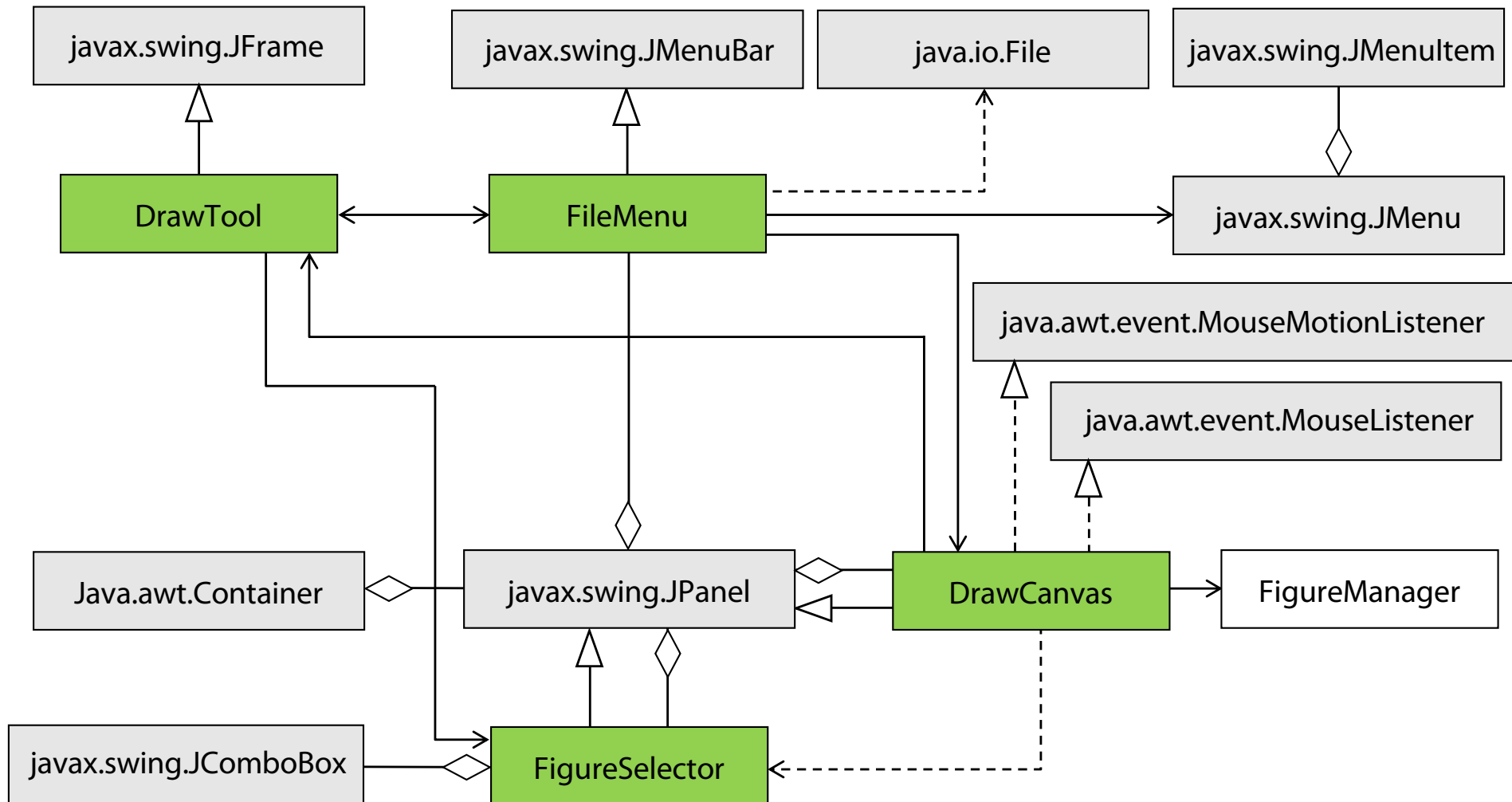
# 設計モデル(クラス図)




# 分析モデル(クラス図)



# 設計モデル(クラス図)





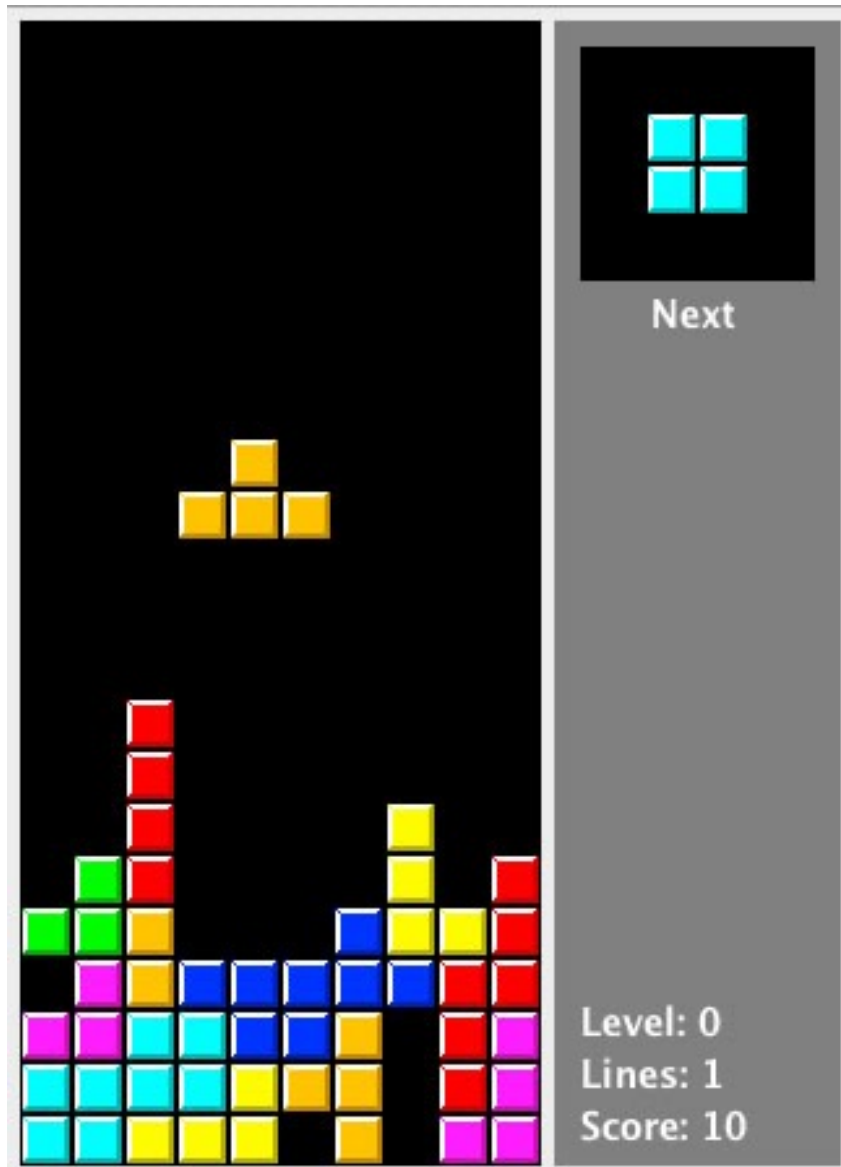
図形選択メニュー

<https://forms.office.com/r/K0V1Hqckju>

## 例題2: テトリスゲーム



# テトリスゲーム(画面イメージ)



## テトリス(Tetris)のルール説明

- 画面上部から落ちてくるブロックを左右に動かしたり, 回転しながら積んでいく
- ブロックは横一列に揃うと消え, 消した列の数により得点が増える
- ブロックが画面上部まで積みあがった時点で, ゲーム終了



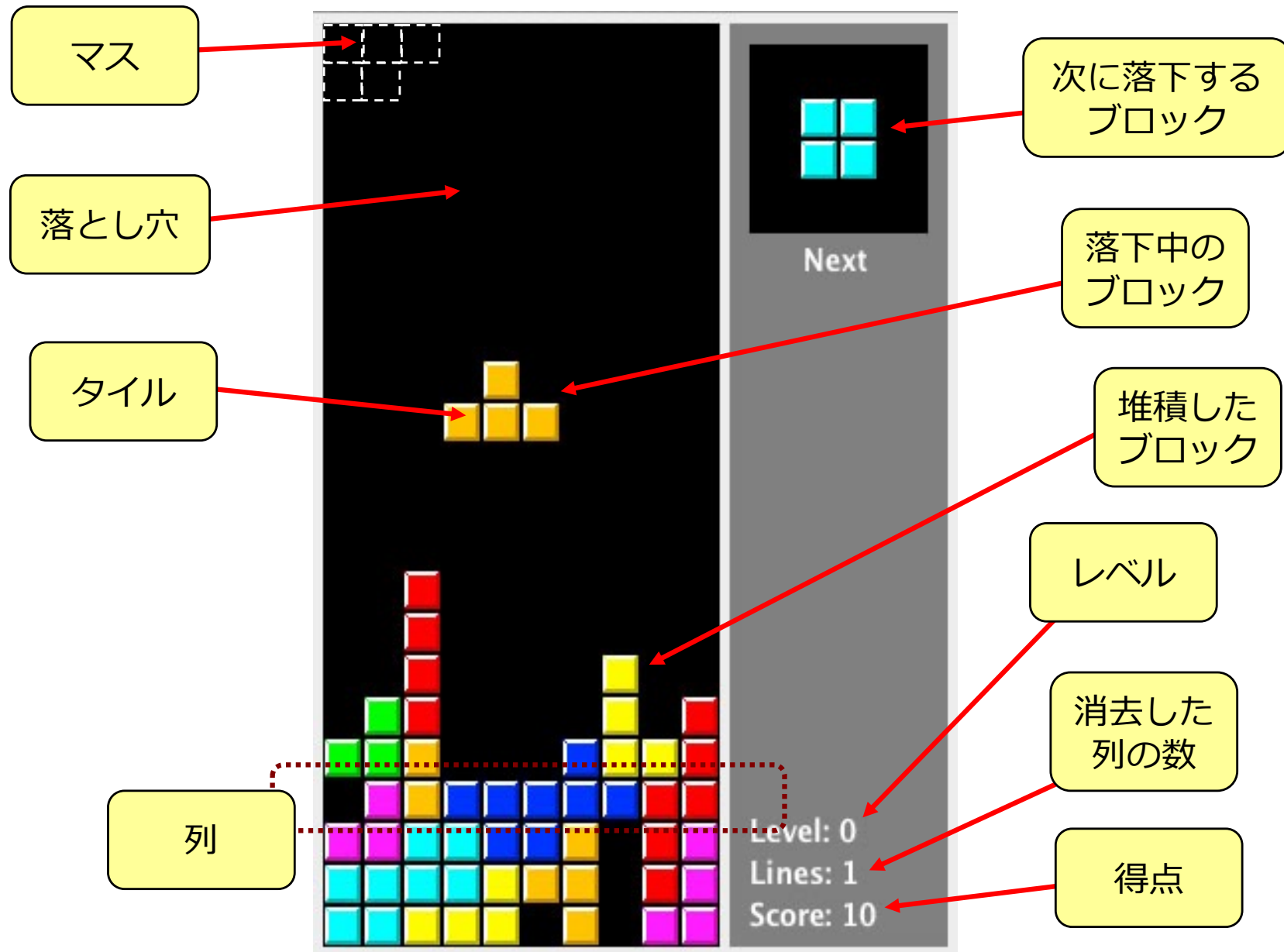
's':	ゲームスタート
'b':	左に移動
'm':	右に移動
'n':	回転

# テトリスゲーム(要求文書)

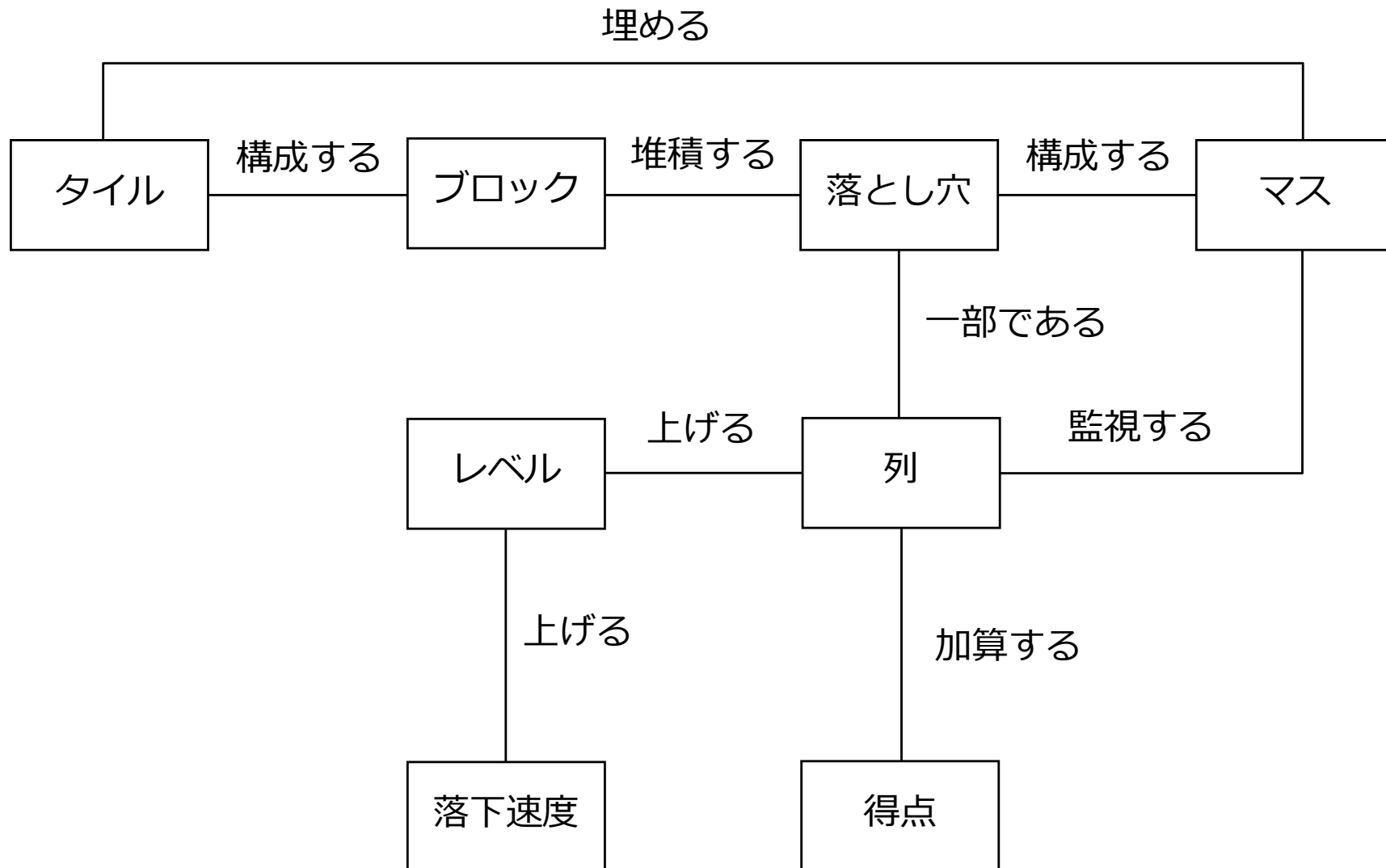
---

- ブロックが落とし穴の上部から落ちてきて、順番に積み重なる
- 落とし穴は縦に22個、横に10個の正方形のマスで構成されている
- ブロックは7種類あり、それぞれ4つの(マスと同じ大きさの)タイルで構成されている
- プレーヤーは、レバーで落下中のブロックを左右に移動させることができる。また、ボタンを押すことでブロックを回転させることもできる
- タイルは横一列に揃うと消去され、その列より上部のタイルが消えた列の数だけ下がる
- ブロックが落とし穴の最上部まで積み重なるとゲームが終了となる
- タイルを消去すると得点が加算される。複数の列のタイルを同時に消去すると、それだけ高得点となる
- 消去したタイルの数に応じてゲームのレベルが上がり、ブロックの落下速度が速くなる
- 次に落ちてくるブロックは落とし穴の横に表示される
- レベル、得点、消去した列の数が落とし穴の横に表示される

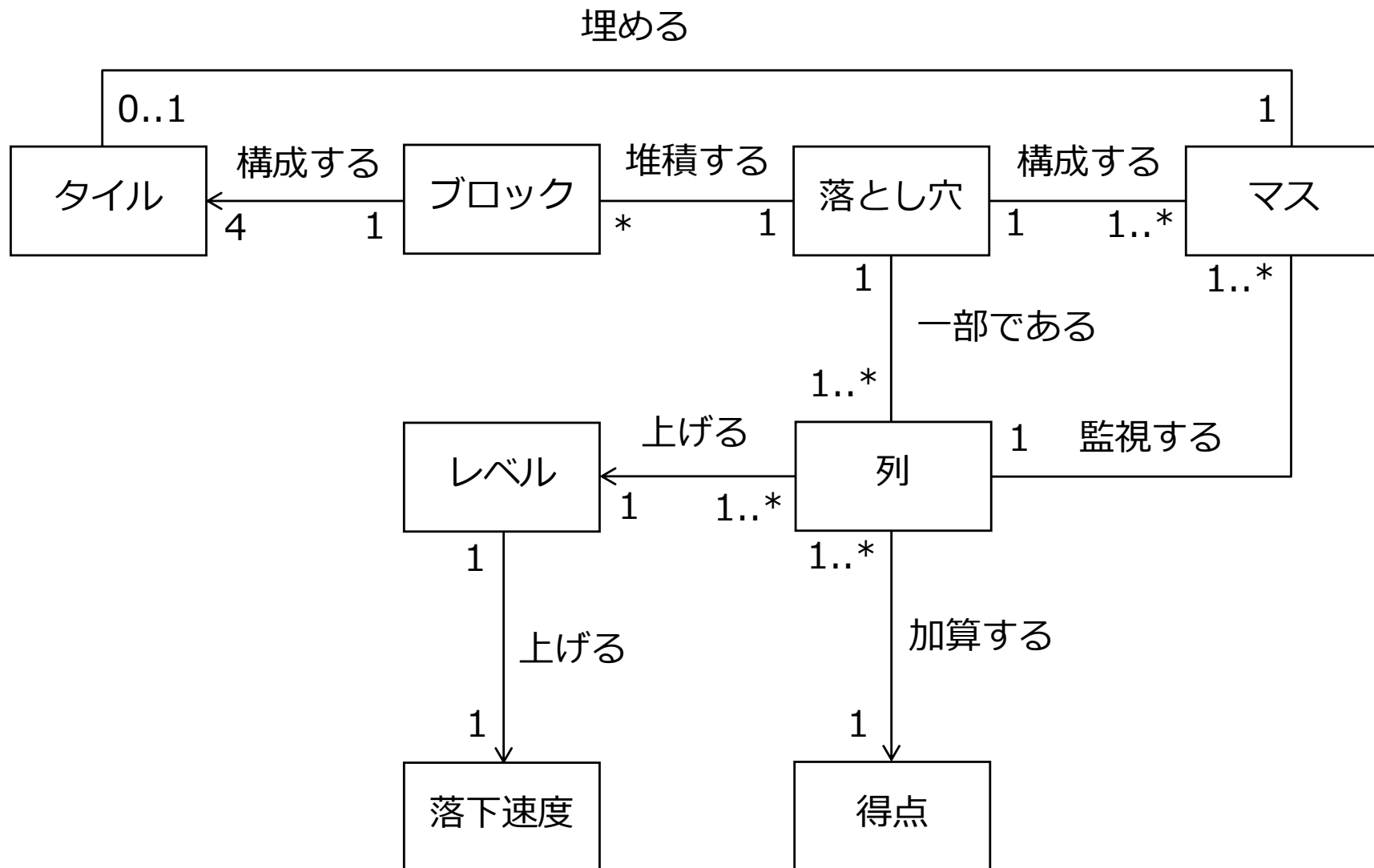
# クラスの抽出



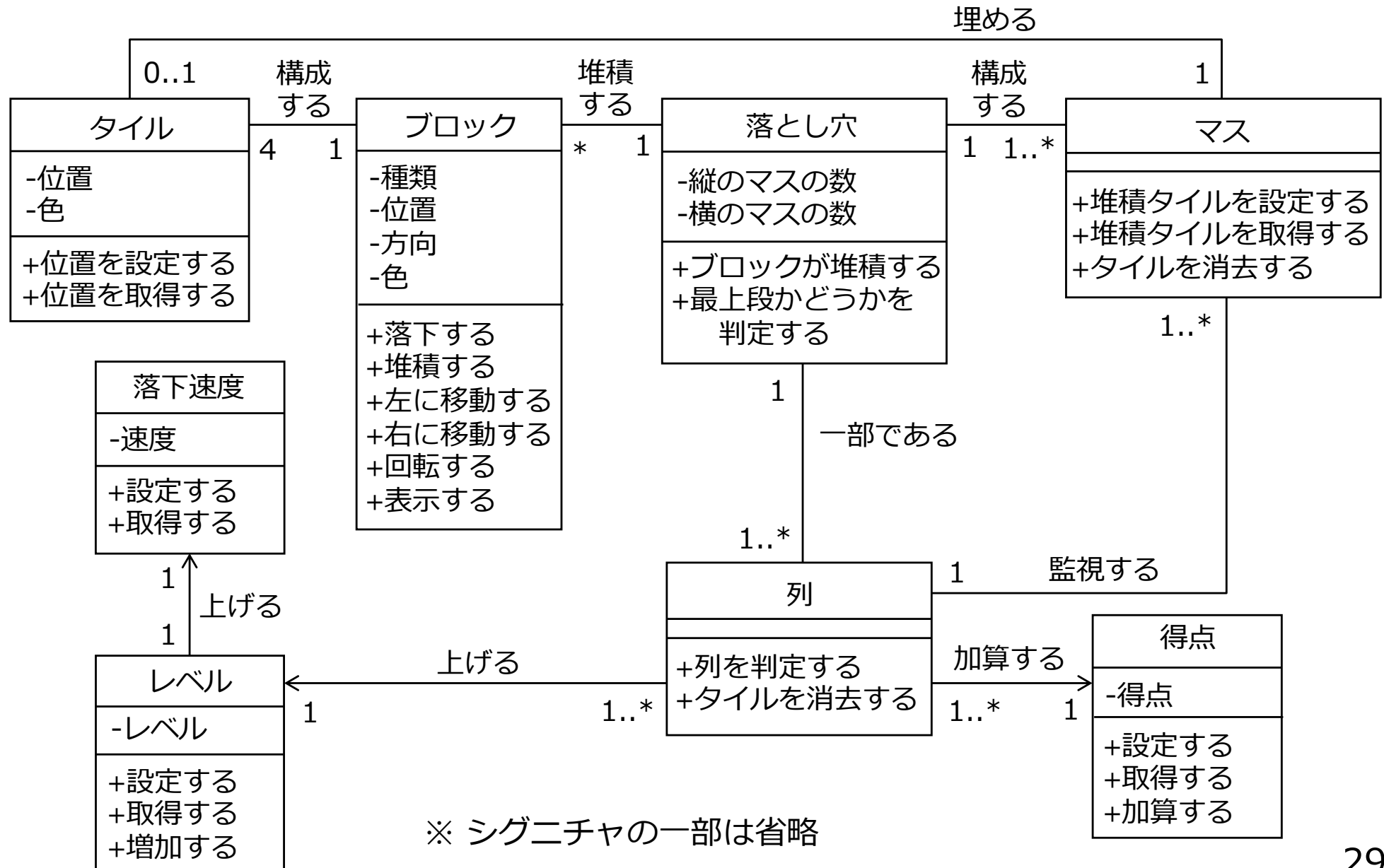
# 概念モデル(クラス図)



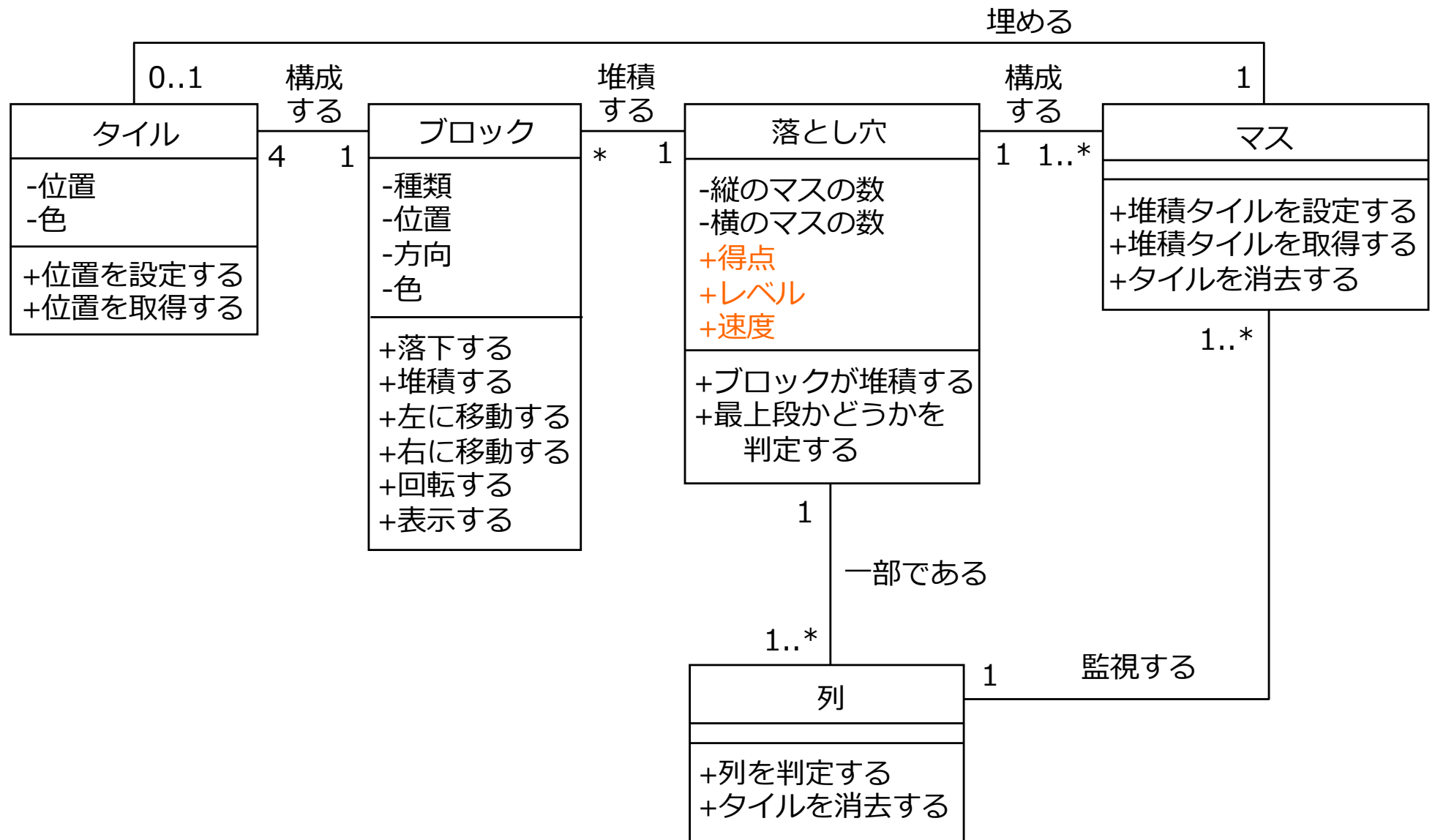
# 概念モデル(多重度付き)



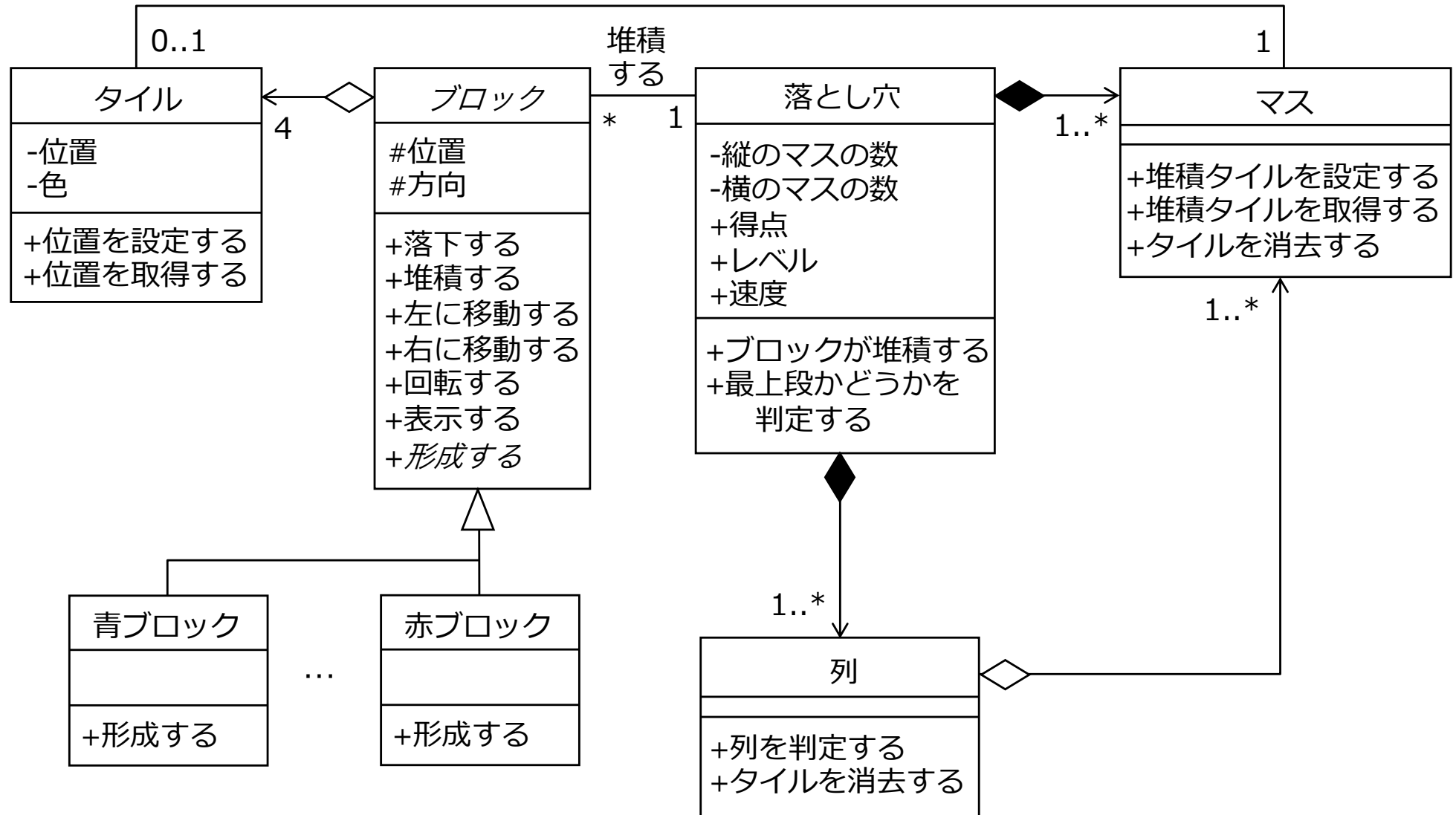
# 分析モデル(クラス図)



# 分析モデル(クラスの洗練 1/3)

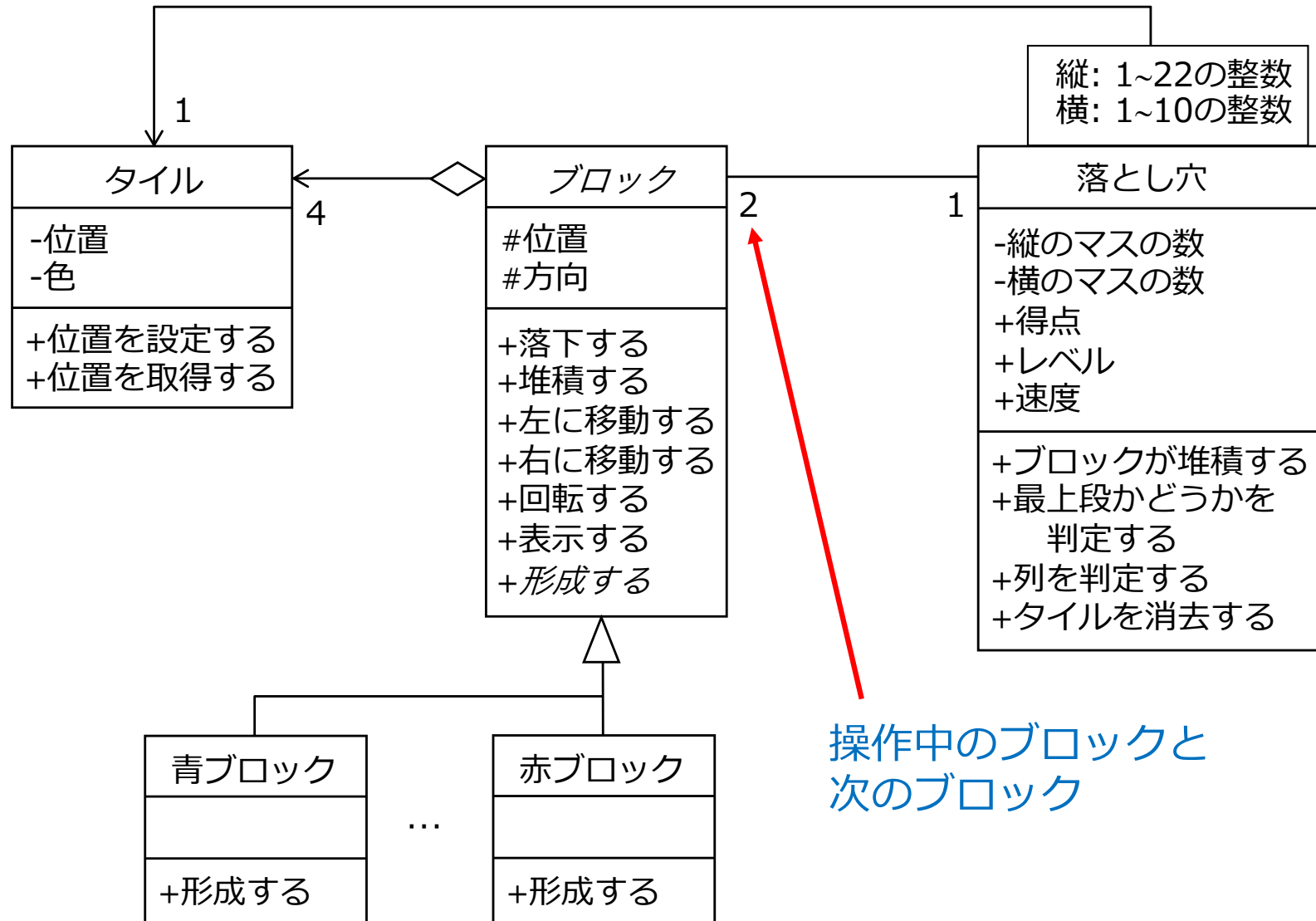


# 分析モデル(クラスの洗練 2/3)

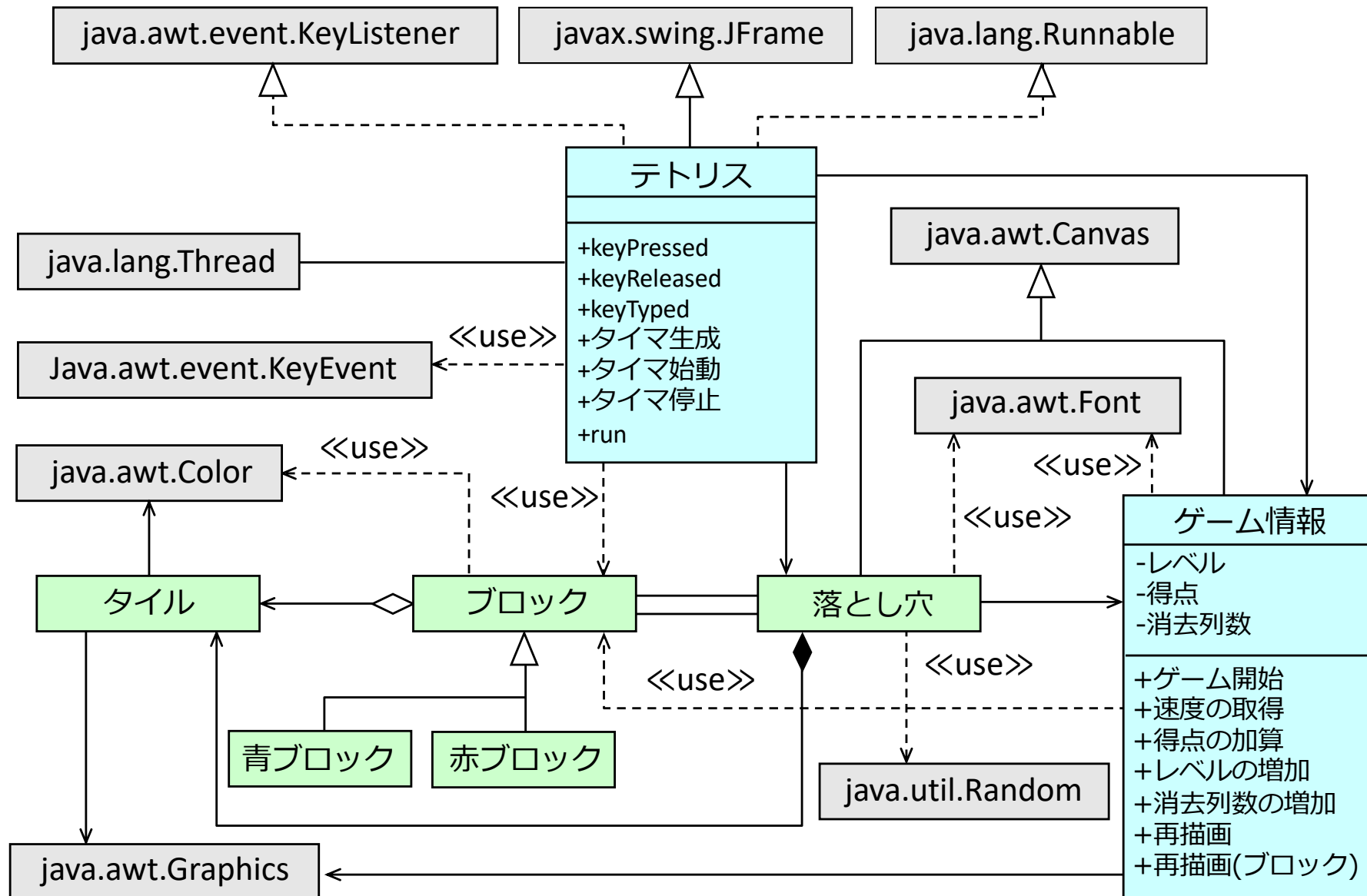




# 分析モデル(クラスの洗練 3/3)



# 設計モデル(クラス図)



# 例題のソースコード

---

<https://www.fse.cs.ritsumei.ac.jp/lesson/DUT-00/>

図形描画ツール (DrawTool.zip)

DrawToolクラスのmain()メソッドから実行

図形描画ツール2 (DrawTool2.zip)

DrawToolクラスのmain()メソッドから実行

テトリスゲーム (Tetris.zip)

Tetrisクラスのmain()メソッドから実行

マインスイーパーゲーム (MineSweeper.zip)

MineSweeperクラスのmain()メソッドから実行