

応用情報工学演習

第4回 PythonによるGUI作成

2019/10/17

谷口研究室

今日やること

1. オブジェクト指向の復習
2. PyQt5の基礎
3. 課題： MinesweeperをGUIアプリにしよう

オブジェクト指向の復習

オブジェクト指向分析とは？

- **オブジェクト指向分析**(object-oriented analysis)：実世界のモデルをソフトウェアで表現するために「オブジェクト」を構成単位としてソフトウェアを構築する枠組み

構造化分析

- 機能中心
- **処理（プロセス）**へデータが流れてくる
- データフロー図
- **手続き型**言語: COBOL, PL/I, C言語

オブジェクト指向分析

- オブジェクト中心
- **オブジェクトへメッセージ**がくる
- クラス図, オブジェクト図など
- **オブジェクト指向**言語 : Java, C++

オブジェクトとは？

- オブジェクト(object)

- (定義 1) 人間が認知できる具象的あるいは抽象的な「モノ」 (に対するコンピュータ内の表現)
- (定義 2) 属性 (データ) とそのデータに対する操作 (メソッド) を一つにまとめたもの

実世界



車



コンピュータ

車オブジェクト

機能 (操作) :

- ・ 前進する
- ・ 停止する



社員

(会議室管理担当)



会議室管理オブジェクト

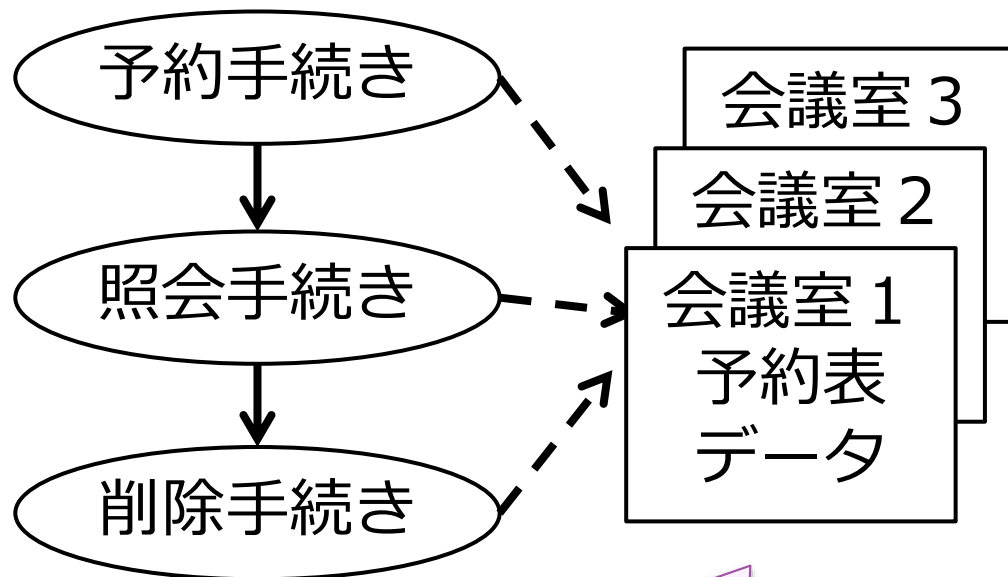
機能 (操作) :

- ・ 予約
- ・ 照会
- ・ 削除

オブジェクトとは？（定義 2）

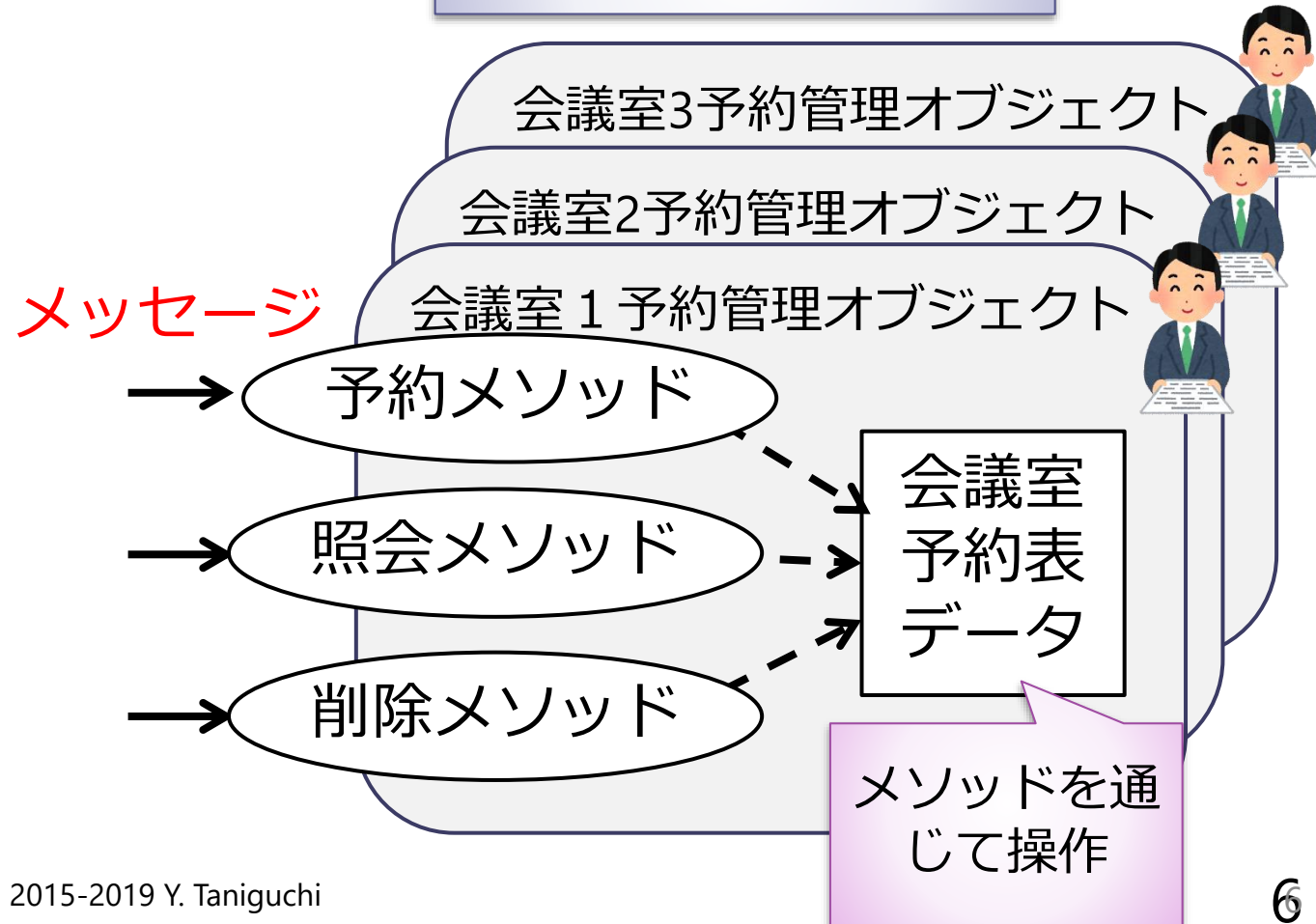
- オブジェクト：属性（**データ**）とそのデータに対する操作（**メソッド**）を一つにまとめたもの

手続き型



外部から自由に
アクセス可能

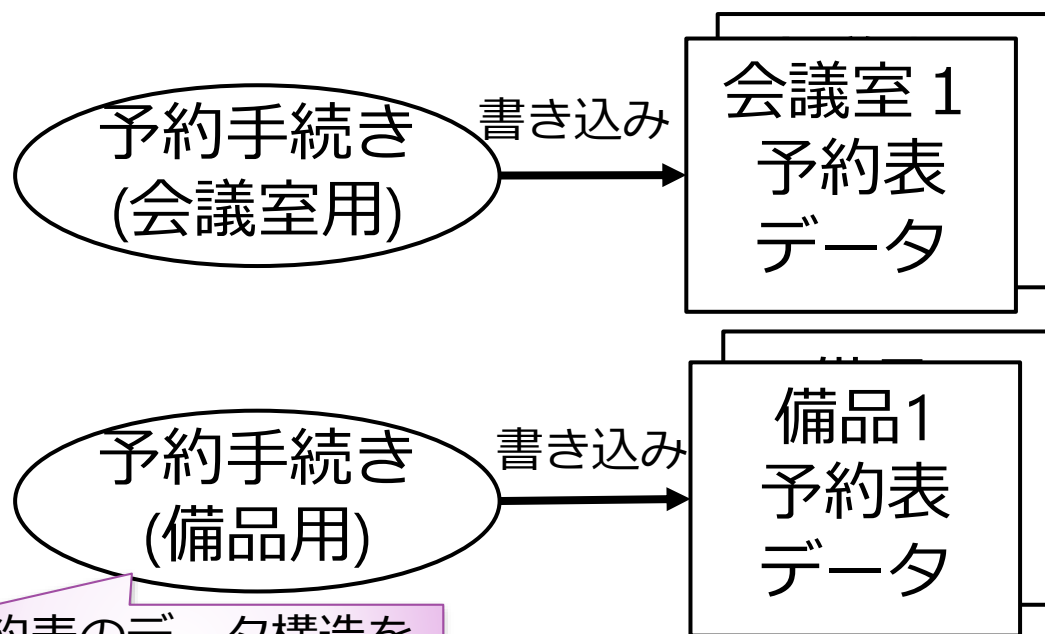
オブジェクト指向



オブジェクト指向のメリット

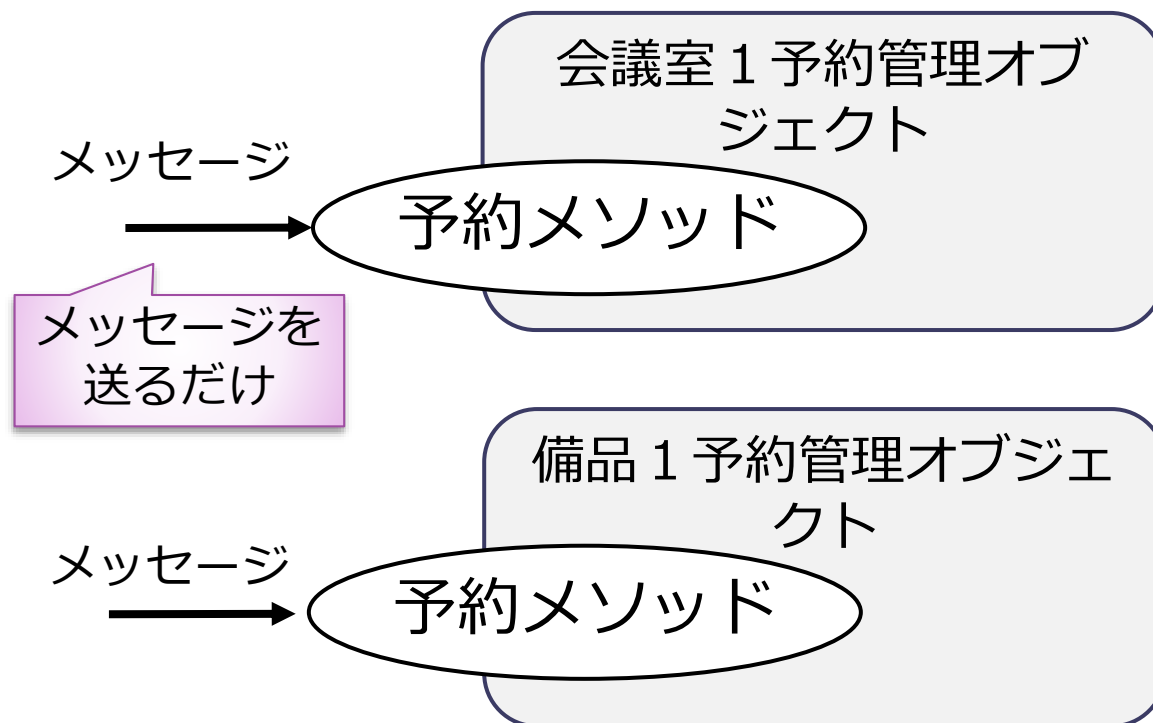
- **手続き型**：プログラマーがデータ型を意識して、適切な手続き（関数）を選ぶ必要がある。
- **オブジェクト指向**：オブジェクトが自分自身のデータの操作方法を知っているの
で、メッセージを送るだけで良い。

手続き型



予約表のデータ構造を意識しておく必要あり

オブジェクト指向



オブジェクト指向の基本概念

メッセージ
パッシング

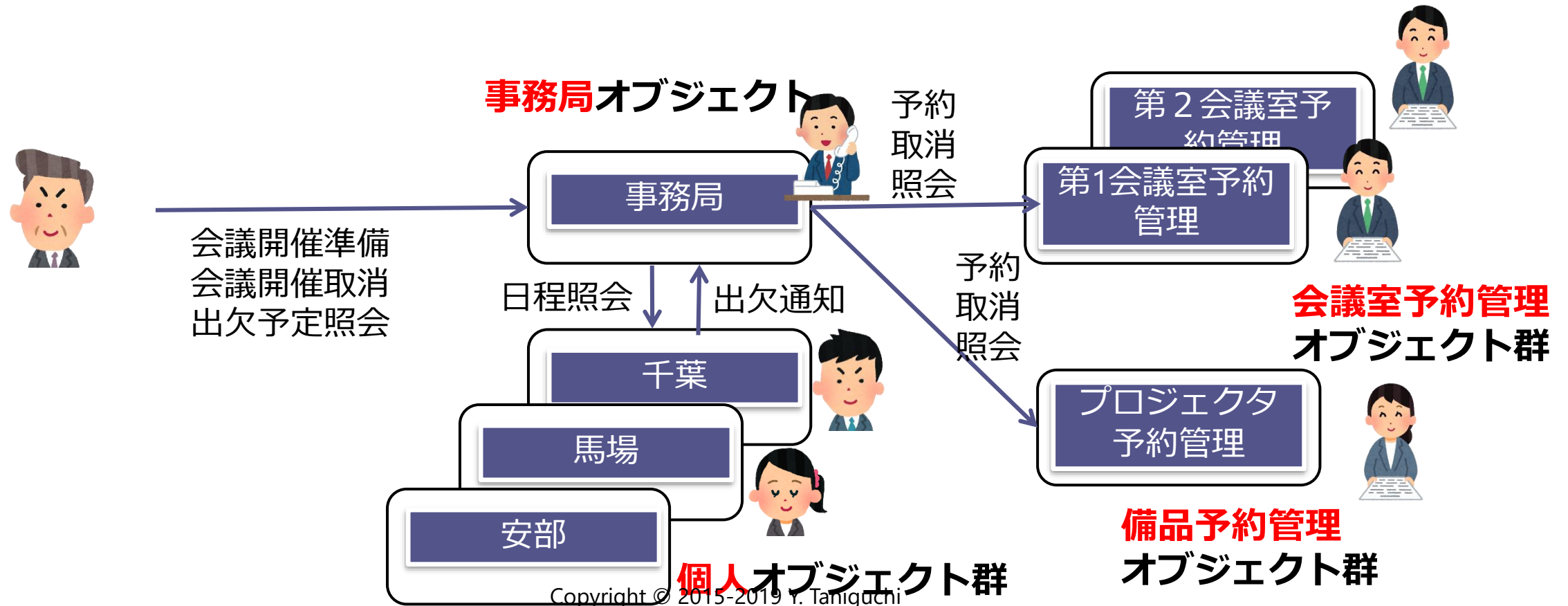
カプセル化

クラスとイン
スタンス

継承

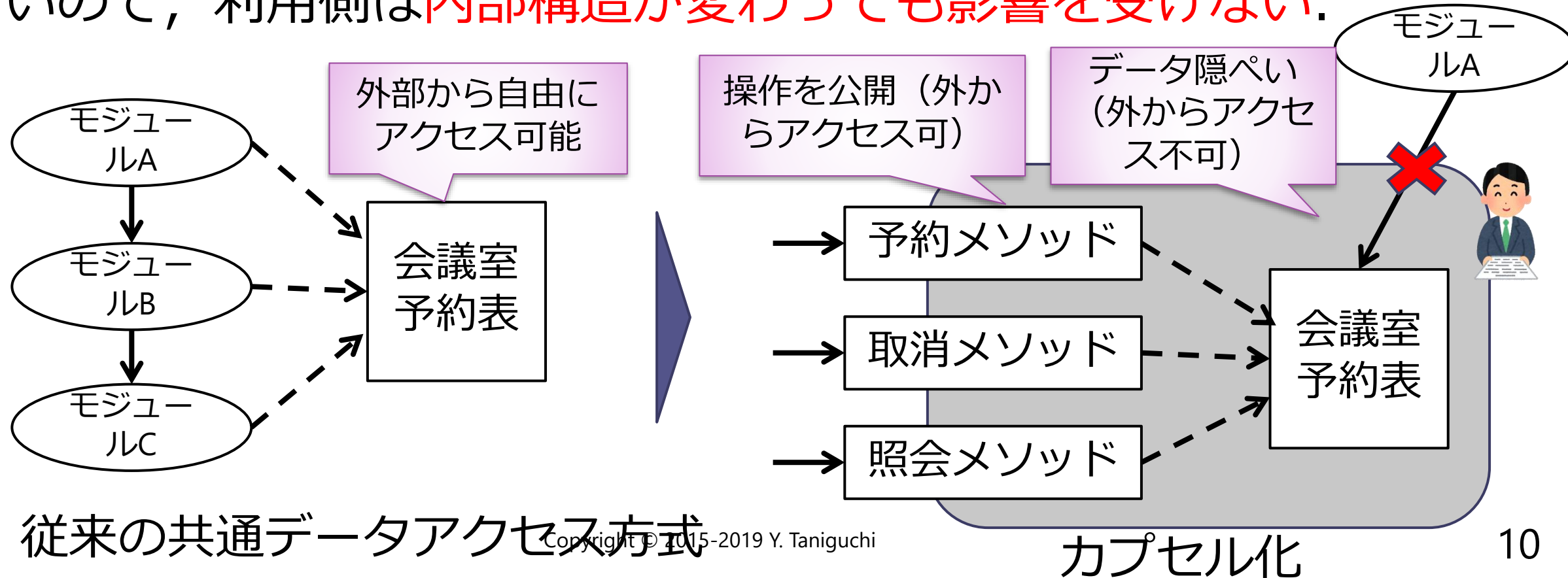
メッセージパッシング

- **メッセージパッシング**：機能呼び出すのではなく、オブジェクトにメッセージを送ることで処理を依頼する考え方
⇒実世界の問題解決に近い業務モデルを表現可能（プログラマも楽）



カプセル化

- カプセル化：データ（属性）とそれに特化した操作（メソッド）をまとめてモジュール化すること
 - ⇒データや操作の隠ぺい／公開を制御できる.
 - ⇒外部に公開された操作（外部インタフェース）しか許可されないのので、利用側は内部構造が変わっても影響を受けない。



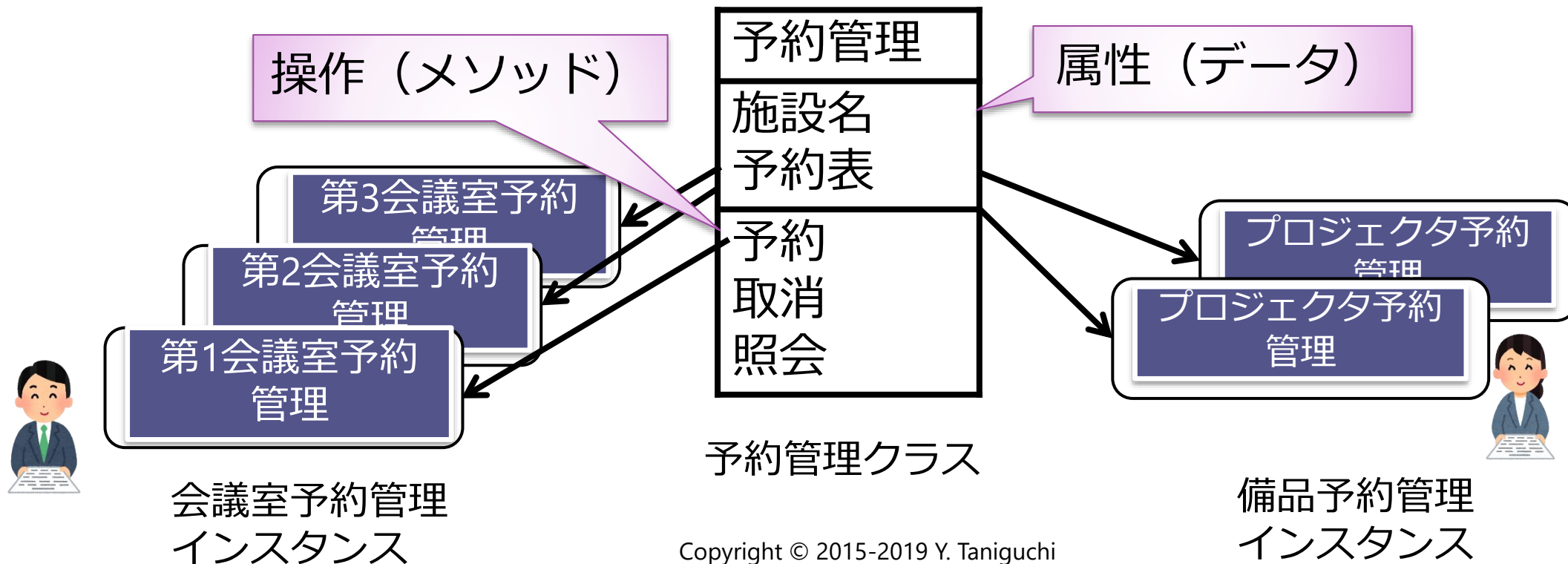
従来の共通データアクセス方式

Copyright © 2015-2019 Y. Taniguchi

カプセル化

クラスとインスタンス

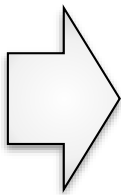
- **クラス**：共通する**属性**（データ変数）や**操作**（メソッド）を持つオブジェクトを**抽象化**したもの（**ひな形**）
 - **インスタンス**：クラスを指定して生成される具体的なオブジェクト（属性の値だけが異なる）
- ⇒データの構造と操作が同じ**インスタンスを簡単に作る**仕組み



クラスとインスタンス

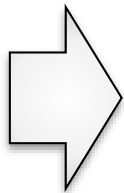


クラス（ひな形）



インスタンス（変更部分だけ指定）

予約管理
施設名 予約表={}
予約 取消 照会



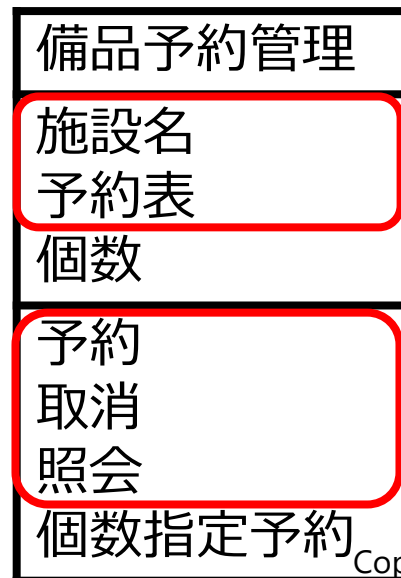
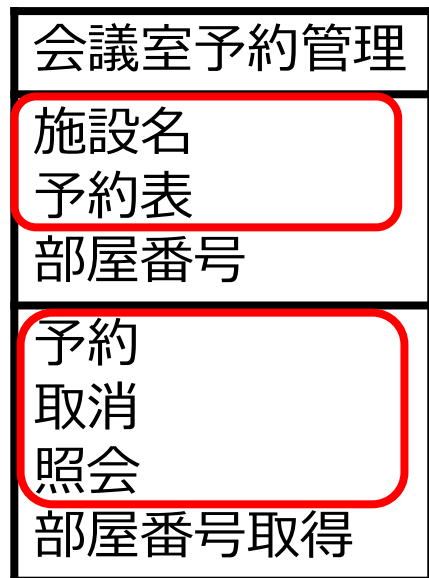
会議室1予約管理
施設名="会議室1" 予約表={}
予約 取消 照会

プロジェクタ1予約管理
施設名="projector1" 予約表={}
予約 取消 照会

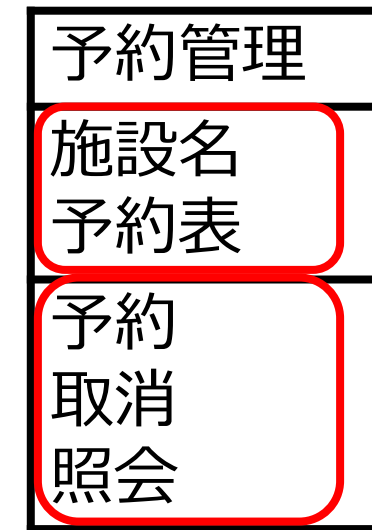
継承（インヘリタンス）

- **継承**(inheritance): クラスを**階層化**し, **親クラス**（上位クラス）のもつ属性・操作を**子クラス**（下位クラス）の属性・操作として共有することで冗長性を削減する仕組み
⇒ **差分のみ定義**すれば良いので,
ソフトウェアの**再利用が容易**になる

- (1) 類似クラスを見つける
- (2) 共通する属性・操作をくくりだして親クラスを作る

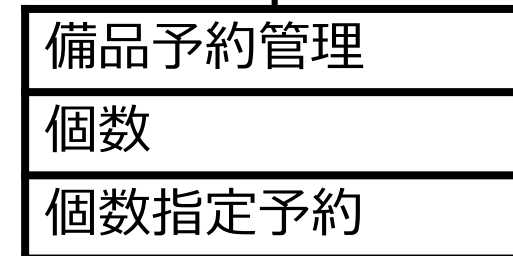
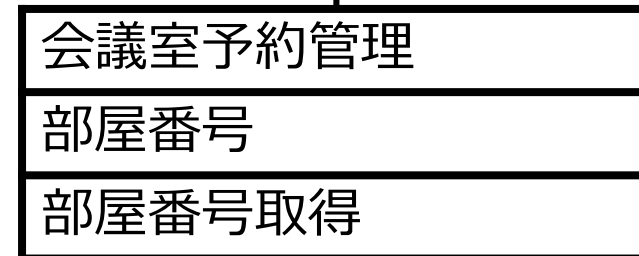


階層化



親クラス
（上位クラス）
（基底クラス）

子クラス
（下位クラス）
（派生クラス）

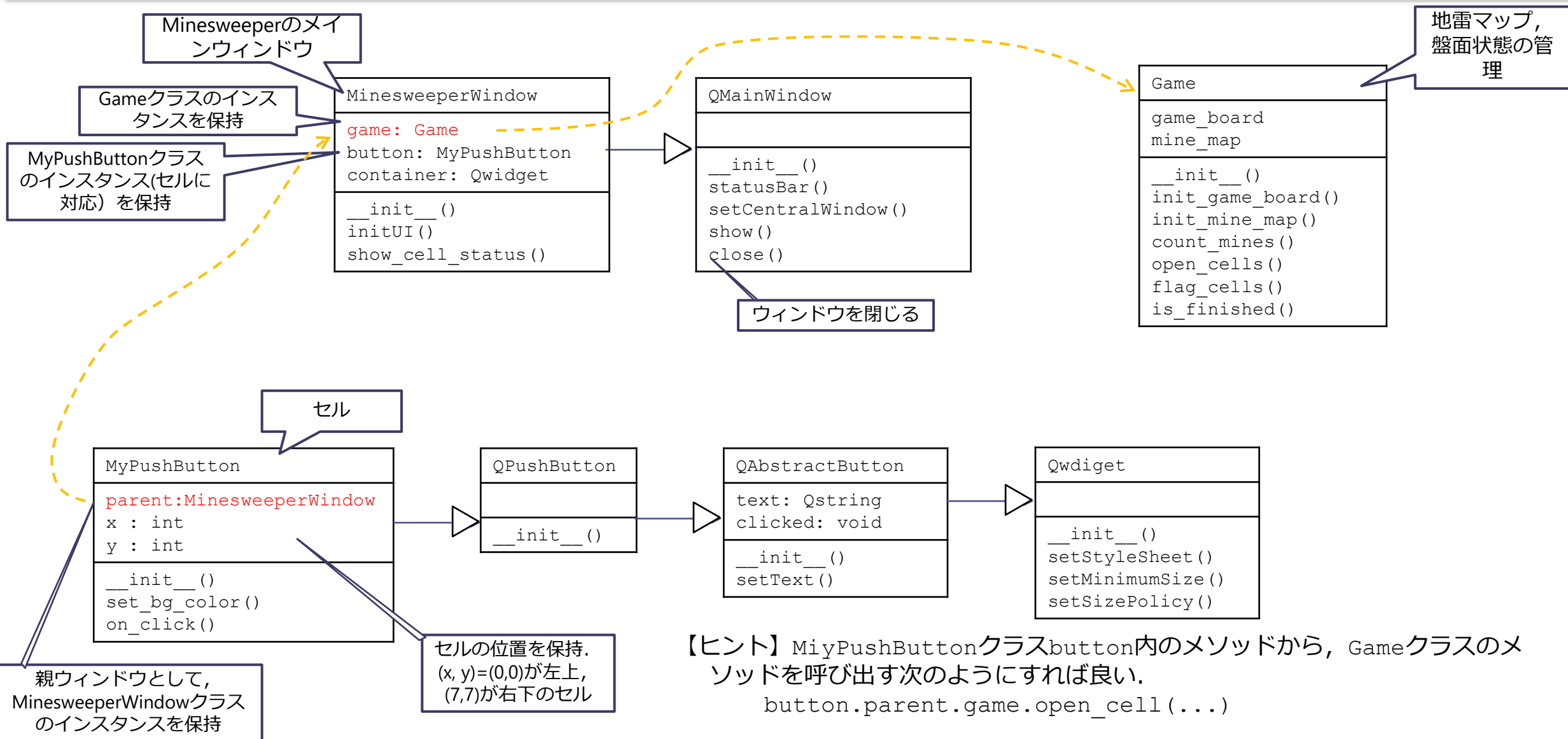


PYQT5の基礎

利用する部品（クラス）

- QApplication -- アプリケーションの生成
- QMainWindow – メインウィンドウの作成
- QStatusBar -- ステータスバーに文字列を表示
- QMenuBar -- メニューバーを作成
- QPushButton -- ボタンを作成
- QHBoxLayout, QVBoxLayout -- ボタンを自由に配置
- QMessageBox -- メッセージボックスを表示
- QGraphicsScene, QGraphicsView -- 図形を描画する
- QSlider, QLabel -- スライダーの利用

Minesweeper_gui クラス図



画面とインスタンスオブジェクトの対応

w: MinesweeperWindow

container: QWidget

vbox: QVBoxLayout

hbox: QHBoxLayout

① button[0][0]: MyPushButton

② button[0][1]: MyPushButton

⋮

