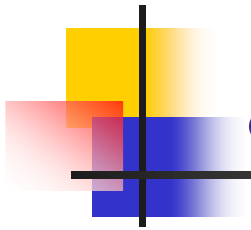
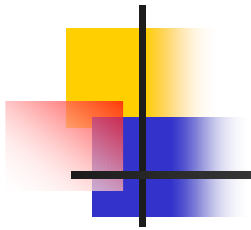


よい設計を行うには？

- 新規によい設計を行うのは難しい
- 既存の設計の修正
- どこを直せばよいか
  - ソフトウェアメトリックス
    - ソフトウェアの数値化．複雑度など
  - リファクタリング：Refactoring
    - 振る舞いを変えることなく，ソフトウェアの内部構造を変える．
    - 理解や変更が容易になるように，



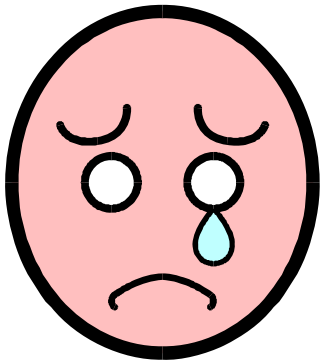
リファクタリング（１）

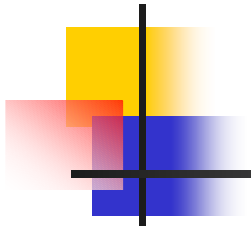
- Martin Fowler：Refactoring
- 「不吉なにおい」を感じる個所に，リファクタリングを施す．
- 「不吉なにおい」リスト
- リファクタリングリスト

リファクタリング（2）

不吉なにおいリスト：全部で22

- コードが重複している
- クラスが巨大
- メソッドが長い
- 引数が多い
- スイッチ文（条件分岐）
- ．．．





リファクタリング（３）

リファクタリングリスト

- 新しいクラス，メソッド，インタフェースの抽出
- メソッドの移動
- オブジェクトそのものの受け渡し
- ポリモルフィズムによる条件記述の置き換え

マイレージプログラムの例（復習）

航空会社のマイレージプログラム

飛行マイルをためる（会員の口座へ）
貯めたマイルに応じてサービス有
無料航空券の獲得など

	ノーマル (Normal)	シルバー (Silver)	ゴールド (Gold)	プラチナ (Platinum)
+ボーナスマイル	100	125	150	200
割引率	0%	3%	5%	10%
会員資格	<25000	<50000	<75000	≧75000

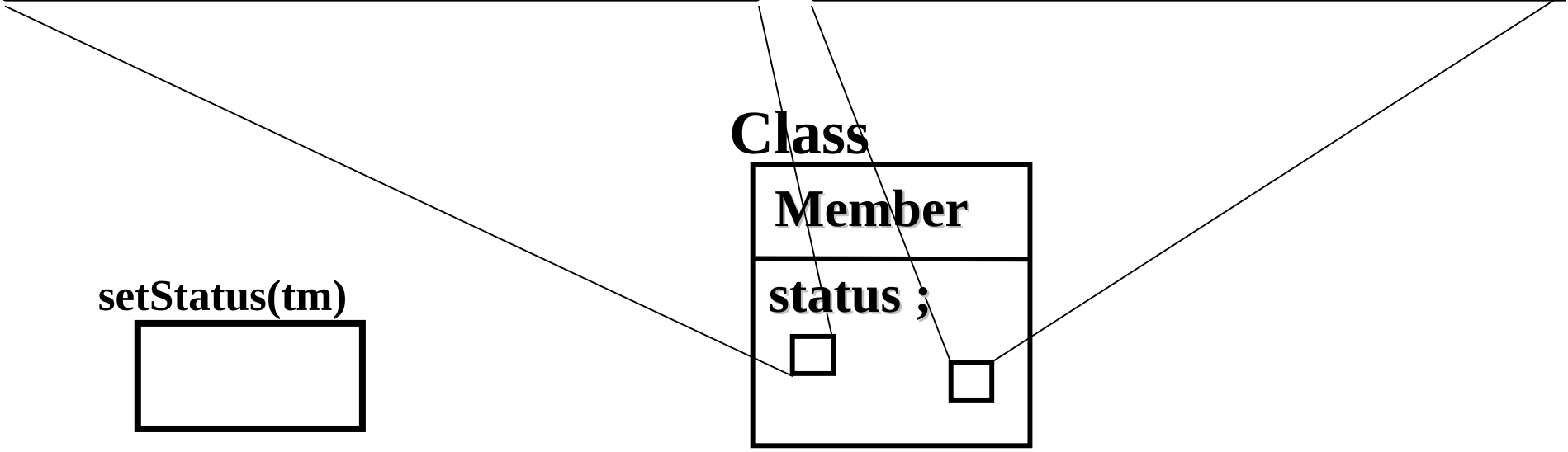
コーディング例

calculateMile(m)

```
if (status == "NORMAL"){
   return m ;
}else if (status == "SILVER"){
   return m*1.25 ;
}else return m*1.5 ;
```

getPrice(p)

```
if (status == "NORMAL"){
   return p ;
}else if (status == "SILVER"){
   return p*0.97 ;
}else return p*0.95 ;
```



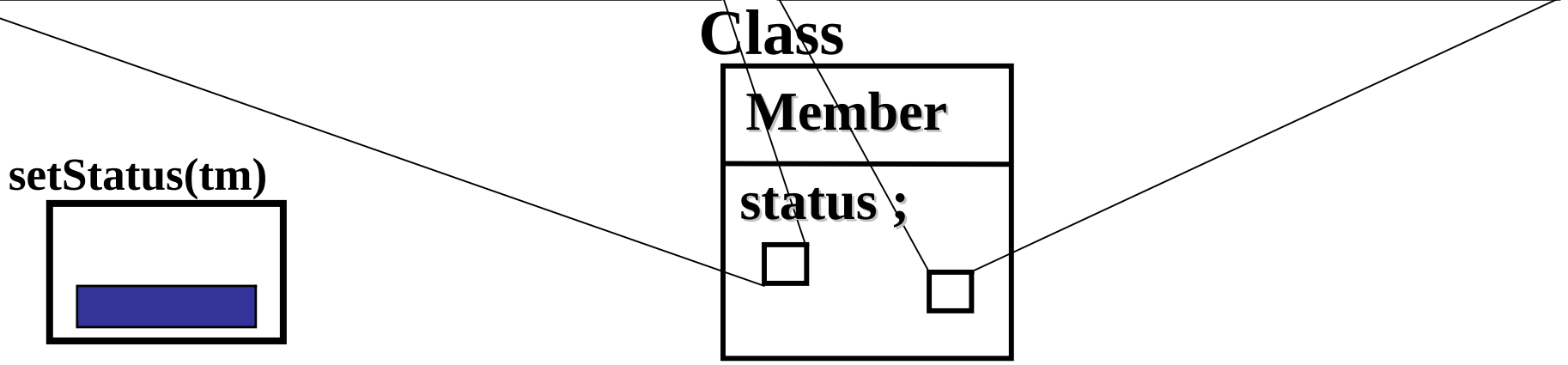
変更例

calculateMile(m)

```
if (status == "NORMAL"){
   return m ;
}else if (status == "SILVER"){
   return m*1.25 ;
}else if (status == "GOLD"){
   return m*1.5 ;
}else return m*2.0 ;
```

getPrice(p)

```
if (status == "NORMAL"){
   return p ;
}else if (status == "SILVER"){
   return p*0.97 ;
}else if (status == "GOLD"){
   return p*0.95 ;
}else return p*0.9 ;
```



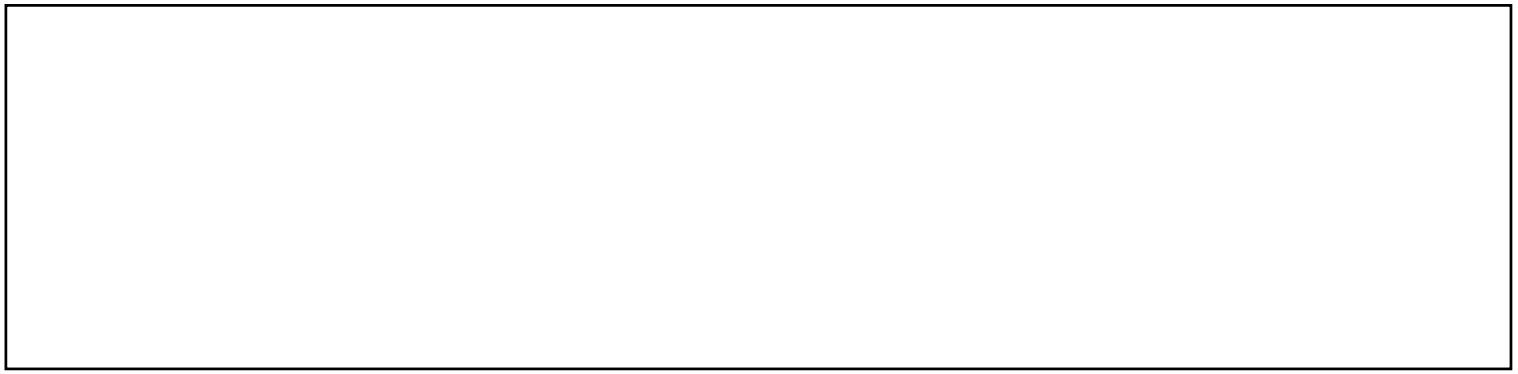
問題点

- 同じif文の構造（条件分岐）が複数のメソッドに出現
  - メンバーのstatusによって処理が異なるという構造
- ifに関連する変更や拡張が，複数のメソッドに及んでしまう.
  - メンバーのstatusの種類を増やした場合など

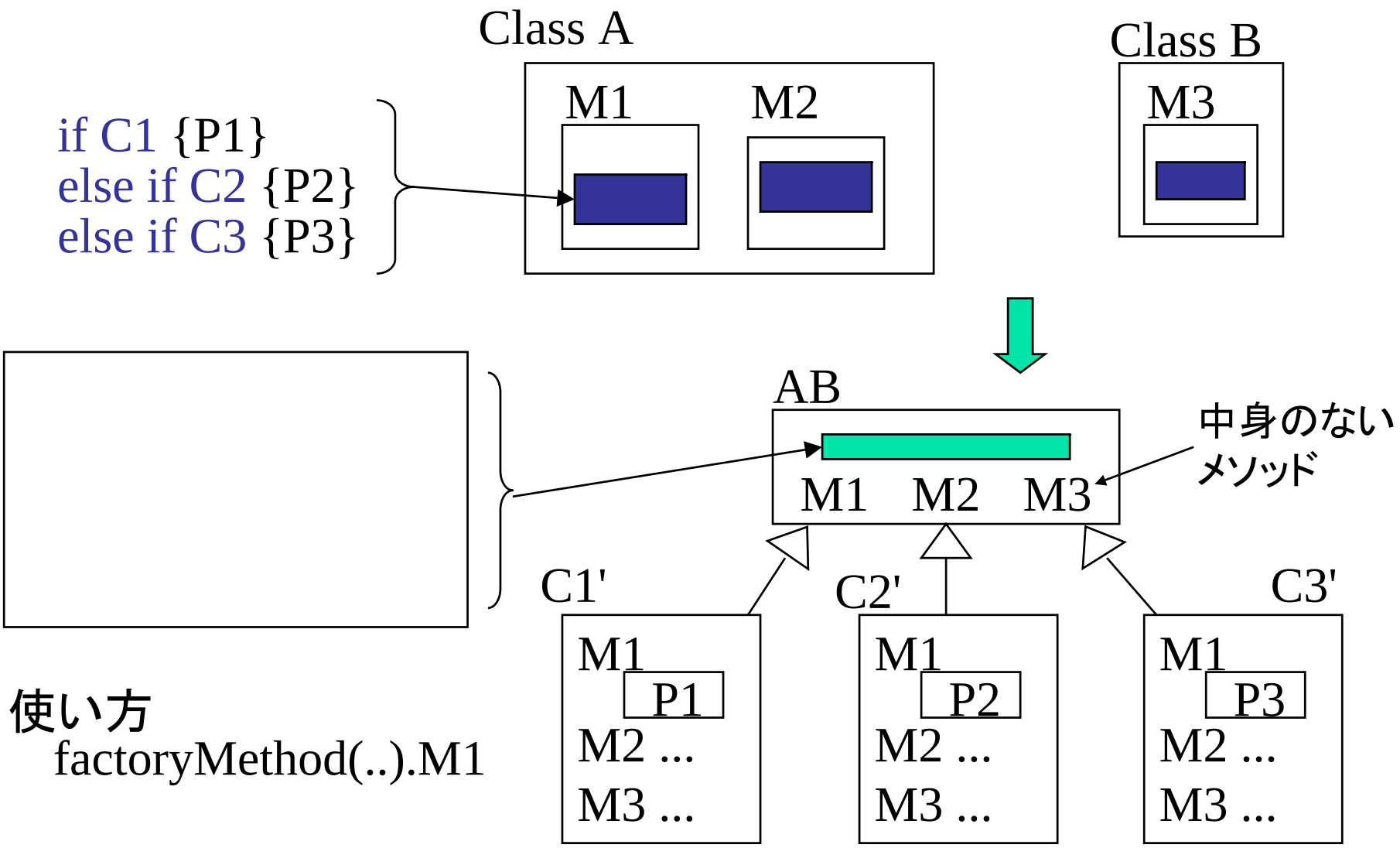
if文を１箇所にまとめられないか

リファクタリングの例

- コードの重複
- スイッチ文（条件分岐文）



１箇所にまとめる



使い方

factoryMethod(..).M1

ポリモルフィズムによる条件分岐

```
factoryMethod("PLATINUM")
                .calculateMile(6000)
factoryMethod("PLATINUM")
                .getPrice(1500)
```