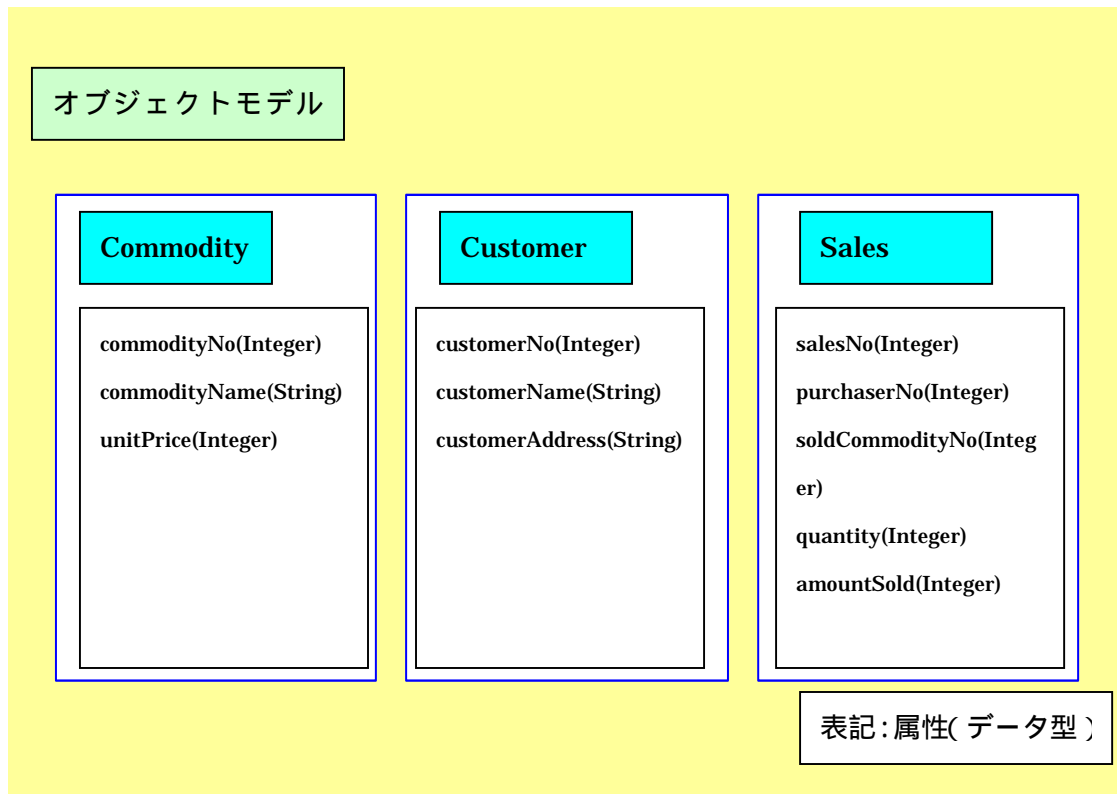


2002.12.13
Jasmine のメソッドの作成

学部4年 才野 大輔

以下のようなクラスを定義（資料を参考）し、データを挿入した。



● クラス定義

・ Customer クラスの定義

```
defineClass Customer
```

```
super:Composite
```

```
description:"Class Customer "
```

```
{
```

```
instance:
```

```
Integer customerNo unique: mandatory::
```

```
String customerName;
```

```
String customerAddress;
```

```
/*Customer クラスに定義した手続き ( addCustomer と deleteCustomer ) */
```

```
Void addCustomer(Integer i,String n,String a);
```

```
Void deleteCustomer(Integer dn);
```

```
};
```

• **Commodity** クラスの定義

```
defineClass Commodity
super:Composite
description:"Class Commodity"
{
  instance:
    Integer commodityNo unique: mandatory;;
    String commodityName;
    Integer unitPrice;
/*Commodity クラスに定義した手続き*/
    Void addCommodity(Integer i,String n,Integer u);
    Void udunitPrice(Integer c,Integer p);
};
```

• **Sales** クラスの定義

```
defineClass Sales
super:Composite
description:"Class Sales"
{
  instance:
    Integer salesNo unique: mandatory;;
    Integer purchaserNo;
    Integer soldCommodityNo;
    Integer quantity;
    Integer amountSold;
/*Sales クラスに定義した手続き*/
    Void addSales(Integer n,Integer pn,Integer sn,Integer q,Integer am);
};
buildClass Customer Commodity Sales; /*クラスの構築*/
```

● **データ**

• **Commodity** クラス

commodityNo	commodityName	unitPrice
1	"Apple"	30
2	"Orange"	20
3	"Grape"	40
4	"Banana"	10

5	"Mellon"	50
---	----------	----

• Customer クラス

customerNo	customerName	customerAddress
1	"Astore"	"Fukuoka"
2	"Bstore"	"Tokyo"
3	"Cstore"	"Osaka"

• Sales クラス

salesNo	purchaserNo	soldCommodityNo	quantity	amountSold
1	1	2	10	200
2	2	1	20	600
3	3	3	10	400
4	2	2	30	600
5	1	1	40	1200

- 商品の単価が変化し、その結果、Commodity クラスの unitPrice と Sales クラスの amountSold が変化するというメソッド (udunitPrice()) を作成した。

作成したメソッドおよび説明

```
defineProcedure Void Commodity::instance:udunitPrice(Integer c,Integer p)
language : "c"
{
/*デフォルトのクラスファミリ名を sainoCF にする*/
$defaultCF sainoCF;
/*Commodity クラスのオブジェクトの集合を格納するための変数 com の宣言*/
$Bag<Commodity> com;
/*Commodity クラスのオブジェクトを一つ格納するための変数 co の宣言*/
$Commodity co;
/*Sales クラスのオブジェクトの集合を格納するための変数 sal の宣言*/
$Bag<Sales> sal;
/*Sales クラスのオブジェクトを一つ格納するための変数 sa の宣言*/
$Sales sa;
/*Commodity クラスから属性 commodityNo が c であるオブジェクトを取り出し、変数 com に格納*/
$com = Commodity from Commodity where Commodity.commodityNo == c;
/* c に一致する commodityNo がなければエラーメッセージを返し、終了*/
$if (com.count() == 0){
```

```

odbGenerateUserMtdError("sainoCF", "Commodity", "udunitPrice", "The chosen
commodityNo doesn't exist!! Please choose another commodityNo.");
$return;
};
/* Commodity クラスから属性 commodityNo が c であるオブジェクトを取り出し、変数 co に
格納*/
$co = Commodity from Commodity where Commodity.commodityNo == c;
/*co に格納されたオブジェクトの unitoPrice の値を p に変更*/
$co.unitPrice = p;
/*Sales クラスから属性 soldCommodityNo が c であるオブジェクトを取り出し、変数 sal に
格納*/
$sal = Sales from Sales where Sales.soldCommodityNo == c;
/*もし c と一致する soldCommodityNo がなければ、終了*/
$if (sal.count() == 0){
$return;
};
/* Sales クラスから属性 soldCommodityNo が c であるオブジェクトを取り出し、変数 sal
に格納*/
$sal = Sales from Sales where Sales.soldCommodityNo == c;
/*sal に格納されたオブジェクトを一つずつ取り出して変数 sa に代入し、amountSold の値
を(p * quantity)に変更して終了*/
$scan(sal,sa){
$sa.amountSold = p * sa.quantity;
};
$return;
};

```

メソッドを書く上で苦労した点、工夫した点

● 苦労した点

- 定義したメソッドをコンパイルするためにはクラスを再構成し、作成したメソッドのファイルを `execFile` コマンドで実行、という手順を踏まなければならないの果たして正しいメソッドが書けているのかどうか調べるのに手間がかかった。
- 行の初めの “ \$ ” を忘れていたのになかなか気づかず、何度もやり直した。

● 工夫した点

- `Commodity` クラスの属性 `commodityNo` は `unique` であるが、`Sales` クラスの属性 `soldCommodityNo` は `unique` ではないうえに、`quantity` もオブジェクトによって

異なるので scan 文を利用した。「\$sa.amountSold = p * sa.quantity;」という書き方は可能なのかわからなかったがうまくいった。

実行結果

```
/*Commodity クラスのオブジェクトを全て allCommodity に格納*/
jasmine(sainoCF) > Bag<Commodity> allCommodity;
jasmine(sainoCF) > allCommodity = Commodity from Commodity;
/*Sales クラスのオブジェクトを全て allSales に格納*/
jasmine(sainoCF) > Bag<Sales> allSales;
jasmine(sainoCF) > allSales = Sales from Sales;
```

● 実行結果 1 (commodityNo が 1 である商品の unitPrice を 5 に変更する場合)

```
jasmine(sainoCF) > Commodity change;
jasmine(sainoCF) > change.udunitPrice(1,5);
jasmine(sainoCF) > allCommodity.print();
```

```
Bag{
  sainoCF::Commodity::1 {
    commodityNo = 1,
    commodityName = "Apple",
    unitPrice = 5
  },
  sainoCF::Commodity::2 {
    commodityNo = 2,
    commodityName = "Orange",
    unitPrice = 20
  },
  sainoCF::Commodity::3 {
    commodityNo = 3,
    commodityName = "Grape",
    unitPrice = 40
  },
  sainoCF::Commodity::4 {
    commodityNo = 4,
    commodityName = "Banana",
    unitPrice = 10
  }
}
```

← c=1,p=5 であるから
unitPrice が 30 から 5 に変更される

```
    },
    sainoCF::Commodity::5 {
        commodityNo = 5,
        commodityName = "Mellon",
        unitPrice = 50
    }
}
jasmine(sainoCF) > allSales.print();
Bag{
    sainoCF::Sales::1 {
        salesNo = 1,
        purchaserNo = 1,
        soldCommodityNo = 2,
        quantity = 10,
        amountSold = 200
    },
    sainoCF::Sales::2 {
        salesNo = 2,
        purchaserNo = 2,
        soldCommodityNo = 1,
        quantity = 20,
        amountSold = 100
    },
    sainoCF::Sales::3 {
        salesNo = 3,
        purchaserNo = 3,
        soldCommodityNo = 3,
        quantity = 10,
        amountSold = 400
    },
    sainoCF::Sales::4 {
        salesNo = 4,
        purchaserNo = 2,
        soldCommodityNo = 2,
        quantity = 30,
        amountSold = 600
    }
}
```

c=1,p=5 であるから
amountSold が 600 から $5 \times 10 = 100$
に変更される



```

},
sainoCF::Sales::5 {
  salesNo = 5,
  purchaserNo = 3,
  soldCommodityNo = 1,
  quantity = 40,
  amountSold = 200
}
}

```



c=1,p=5 であるから
amountSold が 1200 から $5 \times 40 = 200$ に
変更される

- **実行結果 2 (存在しない commodityNo を入力した場合)**

```

jasmine(sainoCF) > change.udunitPrice(7,30);
E_OD60C5_ODB_RNIEUSRERR (hif) ('sainoCF':'Commodity':'udunitPrice': 'The chosen
commodityNo doesn't exist!! Please choose another commodityNo.') のエラーメッセ
ージ .

```

commodityNo が 7 であるオブジェクトは存在しないのでエラー

- **実行結果 3 (変更された Commodity クラスのオブジェクトに対して Sales クラスのオブジェクトが参照していない場合)**

```

asmine(sainoCF) > change.udunitPrice(5,25);
jasmine(sainoCF) > allCommodity.print();
Bag{
  sainoCF::Commodity::1 {
    commodityNo = 1,
    commodityName = "Apple",
    unitPrice = 5
  },
  sainoCF::Commodity::2 {
    commodityNo = 2,
    commodityName = "Orange",
    unitPrice = 20
  },
  sainoCF::Commodity::3 {
    commodityNo = 3,
    commodityName = "Grape",

```

```
        unitPrice = 40
    },
    sainoCF::Commodity::4 {
        commodityNo = 4,
        commodityName = "Banana",
        unitPrice = 10
    },
    sainoCF::Commodity::5 {
        commodityNo = 5,
        commodityName = "Mellon",
        unitPrice = 25
    }
}
```



c=5,p=25 であるから
unitPrice が 50 から 25 に変更される

・ Sales クラスには変化なし

Sales クラスには、commodityNo が 5 である Commodity クラスのオブジェクトを参照しているオブジェクトはないので変化がない