- コンストラクタ
- クラス変数，クラスメソッド
- main メソッド

★7　クラスの作成（承前）

★7.5　コンストラクタ (p.70)

注: 古い刷の教科書では，p.70 の CStepper クラスの API 仕様中に「HTurtle クラスを拡張」という記述がありますが，これは誤りです．p.71 の `CStepper.java` を見るとわかるように，正しくは「Turtle クラスを拡張」です．

前回資料の ImpeiStepper クラスで隠蔽したインスタンス変数 `n,size` は，そのオブジェクトが描こうとしている多角形の辺の数と長さを表すものである．これらの値は，上記のようにメソッド経由で設定するのではなく，インスタンス生成時に設定することにしてもよい．その場合，次のように `n,size` の値を受け取ってインスタンスを生成する**コンストラクタ**を定義してやればよい．

復習: コンストラクタって何？

```
Turtle m = new Turtle();


Turtle m1 = new Turtle(100, 100, 45);
```

CStepper.java の一部

```
 1  public class CStepper extends Turtle{

 2     private int n;
 3     private int size;
 4     private int j = 0;

 5     public CStepper(int n, int size, int x, int y, int angle){
 6        super(x, y, angle);    //Turtle のコンストラクタ呼出し
 7        this.n = n;            //インスタンス変数 n に 引数 n を設定
 8        this.size = size;      //インスタンス変数 size の設定
 9     }

10     public CStepper(int n, int size){
11        this(n, size, 200, 200, 0); //上記のコンストラクタ呼出し
12     }

13     public void step() {
 :        // Stepper.java と同じ
       }
    }
```

Q1. 次のプログラムを CStepper クラスを用いるものに書きかえなさい．

```
Stepper m = new Stepper();
m.n = 6;  m.size = 200;
```

これまでの HTurtle や Stepper などのクラスでは，コンストラクタを定義しなかったのに，引数なしのコンストラクタを用いてインスタンスを生成することができた．これは，「コンストラクタの定義がなければ引数なしのコンストラクタが自動的に定義される」からである．この辺りの詳細は教科書 p.71 参照．

★7.6 クラス変数，クラスメソッド (p.72)

以前「クラス変数とクラスメソッド」のところで説明したように，クラス変数とクラスメソッドは，API 仕様の該当項目の先頭に static という修飾子がついている．実際のクラス定義の中でも，static をつけて宣言した変数はクラス変数となり，static をつけて定義したメソッドはクラスメソッドとなる．

Q2. Stepper クラスのインスタンス変数 n は，Stepper.java の中で public int n; と宣言されています．これをクラス変数の宣言にするにはどう書きかえたらよいですか．

次の Shop.java は，main とクラスメソッドのみから成るプログラムの例である（アクセス修飾子 public をつけない例にもなっている）．このようなクラスメソッドの使い方は C 言語の関数の使い方と似ている．

Shop.java

```
 1  class Shop {
 2      static double ritsu = 1.8;          // クラス変数
 3
 4      static int urine(int shiirene){     // クラスメソッド
 5          return (int)(ritsu * shiirene);
 6      }
 7
 8      public static void main(String[] args){
 9          int x = 2000, y;
10          y = Shop.urine(x);   // urine メソッドの呼び出し．"Shop."は省略可
11          System.out.println("仕入値: " + x + "円，売値: " + y + "円");
12          Shop.ritsu = 1.2;   // 変数値の変更．"Shop."は省略可
13          y = Shop.urine(x);
14          System.out.println("仕入値: " + x + "円，売値: " + y + "円");
15      }
16  }
```

このように，インスタンス変数／メソッドの場合は this.[変数名／メソッド名] と書くところを，クラス変数／メソッドの場合は[クラス名].[変数名／メソッド名] と書く．「this.」を省略できる場合があるのと同様に，あるクラスを定義したソースの中でそのクラスのクラス変数やクラスメソッドにアクセスする場合には，「[クラス名].」は省略できる．

次に，上述の Shop クラスを別のプログラムから利用する例を示す．

Maido.java

```
1  public class Maido{
2      public static void main(String[] args){
3          Shop.ritsu = 100.0; // ぼったくり
4          int Tsubo = 100, Tawashi = 5;
5          System.out.println("いらっしゃ〜い．");
6          System.out.println("壷は" + Shop.urine(Tsubo) + "円，束子は"
                              + Shop.urine(Tawashi) + "円でっせ．");
7          System.out.println("え，まけろて？かなんな〜");
8          Shop.ritsu = 80.0; // あこぎ
9          System.out.println("よっしゃ，" + Shop.urine(Tsubo) + "円と"
                              + Shop.urine(Tawashi) + "円でええわ．");
10         System.out.println("はいよ，まいどおおきに．");
11     }
12 }
```

★7.7 main メソッド (p.73)

Java のプログラムを実行するには，java コマンドを java Hoge のように実行する (☆ 1)．こうすると，java コマンドは Hoge.class というクラスファイルを探し出してこれを実行する (☆ 2)．java コマンドで起動するプログラムの中には，次のような形で main を必ず書く必要がある．

```
public static void main(String[] args){ ... }
```

これは「main は戻り値なしのクラスメソッドであり公開されている」 ということを意味している．main メソッドは public でなければならない．一方，java コマンドで直接起動されていないクラスファイルの中に main メソッドがあっても，明示的に呼び出されない限りはそれは無視される．

例えば，前節の Shop クラスと Maido クラスの例では，Maido クラスだけでなく，Maido クラスが利用している Shop クラス自身にも main メソッドが定義されている．この場合，java Maido と実行すれば Maido クラスの main メソッドが実行され，java Shop と実行すれば Shop クラスの main メソッドが実行される．

☆ 1) java Hoge 123 abc のように引数をつけて実行することもできるのだった．第 6 回「main の引数」参照．

☆ 2) これまで扱ってきたプログラムのようにクラス定義に public をつけている場合，Hoge.class のもとになっているのは Hoge.java というソースファイルである．クラスの定義に public をつけない場合，クラスの名前とそれを記述したソースファイルの名前を一致させる必要がない（一つのソースに複数のクラス定義があってもよい）ので，Hoge.class のソースは Fuga.java という名前かもしれない．その場合，javac Fuga.java すると Hoge.class ができる．

宿題？

- 授業中に解説されなかった Q を解いてみよう．
- 7 章の残り（内部クラス，匿名クラス）と 8 章から 10 章の内容はこの授業では説明しません．次回は 11 章の予定．