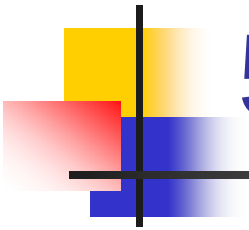
クラスの作成

5.1 ： クラスの一般形式

5.2 ： 簡単なクラスの作成

5.3 ： コンストラクタの追加

5.4 ： コンストラクタのオーバーロード



5.1：クラスの一般形式

- Javaプログラムで実行される全てのプログラム活動はクラスの内部で行われる。

- 「クラス」とは・・・

  プロジェクトを定義する手段。

クラスのメンバ

- 変数

・クラスタの状態を表す。

- メソッド

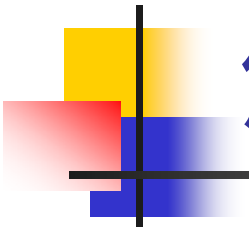
・クラスによって定義された動作を構成するロジックを提供する。（静的変数と静的メソッド・
インスタンス変数とインスタンスメソッド）

- コンストラクタ

・クラスの新しいインスタンスの状態を初期化する特殊なメソッド。

簡単なクラス宣言例

- class clsName {
- type1 varName1 = value1;　//インスタンス変数の宣言
- …
- typeN varNameN = valueN;
- clsName(cparams1) {　//コンストラクタ
- //コンストラクタの本体
- }
- …
- clsName(cparamsN){
- //コンストラクタの本体
- }
- rtype1 mthName1(mparams1) {　//メソッド
- //メソッドの本体
- }
- …
- rtypeN mthNameN(mparamsN){
- //メソッドの本体
- }
- }



解説①

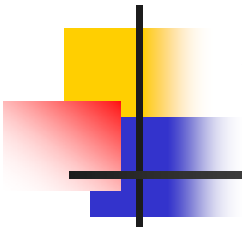
インスタンス変数の宣言

・classキーワードはclsNameという名のクラスを宣言する事を意味する。

・通常の変数宣言構文を使用し、vaName1~varNameNをクラスに定義。

・各変数には、type1~typeNで示した型を割り当て、value1~valueNで示した値を初期値として設定。

・変数に初期値を与えなくてもよい。



解説②

- コンストラクタ
  - 常にクラスと同じ名前でなければならない。
  - 戻り値がない。
  - cparams1~ccparamsNは省略可能なパラメータリスト。

- メソッド
  - mthName1~mthNameNという名のメソッド。
  - メソッドの戻り値の型はrtype1~rtypeN。
  - 省略可能なパラメータリストはmparams1~mparamsN。

5.2 : 簡単なクラスの作成

```
class  sample {
 int a;
 int b;
 int c;
}
```

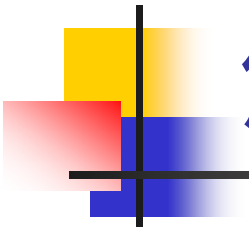
- オブジェクトを作成する方法

```
sample one = new sample();
sample two = new sample();
```

・sampleクラスで定義された通りのオブジェクトが2つ生成。
・変数oneには一方のオブジェクトへの参照が格納される。
・変数twoにはもう一方のオブジェクトへの参照が格納される。

クラスの作成例

```
class Point3D {
  double x,y,z;
}
class Point3DExample {
  public static void main(String args[]){
    Point3D p = new Point3D();      //Point3Dのクラスのインスタンス作成
    p.x = 1.1;
    p.y = 3.4;
    p.z = -2.8;
    System.out.println("p.x = " + p.x);
    System.out.println("p.y = " + p.y);
    System.out.println("p.z = " + p.z);
  }
}
```



解説

- Point3D

・クラスのインスタンスには、空間のある１点を表す座標がカプセル化される。

・Point3Dクラスにはx,y,zという3つのdouble型インスタンス変数がある。

- Point3DExample

・main()という静的メソッドを宣言。

・new演算子を使用し、Point3Dクラスのインスタンスを作成。

・このオブジェクトへの参照はローカル変数pに代入。

・3つのステートメントでオブジェクトのインスタンス変数に値を代入。

・println()メソッド使用し、変数の値を表示。

コンストラクタの追加

- クラスのオブジェクトは作成するのと同時にオブジェクトを初期化せねばならない場合がある。

  ⇒クラスのコンストラクタを定義する機能

- コンストラクタとは

  ・特定のクラスのオブジェクトを作成して初期化するメソッド。
  ・コンストラクタの名前はクラス名と同じ。
  ・引数を受け取る事が出来る。
  ・コンストラクタから戻り値を返すことが出来ない。
  ・コンストラクタは常にnew演算子を使って呼び出す。

コンストラクタの追加例

```
class Point3D {
  double x,y,z;
  Point3D(double ax,double ay,double az) {
    x = ax;
    y = ay;
    z = az;
  }
}
class Point3DConstructor {
  public static void main(String args[]) {
    Point3D = new Point3D(1.1,3.4,-2.8);  //3つの引数をコンストラクタに指定し、
                                          オブジェクトを作成
    System.out.println("p.x = " + p.x);
    System.out.println("p.y = " + p.y);
    System.out.println("p.z = " + p.z);
  }
}
```

コンストラクタのオーバーロード

- コンストラクタのオーバーロードとは
  ・クラスに複数のコンストラクタを用意できる事。
  ・名前は全てクラス名と同じ（シグネチャが異なる）。

- コンストラクタのシグネチャとは
  ・その名前とパラメータ方のリストを合わせたもの。
  ・同じシグネチャのコンストラクタが同クラスにあると、コンパイラは判断できないので、エラーが表示される。

コンストラクタのオーバーロードの特徴・利点

- 特徴

  ・特定のクラスから数どおりのオブジェクトを作成できる。

- 利点

  ・オブジェクトの作成と初期化に様々なデータ型を使用可能。

  ⇒データ型を変換する手間をコンストラクタに任せられる。

コンストラクタのオーバーロードの使用例①

```
class Point3D {
  double x,y,z;
//シグネチャの異なる3つのコンストラクタを定義
  Point3D(double ax){
    x = ax;
    y = 1;
    z = 1;
  }
  Point3D(double ax,double ay) {
    x = ax;
    y = ay;
    z = 1;
  }
  Point3D(double ax,double ay,double az) {
    x = ax;
    y = ay;
    z = az;
  }
}
```

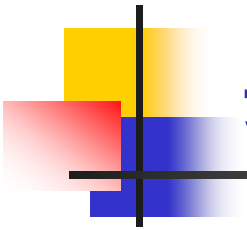
コンストラクタのオーバーロードの使用例②

```
class Point3DOverLoadConstructors  {
  public static void main(String args[]) {

    Point3D p1 = new Point3D(1.1);
    System.out.println("p1.x = " + p1.x);
    System.out.println("p1.y = " + p1.y);
    System.out.println("p1.z = " + p1.z);

    Point3D p2 = new Point3D(1.1,3.4);
    System.out.println("p2.x = " + p2.x);
    System.out.println("p2.y = " + p2.y);
    System.out.println("p2.z = " + p2.z);

    Point3D 3 = new Point3D(1.1,3.4,-2.8);
    System.out.println("p3.x = " + p3.x);
    System.out.println("p3.y = " + p3.y);
    System.out.println("p3.z = " + p3.z);
  }
}
```



実行結果

- p1.x = 1.1
- p1.y = 1.0
- P1.z = 1.0
- p2.x = 1.1
- p2.y = 3.4
- P2.z = 1.0
- p3.x = 1.1
- p3.y = 3.4
- P3.z = -2.8

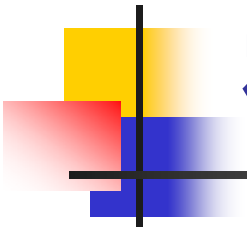
解説（手順）

①このプログラムはPoint3Dに3つのコンストラクタを用意。

②1つ目のコンストラクタでは、double型の引数を1つ受け取る。この値は、インスタンス変数xに初期値として代入。変数yと変数zには初期値として1を代入。

③2つ目のコンストラクタでは、double型の引数を2つ受け取る。この値は変数xと変数yに初期値として代入。変数zには初期値として1を代入。

④3つ目のコンストラクタでは、double型の引数を3つ受け取る。この値は3つのインスタンス変数に初期値として代入。



宿題

1．Personというクラスを宣言するアプリケーションを作成しなさい。このクラスのインスタンス変数には名前・年齢・収入を記録し、データ型は順に、String,int,floatを使用しなさい。