クラスの継承

クラスを再利用する方法

- 作成したクラスを拡張できる
  - 既存クラスにメンバ変数とメンバ関数を追加
  - 拡張とは，クラスの性質を「継承する」こと
- 拡張される元のクラスは「基本クラス」
  - スーパークラス，親クラスとも呼ばれる
- 拡張された新たなクラスは「派生クラス」
  - サブクラス，子クラスとも呼ばれる
  - 派生クラスは，通常は基本クラスのメンバを参照する（参照の可否はアクセス指定子で制御する）
  - アクセス指定子は，多くの場合で公開 (public) が宣言される
- 複数のクラスを基本クラスとする事ができる
  - 「多重継承」と呼ばれる（Java ではできない）

派生クラスの作成書式

宣言形式

```
class 派生クラス名 : アクセス指定子　基本クラス名　{
    派生クラスに追加するメンバ変数，メンバ関数宣言
};
```

実装例

```
class Point3D : public Point { // 2次元点を3次元点に拡張
public:
    int z; // 3次元目の座標軸（x, y は Point クラスで宣言済み）
    Point3D() : Point() { z = 0; } // 引数無しの構築子
    Point3D(int _x, int _y, int _z) : Point (_x, _y); //引数有
    Point3D* median(Point3D p); // 中点の計算
};
```

派生クラスの構築子

```
派生クラス名（引数リスト）：基本クラスの構築子呼び出し（引数リスト）{
    拡張する処理
}
```

- 構築子にのみ，適用される書式
- 基本クラスの構築子は，最初に実行される
- 基本クラスの構築子は，呼び出さなくても良い

実装例：

```
Point3D (int _x, int _y, int _z) : Point (_x, _y) {
    z = _z; //x = _x; y = _y;は Point(_x, _y) 内で実行
}
```

メンバ変数のアクセス指定

基本クラスでの宣言

```
class Point {
protected: // 以下のメンバを派生クラスにのみ公開
    int x, y; // メンバ変数
public:  // 以下のメンバを全クラスに公開
    Point () { x = y = 0; } …
```

派生クラスでの宣言

```
class Point3D : public Point {
    int z; // x, y, z がメンバ変数となる
public:  // 以下のメンバを全クラスに公開
    Point3D () : Point () { z = 0; } …
```

フレンド（friend）クラス

- アクセス指定子（public, private, protected）とは無関係にアクセスできるクラス

```
class Point {
      friend class Triangle;   //Triangleクラスからのアクセス許可
protected: // メンバ変数は継承クラスにだけ公開する
      int x, y;
public:
      Point () { x = y = 0; } …
```

Triangle.h（または，Triangle.cpp）での使用例

```
class Triangle {
public:
      Point pnt[3]; // 3頂点のデータ
      Triangle () { // Point クラスのメンバ変数に代入可能
            pnt[0].x = pnt[0].y = 1;
       pnt[1].x = 1; pnt[1].y = 0;
      …}
```

クラスの消滅方法

- クラスには消滅子を実装できる
- ～クラス名() { 消滅する際の処理 } と記述する

  ~Point3D() { std::cout << x << “,” << y << “,” << z << “が消滅…”;}
- delete オブジェクトのポインタ変数; で陽に呼び出される
- 消滅子も派生クラスで継承される
  - delete 派生クラス変数; でまず派生クラス消滅子が呼び出され，次に基本クラスの消滅子が呼び出される 例 : ~Point () { std::cout << “基本消滅¥n”; }

```
int main () {
      Point3D pointA (1,2,3), pointB(4,6,8);
      Point3D *med = pointA.median(pointB);
      delete med;
}
```

2,4,5が消滅 … 基本消滅
1,2,3が消滅 … 基本消滅
4,6,8が消滅 … 基本消滅

medの消滅
pointA の消滅
pointB の消滅

これらの変数は，main 関数の終了に伴い，自動消滅する