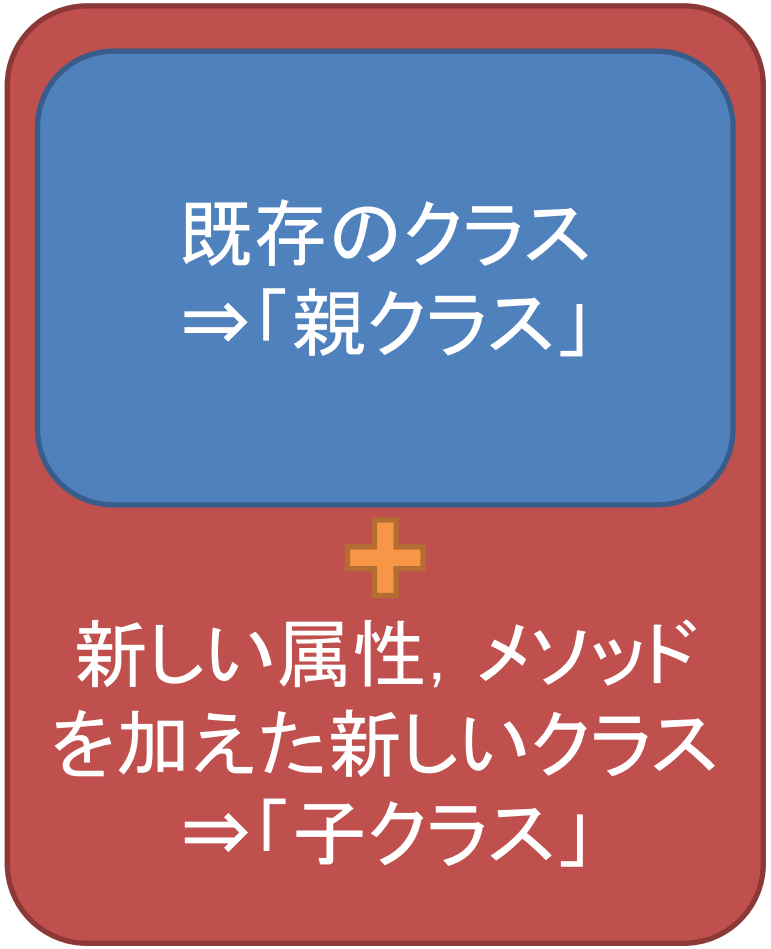
プログラミング基礎

第10.5回

継承復習・補足

継承



すでにあるクラスを使いまわして，
新しいクラスを作る
⇒無駄なプログラムをなくす

書き方

```
class 子クラス extends 親クラス
{
    …
}
```

注意

親クラスと子クラスはis-a関係
でなければならない
例：（親）電卓-（子）関数電卓⇒○
（親）CD-（子）CDプレイヤー⇒×

superの使い所

- 子クラスのコンストラクタ内で，
  親クラスのコンストラクタを使いたい時

- オーバーライドしたメソッドで
  親クラスの同名メソッドを使いたい時

superの使い所

- 子クラスのコンストラクタ内で，
  親クラスのコンストラクタを使いたい時

```
class FIStudent extends TDUStudent
{
   int cg;

   FIStudent(String n)
   {
      super(n);
   }
}
```

```
class TDUStudent
{
   String name;
   int english;

   TDUStudent(String n)
   {
      name = n;
   }

   String getName()
   {
      return name;
   }
}
```

superの使い所

- オーバーライドしたメソッドで
  親クラスの同名メソッドを使いたい時

```
class FIStudent extends TDUStudent
{
   int cg;

   FIStudent(String n)
   {
      super(n);
   }

   String getName()
   {
       double str = super.getName();
       return str + "FI";
   }
}
```

```
class TDUStudent
{
   String name;
   int english;

   TDUStudent(String n)
   {
      name = n;
   }

   String getName()
   {
      return name + ":TDU" + ":" ;
   }
}
```

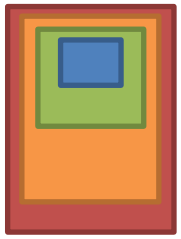
superが不要な場合

```
class FIStudent extends TDUStudent
{
   int cg;

   FIStudent(String n)
   {
      super(n);
   }

   String getName()
   {
       double str = super.getName();
       return str + "FI";
   }

   int averageScore()
   {
      int e = getEnglishScore();
      iny a = (e + cg) / 2;
      return a;
   }
}
```

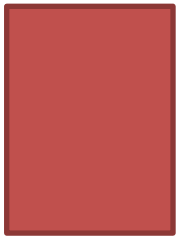
```
class TDUStudent
{
   String name;
   int english;

   TDUStudent(String n)
   {
      name = n;
   }

   String getName()
   {
      return name + ":TDU" + ":" ;
   }

   int getEnglishScore()
   {
      return english;
   }
}
```

補足

- 本当に継承が必要なのか検討する



プログラムの無駄を
省くため，とても小さい
機能を親クラスにして
何重にも拡張していくが…



結局，小さい機能しかない
親クラスを使いまわすことが
ないので，
1つのクラスにしたほうが
見やすかった…など

- 包含（has-a関係）でも同様のことができないか検討する