

Fig.0-1 クラスの階層関係を表したクラス図

```
abstract class ParentClass {
    int char field2;
    static char field2 ;
    abstract void methodA
    double methodB () {
        // ...
    }
}
```

```
class ChildClass extends ParentClass {
    void methodA() {
        // ...
    }
    static void methodC () {
        // ...
    }
}
```

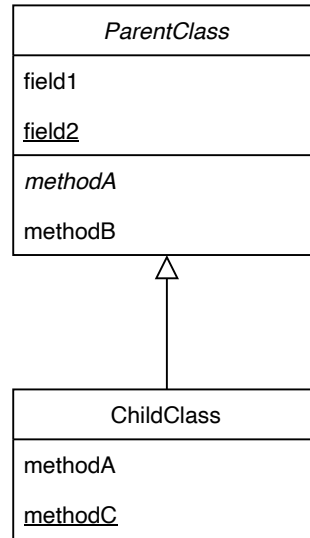


Fig.0-2 インターフェースと実装クラスを表したクラス図

```
interface Printable {
    abstract void print ();
    abstract void newPage()
}
```

```
class PrintClass implements Printable {
    void print () {
        // ...
    }
    void newPage () {
        // ...
    }
}
```

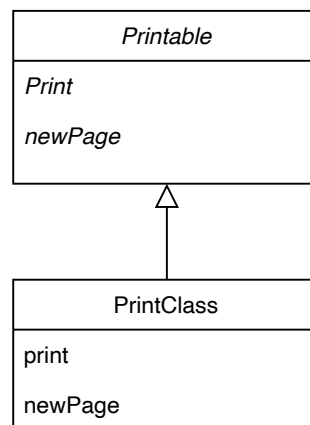


Fig.0-3 集約を表したクラス図

```
class Color {
    // ...
}

class Fruit {
    Color color {
        // ...
    }
}

class Basket {
    Fruit [] fruits;
    // ...
}
```

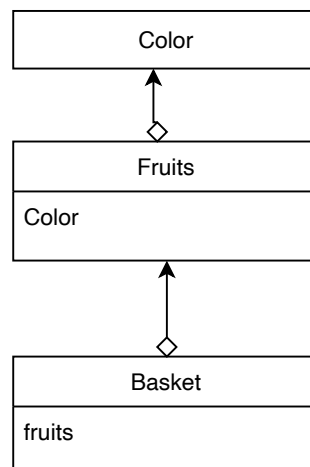


Fig.0-4 アクセス制御を明示したクラス図

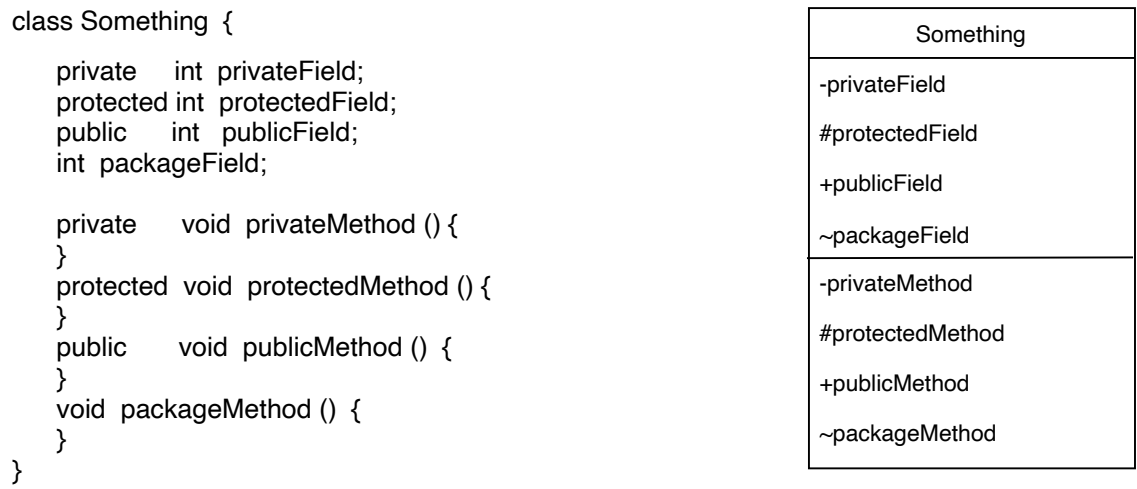


Fig.0-5 クラスの関連

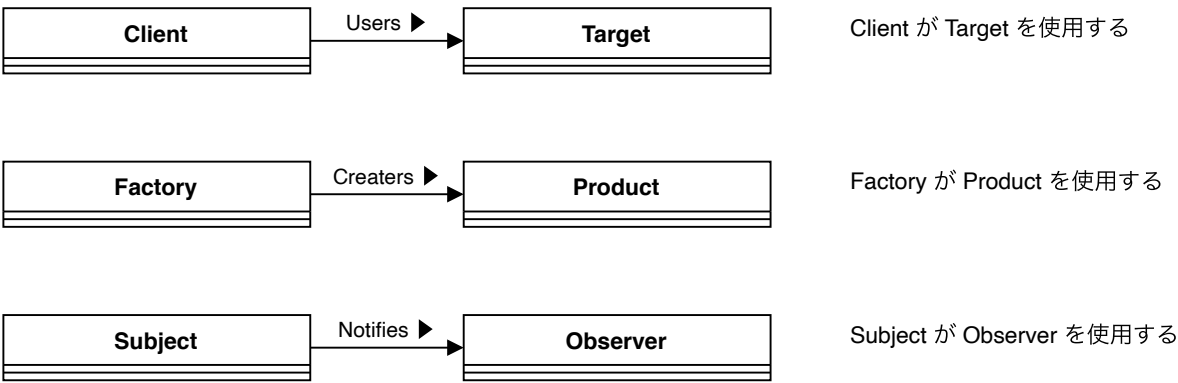


Fig.0-6 シーケンス図の例（メソッドの呼び出し）

```
class Client {
  Server server;
  void work() {
    server.open();
    server.print ("Hello");
    server.close();
  }
  // ...
}

class Server {
  Device device;
  void open() {
    // ...
  }
  void print (String s) {
    device.write(s);
    // ...
  }
  void close() {
    // ...
  }
  // ...
}

class Device {
  void write(String s) {
    // ...
  }
}
```

