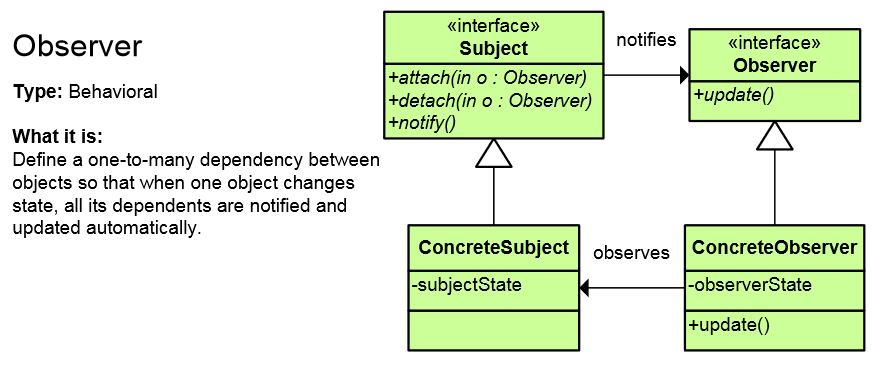
Observer (наблюдател) Шаблон

Дизайн-шаблонът „наблюдател” се използва за връзка между няколко класа без да има пряка връзка между тях и без те да знаят взаимно за съществуването си като по този начи се запазва възможно най-„свободно свързване”(**loose coupling**) и възможността да се ползват различни методи обработващи едно и също състояние.

Използва се така нереченото „**събитие**” – което е състояние породено по време на изпълнение на програма, което тя може да обработи.

Обикновено става въпрос за събития предизвикани асинхронно (независимо от хода на програмата) – например от потребител натискащ бутон ( на клавиатурата, мишката или в някакъв вид форма). Тогава Обектът следящ за тези действия изстрелва и този обект (или обекти), който е „абониран” за това събитие го получава и задейства своята логика.



В .NET този шаблон се използва за така нареченото „справяне със събития” (event handling).

**delegate** **void** Notifier (**string** sender);

**class** Model {

**public** **event** Notifier notifyViews;

**public** **void** Change() { ... notifyViews("Model"); }

}

**class** View1 {

**public** View1(Model m) {

m.notifyViews += new Notifier(**this**.Update1);

}

**void** Update1(**string** sender) {

Console.WriteLine(sender + " was changed during update");

}

}

**class** View2 {

**public** View2(Model m) {

m.notifyViews += new Notifier(**this**.Update2);

}

**void** Update2(**string** sender) {

Console.WriteLine(sender + " was changed");

}

}

**class** Test {

**static** **void** Main() {

Model m = new Model();

new View1(m);

new View2(m);

m.Change();

}

}