



### Подклассы WindowImp

Подклассы WindowImp преобразуют запросы в операции, характерные для конкретной оконной системы. Рассмотрим пример из раздела 2.2. Мы определили `Rectangle::Draw` в терминах операции `DrawRect` над экземпляром класса `Window`:

```
void Rectangle::Draw (Window* w) {
    w->DrawRect(_x0, _y0, _x1, _y1);
}
```

В реализации `DrawRect` по умолчанию используется абстрактная операция рисования прямоугольников, объявленная в `WindowImp`:

```
void Window::DrawRect (
    Coord x0, Coord y0, Coord x1, Coord y1
) {
    _imp->DeviceRect(x0, y0, x1, y1);
}
```

где `_imp` — переменная-член класса `Window`, в которой хранится указатель на объект `WindowImp`, использованный при конфигурировании `Window`. Реализация окна определяется тем экземпляром подкласса `WindowImp`, на который указывает `_imp`. Для `XWindowImp` (то есть подкласса `WindowImp` для оконной системы X Window System) реализация `DeviceRect` могла бы выглядеть так:

```
void XWindowImp::DeviceRect (
    Coord x0, Coord y0, Coord x1, Coord y1
) {
    int x = round(min(x0, x1));
    int y = round(min(y0, y1));
    int w = round(abs(x0 - x1));
    int h = round(abs(y0 - y1));
    XDrawRectangle(_dpy, _winid, _gc, x, y, w, h);
}
```

DeviceRect определено именно так, поскольку XDrawRectangle (функция X Windows для рисования прямоугольников) определяет прямоугольник с помощью левого нижнего угла, ширины и высоты. DeviceRect должна вычислить эти значения по переданным ей параметрам. Сначала находится левый нижний угол (поскольку (x0, y0) может обозначать любой из четырех углов прямоугольника), а затем вычисляется длина и ширина.

Подкласс PMWindowImp (подкласс WindowImp для Presentation Manager) определил бы DeviceRect по-другому:

```
void PMWindowImp::DeviceRect (
    Coord x0, Coord y0, Coord x1, Coord y1
) {
    Coord left = min(x0, x1);
    Coord right = max(x0, x1);
    Coord bottom = min(y0, y1);
    Coord top = max(y0, y1);

    PPOINTL point[4];

    point[0].x = left; point[0].y = top;
    point[1].x = right; point[1].y = top;
    point[2].x = right; point[2].y = bottom;
    point[3].x = left; point[3].y = bottom;

    if (
        (GpiBeginPath(_hps, 1L) == false) ||
        (GpiSetCurrentPosition(_hps, &point[3]) == false) ||
        (GpiPolyLine(_hps, 4L, point) == GPI_ERROR) ||
        (GpiEndPath(_hps) == false)
    ) {
        // сообщить об ошибке
    }
    else {
        GpiStrokePath(_hps, 1L, 0L);
    }
}
```

Откуда такое отличие от версии для X? Дело в том, что в Presentation Manager (PM) нет явной операции для рисования прямоугольников, как в X. Вместо этого PM имеет более общий интерфейс для задания вершин фигуры, состоящей из нескольких отрезков (множество таких вершин называется *траекторией*), и для рисования границы или заливки той области, которую эти отрезки ограничивают.

Очевидно, что реализации DeviceRect для PM и X совершенно непохожи, но это не имеет никакого значения. Возможно, WindowImp скрывает различия интерфейсов оконных систем за большим, но стабильным интерфейсом. Это позволяет автору подкласса Window сосредоточиться на абстракции окна, а не на деталях оконной системы. Заодно мы получаем возможность добавлять поддержку для новых оконных систем, не изменяя классы из иерархии Window.