

Конфигурирование класса *Window* с помощью *WindowImp*

Важнейший вопрос, который мы еще не рассмотрели, – как сконфигурировать окно с помощью подходящего подкласса *WindowImp*. Другими словами, когда инициализируется переменная *_imp* и как узнать, какая оконная система (следовательно, и подкласс *WindowImp*) используется? Ведь окну необходим объект *WindowImp*.

Тут есть несколько возможностей, но мы остановимся на той, где используется паттерн абстрактная фабрика. Можно определить абстрактный фабричный класс *WindowSystemFactory*, предоставляющий интерфейс для создания различных видов системно-зависимых объектов:

```
class WindowSystemFactory {
public:
    virtual WindowImp* CreateWindowImp() = 0;
    virtual ColorImp* CreateColorImp() = 0;
    virtual FontImp* CreateFontImp() = 0;

    // операции "Create..." для всех видов ресурсов оконной системы
};
```

Далее разумно определить конкретную фабрику для каждой оконной системы:

```
class PMWindowSystemFactory : public WindowSystemFactory {
    virtual WindowImp* CreateWindowImp()
    { return new PMWindowImp; }
    // ...
};

class XWindowSystemFactory : public WindowSystemFactory {
    virtual WindowImp* CreateWindowImp()
    { return new XWindowImp; }
    // ...
};
```

Чтобы инициализировать член *_imp* указателем на объект *WindowImp*, соответствующий данной оконной системе, конструктор базового класса *Window* может использовать интерфейс *WindowSystemFactory*:

```
Window::Window () {
    _imp = windowSystemFactory->CreateWindowImp();
}
```

Переменная *windowSystemFactory* – это известный программе экземпляр подкласса *WindowSystemFactory*. Она, аналогично переменной *guiFactory*, определяет внешний облик. И инициализировать *windowSystemFactory* можно точно так же.

Паттерн мост

Класс *WindowImp* определяет интерфейс к общим средствам оконной системы, но на его дизайн накладываются иные ограничения, нежели на интерфейс