

но добиться баланса в этом случае можно и с помощью создания элементов в едином стиле, использования принципов модульной сетки и иерархий.

От того, насколько правильно построена композиция, зависит насколько «приятным» будет конечный продукт. Чтобы прийти к этому, необходимо соблюдать основные правила и принципы создания композиции при выборе цвета и расположении элементов, учитывать актуальные тенденции. При хорошей композиции дизайн будет работать, ориентировать и побуждать человека к необходимому действию.

УДК 004

Каврук В.А.

ИСПОЛЬЗОВАНИЕ ШАБЛОНОВ В ПРОГРАММИРОВАНИИ

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Научный руководитель: канд. техн. наук, доцент Дробыш А.А.

Каждый программист может вполне успешно работать, не зная ни одного шаблона проектирования. Но это не означает, что он никогда не реализовывал ни один из них. Программист может использовать шаблоны, даже не подозревая об этом. Однако именно осознанное владение инструментом как раз и отличает профессионала от любителя.

Шаблон проектирования (паттерн) – это неоднократно встречающееся решение какой-то определённой проблемы при проектировании архитектуры программ.

Паттерн нельзя просто скопировать в программу, так как он представляет собой общую концепцию решения проблемы, а не какой-то конкретный код.

Паттерны часто путают с алгоритмами, ведь оба понятия описывают типовые решения каких-то известных проблем. Но если алгоритм – это чёткий набор действий, то паттерн – это высокоуровневое описание решения, реализация которого может отличаться в двух разных программах.

Описания паттернов обычно очень формальны и в основном состоят из следующих пунктов:

- проблемы, которую решает паттерн;
- мотивации к решению проблемы способом, который предлагает паттерн;
- структуры классов, составляющих решение;
- примера на одном из языков программирования;
- особенностей реализации в различных контекстах;
- связей с другими паттернами.

Шаблоны проектирования отличаются по уровню сложности, детализации и охвата проектируемой системы.

Самые низкоуровневые и простые паттерны – идиомы. Они могут применяться в рамках только одного языка программирования, поэтому они являются не универсальными.

Самые универсальные – архитектурные паттерны, их можно использовать практически в любом языке программирования. Такие шаблоны необходимы для проектирования всей программы, а не отдельных её элементов.

Шаблоны также отличаются и предназначением. Ниже рассмотрены три основные группы паттернов:

1. Порождающие паттерны беспокоятся о гибком создании объектов без внесения в программу лишних зависимостей.

1.1. Простая фабрика (Simple Factory) – это функция или метод, который возвращает объекты изменяющегося прототипа или класса из некоторого вызова метода, который считается «новым»).

1.2. Одиночка (Singleton) – порождающий шаблон проектирования, гарантирующий, что в однопроцессном приложении будет единственный экземпляр некоторого класса, и предоставляющий глобальную точку доступа к этому экземпляру.

1.3. Строитель (Builder) – порождающий шаблон проектирования, который предоставляет способ создания составного объекта.

2. Структурные паттерны показывают разнообразные способы построения связей между объектами.

2.1. Адаптер (Adapter) – структурный шаблон проектирования, предназначенный для организации использования функций объекта, недоступного для модификации, через специально созданный интерфейс.

2.2. Мост (Bridge) – структурный шаблон проектирования, используемый в проектировании программного обеспечения.

3. Поведенческие паттерны заботятся об эффективной коммуникации между объектами.

3.1. Посредник (Mediator) – поведенческий шаблон проектирования, обеспечивающий взаимодействие множества объектов, формируя при этом слабую связанность, и избавляя объекты, от необходимости явно ссылаться друг на друга.

3.2. Хранитель (Memento) реализуется тремя объектами: «Создателем» (originator), «Опекуном» (caretaker) и «Хранитель» (memento).

В современном мире программирования существует множество технологий и шаблонов и подходов к разработке программного продукта. Используя эти шаблоны проектирования, всегда следует проверять, что решается правильно поставленная задача. Шаблоны проектирования необходимо использовать очень осторожно: они – при использовании в ненадлежащем контексте – могут ухудшить ситуацию, но при правильном использовании они просто жизненно необходимы.

УДК 004.652.4

Каврук В.А.

НЕОБХОДИМОСТЬ ИСПОЛЬЗОВАНИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

*Белорусский национальный технический университет,
г. Минск, Республика Беларусь*

Научный руководитель: ст. преподаватель Липень С.Г.

Основными идеями современных информационных технологий является концепция о том, что все данные должны быть организованы в базы данных. Это делается для того, чтобы была возможность адекватно отображать изменяющийся реальный мир и в полном объеме удовлетворить информационные потребности пользователей.

Развитие реляционных баз данных началось в конце 60-х годов, когда появились первые работы, в которых обсуждались; возмож-