

## ПРИНЦИПЫ ТЕОРИИ КОНТЕЙНЕРОВ В ПРИМЕНЕНИИ К СОЗДАНИЮ ГИБКИХ ПРОГРАММНЫХ СИСТЕМ

Прыгунков М.О.

*Волгоградский государственный технический университет*

*Волгоград, Россия*

Современные программные системы становятся сложнее, претендуя на решение глобальных задач, например таких, как создание единой системы управления предприятием. При этом автоматизация отдельных операций или отделов фактически исчерпала свой потенциал, а возможность безболезненного объединения нескольких подсистем в единое целое, как правило, требует значительных усилий.

Гибкость программного обеспечения определяется набором типовых частей, из которых можно было бы собирать множество программ.

Объектно-ориентированные языки программирования дали четыре новых механизма использования типовых частей:

§ механизм классов, порождающих при выполнении любое количество однотипных объектов, например, ряд однотипных кнопок;

§ возможность тиражирования объектов от породившей программы во все новые программы;

§ динамически линкуемые библиотеки с порождающими объектами классами;

§ механизм сборки программ из компонентов – объектов в процессе выполнения программ.

Первый механизм облегчил развитие систем визуального программирования, при работе в которых, значительная часть программы может быть создана путем отбора мышкой стандартных компонентов.

Второй механизм привел к возникновению объектных систем управления базами данных (СУБД), поставляющих программам не только данные, но и код, обрабатывающий эти данные.

На основе третьего механизма возникли СОМ-технологии, что позволило строить гибкие программы, которые обладают свойством возможного развития при изменении условий их эксплуатации.

Четвертый механизм пока не нашел должной поддержки в инструментальных средствах разработки программ, что и предопределило направление данных исследований.

Введём два вида контейнеров: динамический и статический.

Если контейнер для «общения» с вложенными объектами общается через единый интерфейс, то такой контейнер будем называть статическим. Если контейнер работает со всеми вложенными объектами не по одной схеме, а используя разные методы, то такой контейнер будем называть динамическим.

Чтобы приведённая классификация контейнеров стала более понятной, стоит оговориться о том, что разделение контейнеров на статические и динамические относится к строению контейнера.

Контейнеры по определению могут общаться с другими контейнерами, с которыми находится на одном уровне иерархии, а также на одну ступень выше и ниже по иерархической лестнице. Соответственно, сообщение может быть внутренним относительно контейнера – ситуация, когда контейнер передаёт сообщения между своими компонентами, и внешним – ситуация, когда контейнер передаёт сообщения другим контейнерам или контейнеру, в состав которого входит источник сообщения.

Для обеспечения заявленной возможности замены (добавления/удаления) новых свойств и методов принято решение хранить в контейнерах только ссылки на свойства и методы.

Помимо такой программной сущности, как контейнер, необходимо внедрить модули-диспетчеры – виртуальные таблицы, в которых содержатся ссылки на экземпляры свойств и методы, обращаясь к которым объект-контейнер фактически агрегирует их в своей структуре.

**Заключение**

Применение контейнеров для построения крупных программных систем, которые должны обладать возможностью быстрого изменения исходя из требований бизнеса, предоставляет увеличивает скорость создания такой системы и существенно облегчает процесс её поддержки.

---

Работа представлена на III научную международную конференцию «Фундаментальные исследования», Доминиканская республика, 10-20 апреля 2008 г. Поступила в редакцию 20.03.2008.