УДК 621.762.4

Пачук В.И.

ПРОБЛЕМНО-ОРИЕНТИРОВАННЫЕ ЯЗЫКИ

БНТУ, Минск
Научный руководитель: Астапчик Н.И.

В настоящее время наиболее распространенным подходом к использованию абстракции является применение проблемно-ориентированных языковых (ПОЯ) средств специализированного характера. На границах спектра подобного инструментария, с одной стороны, можно расположить просто наборы сложных структур данных и соответствующих процедур обработки, построенных средствами одного из распространенных языков программирования. Другим предельным случаем может служить специально созданный под конкретный узко-очерченный класс задач новый язык. Такой язык не обладает никаким функционалом, выходящим за рамки решаемой проблемы, и покрывает только нужды данной предметной области. В данной статье, мы основное внимание уделим проблемно-ориентированным языкам (ПОЯ). Что же такое ПОЯ?

ПОЯ (DSL – Domain Specific Language) – это язык программирования, который обладает ограниченной степенью выразительности силы, сфокусированный на определенной предметной области и содержащий конструкты, отражающие абстракции именно этой предметной области.

ПОЯ принято разделять на три группы:

- внешние,
- внутренние,
- инструментальные средства языка.

Рассмотрим эти группы подробнее.

Внешние ПОЯ. Используют синтаксис, полностью отличный от синтаксиса языка основного приложения. Единственным исключением, пожалуй, является ситуация, когда в качестве основного языка берется XML.

Внешние ПОЯ – это наиболее яркие представители современных инструментов введения абстракции.

Самый большой плюс внешних ПОЯ состоит в том, что их можно писать, не оглядываясь на стандарты и спецификации. Иначе говоря, разработчик может выразить предметную область в самой лаконичной и пригодной для чтения форме. Качество такого, ПОЯ, определяется лишь умением разработчика создать транслятор, который сможет откомпилировать входной файл и выдать исполняемый. Отсюда же следует, что очевидным недостатком внешних ПОЯ – это необходимость создания этого самого транслятора.

Еще один существенный недостаток внешних ПОЯ определяется отсутствием у них того, что принято называть «символической интеграцией». Внешний ПОЯ, на самом деле, никак не связан с основным языком приложения. Программная среда разработки для базового языка, на котором пишется это приложение, не содержит информации о новом ПОЯ, и, следовательно, проблема редактирования связей исполняемого кода стоит довольно остро.

Внутренние ПОЯ. Используют существующий язык как свою основу. При этом создаваемый ПОЯ, как правило, получается синтаксически совместимым с языком-носителем и может быть обработан, используя его инфраструктуру. Полученный таким образом ПОЯ является одновременно как расширением языка-носителя, так и его ограничителем.

При построении внутреннего ПОЯ как расширения, нововведенные концепции становятся доступными для языка-носителя, и окончательным результатом является новый язык, который обладает полным функционалом исходного языка и добавляет специфичные расширения.

В случае реализации внутреннего ПОЯ как ограничителя языка-носителя, новый язык отражает специфику предметной области, при этом скрывая большинство конструкций

языка-носителя, которые не имеют отношения к задачам данной предметной области. Окончательный результат в таком случае – это также новый язык, только, в некотором смысле, суженный.

Инструментальные средства языка – это среды разработки (IDE – Integrated Development Environment), предназначенные для создания новых ПОЯ. Среда позволяет определять абстрактный синтаксис языка совместно с редакторами и генераторами языка. Редактор позволяют получить продвинутое и удобное окружение, которое учитывает специфику создаваемого ПОЯ по аналогии с IDE для языков общего назначения.

При использовании подобного инструментария разработчик, практически, не пишет код, а лишь манипулирует абстрактным представлением программы. Разумеется, IDE отображает эти изменения в тексте программы, но сути это не меняет – разработчик модифицирует не код, а абстрактное представление.

Проблемно-ориентированный язык остается довольно специфичным инструментом – он не вполне вписывается в объектно-ориентированную методологию разработки или в другие модели проектирования, которые представляют собой существенный сдвиг в понимании процесса разработки.

ЛИТЕРАТУРА

1. Грэхем, И. Объектно-ориентированные методы. Принципы и практика / И. Грэхем. – М.: Вильямс, 2004. – 380 с.

2. Гради, Б. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Б. Гради. – М.: Бином, 1998. – 345 с.

3. Синтес, А. Освой самостоятельно объектно-ориентированное программирование за 21 день / А. Синтес. – М.: Вильямс, 2002. – 372 с.