

*Бертран Мейер*

## РЕЧЬ НА ТОРЖЕСТВЕННОМ ВРУЧЕНИИ МАНТИИ И ДИПЛОМА ПОЧЕТНОГО ДОКТОРА СПБГУ ИТМО

Я очень благодарен за предоставленную возможность выступить перед студентами и преподавателями вашего университета. Во всем мире уважают талант русских программистов.

Я познакомлю вас с некоторыми направлениями моей работы в течение последних лет. Некоторые результаты вы можете найти в моей книге и в статьях.

Общая тема – Software engineering, которую я здесь определяю как совокупность процессов, методов, технических приемов, инструментария и языков для разработки качественного программного продукта.

Самое важное слово в этом определении, это, конечно, слово «качественный». Без учета темы качества нельзя говорить о Software engineering, потому что системы, которые мы сейчас развиваем, все более сложные. Цель Software engineering – создание систем высокого качества, которые в дальнейшем можно совершенствовать.

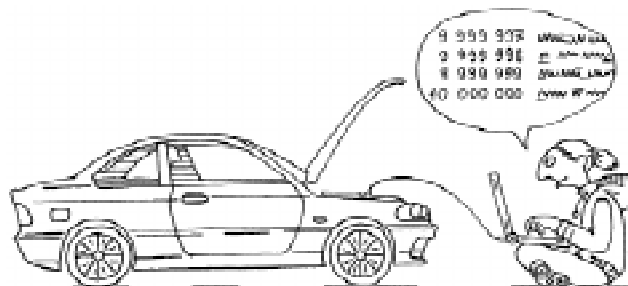
Примерами таких систем являются операционные системы, например, Windows. Windows XP содержит 45 миллионов строк кода.

В современных автомобилях фирмы BMW использованы десятки процессоров, которые содержат в общей сложности около 10 миллионов строк кода. Через несколько лет в машинах BMW будут использоваться программы в 100 миллионов строк. Человеческому уму очень трудно это представить.

Как мы развиваем системы, чтобы сделать их более надежными? Ответ на этот вопрос состоит из нескольких направлений, и в докладе я хотел бы коснуться одного из таких направлений – *системы контрактов*.

Каждый элемент программы создается для достижения определенной цели. Мы заставляем себя четко сформулировать цель каждого элемента программы. Здесь мы воспользуемся системой контрактов. Мы составим контракт не как документацию, а как часть программы. Контракт каждого элемента программы должен быть частью этого элемента.

В названии «контракт» прослеживается связь с понятием контракта между людьми и между организациями, когда оговариваются права и обязанности сторон. То, что для вас обязательство, то для вашего партнера право, и наоборот. Если одна сторона не выполнила условий контракта, то другая



*В современных автомобилях фирмы BMW использованы десятки процессоров, которые содержат в общей сложности около 10 млн строк кода.*

сторона имеет право ничего не делать: не требуется никаких действий, если клиент не выполнил своей части контракта.

Это пример действия контракта в повседневной жизни. Для элементов компьютерной программы контракт также связывает двух участников действия, определяя их права и обязанности.

Создается величина, обозначающая баланс (сальдо). Необходимое условие контракта – чтобы сальдо не было ниже заранее заданного минимального значения. Это часть семантики понятий.

Чтобы выразить свойства сальдо, мы напишем предусловие, которое показывает, что сумма должна быть положительной и не превосходить наперед заданной величины. Затем будет постусловие, которое показывает, что после операции новое сальдо должно быть равно разности старого сальдо и снятой суммы. Выражаем основные свойства этой операции через свойства высокого уровня, которые не зависят от реализации. Возможны другие способы реализации, например, с наследованием, но свойства остаются теми же самыми.

Есть понятие инварианта и фундаментальные характеристики класса, которые не изменяются при операции. Предусловие и постусловие связаны с одной операцией. То, что сальдо не будет меньше минимальной величины, это фундаментальное семантическое свойство, фундаментальная характеристика класса, или характеристика всех

объектов данного класса. Без этого трудно писать надежные программы.

Еще одно важное понятие – список. Можно выбирать элемент списка, например, наведением курсора. Позиция курсора выражается числом от 0 до 1. Это очень легко описать неформально, но гораздо лучше, если такая характеристика является частью текста программы.

Первое применение системы контрактов – это методика создания программы. В программировании приходится отлаживать программы. Гораздо лучше построить программу так, чтобы с самого начала она корректно работала.

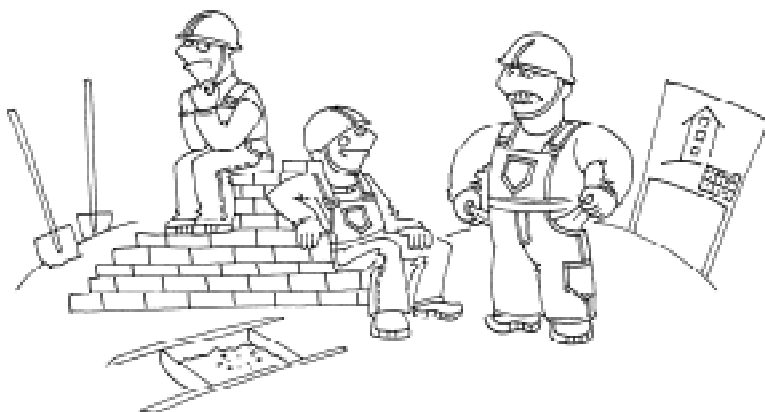
Второе применение – это документация. Здесь идет речь о документации не для пользователя, а для программиста.

Документация – это наиболее неприятная часть программирования. Большинство из нас программисты, и мы любим писать программу или устно о ней говорить, но не любим составлять комментарии. Писать текст комментариев – это трудно и скучно. Кроме того, приходится менять документацию в процессе отладки программы.

Часто, когда пишешь программу, поджимают сроки, и клиенты ждут, чтобы вы побыстрее программу закончили, при этом надо успеть написать комментарии. Совершенного решения этой проблемы нет, но писать документацию проще, когда она является не отдельным объектом, а частью программы, пишется в то же самое время и

в том же самом месте. Хорошая документация должна быть более абстрактной, чем то, что документируется. Если документация составлена на таком же самом уровне абстракции, как программа, то пользоваться такой документацией будет сложно.

Документация извлекается из самого текста программы. Такая документация очень проста в составлении и хорошо соответствует программе, потому что если ме-



*Если одна сторона не выполнила условий контракта, то другая сторона имеет право ничего не делать ...*

нешь программу, то автоматически меняется документация.

Третье применение контрактов. Язык программирования – не только способ коммуникации с машиной, но и способ коммуникации с другими людьми. Если говорить о способах создавать системы, менять системы, то современные языки программирования – эффективные способы выражения.

При этом система не обязательно является компьютерной программой. Это может быть просто инженерная система. И такими терминами, как классы, генерирование, контракты можно точно описать характеристики системы способами выражения очень высокого уровня.

Трудность в том, что описание должно одновременно быть и абстрактным, и точным. Языки программирования точно описывают алгоритм, но они не абстрактные. А здесь, с помощью контрактов, можно описывать систему абстрактно и точно.

Последнее, четвертое, применение контрактов, о котором я буду говорить – это тестирование. Немногие любят делать тестирование и отладку. Об этом мало говорят



*Язык программирования —  
не только способ  
коммуникации с машиной,  
но и способ коммуникации  
с другими людьми...*

в наших университетах. Обычно рассказывают о более возвышенных аспектах программирования. Но на практике, когда мы пишем программу, она не всегда сразу работает так, как мы хотим. А с помощью контрактов можно гораздо более систематично заниматься этой работой.

*Бертран Мейер,  
заведующий кафедрой Software  
Engineering декан факультета  
Computer Science ETH (Высшей  
Политехнической школы в Цюрихе).*



Наши авторы, 2006.  
Our authors, 2006.