

# Лицо программы

Андрей Ковалев

Этот сюжет навеян переводом статьи Билла Дэя о [программировании устройств на Java](#). Там устройства подразделяются на бытовые и встроенные. Возможно, имеет право на существование перевод слова "consumer" на русский как "потребительский", что точнее отражает способность устройства приносить пользу, а также говорит о наличии у него пользовательского интерфейса. Тем не менее отметим, что любое устройство в конечном итоге так или иначе потребляется, так что отличие между двумя типами устройств обозначим введением двух уровней — бытового и встроенного. Таким образом, получится, что устройства первого типа являются, так сказать устройствами наружного потребления и обладают пользовательским интерфейсом, а устройства второго уровня предназначены исключительно для внутреннего потребления (конечно, другими устройствами 8-) и в них имеют место только электронные и механические интерфейсы.

К какому уровню отнести привычные программистам пользовательские интерфейсы, созерцаемые ими на экране персонального компьютера? Большинство из того, что изображается на экране является моделью происходящего в реальной жизни, а компьютеры, как известно, разрабатываются для моделирования. Поэтому мы назовем этот уровень модельным. Это определение подтверждает так же тот факт, что все пользовательские интерфейсы устройств в первом приближении создаются, моделируются на компьютере. Элементы управления на модельном уровне неосвязаемы, опосредованы, как и сама рабочая среда, являющаяся виртуальной.

Поэтому на рис. 1 этот уровень покрывает иерархию сверху. Снизу мы расположим уровень встроенных устройств, которые по функциональному признаку являются вспомогательными и далеки от темы настоящей колонки, так как пользовательским интерфейсом не обладают вовсе или поддерживают минимальные возможности такового. Посередине располагается слой *целевых* интерфейсов многих устройств, которыми люди привыкли пользоваться *практически* в повседневной жизни.

модельный
бытовой
встроенный

**Таблица 1. Иерархия пользовательских интерфейсов.**

Разработка бытовых интерфейсов не может быть универсально автоматизирована. Всегда в прикладной отрасли создаются специальные средства для их моделирования, конструирования и разработки технологической документации. Этого требует конкуренция, этого требует маркетинг. Более того, сами устройства, управляемые этими интерфейсами, требовательны к расходу ограниченных вычислительных ресурсов. В результате библиотеки Java переформируются в профили, из которых вычищается все лишнее, что накоплено на жестких дисках тысячами программистов.

Программные оболочки для разработки приложений и интерфейсов не могут учесть специфики многочисленных целевых устройств и задач. Даже языки моделирования, такие как UML, способны грамотно представить задачу только на уровне алгоритмов и структур классов. Поэтому мы приходим здесь к выводу, что автоматическое генерирование пользовательского интерфейса в среде разработки предназначено только для освоения техники создания интерфейсов и может помочь только на этапе обучения. Необходимо помнить об эстетических потребностях пользователя и понимать, что автоматизация порождает "бездушное" лицо, как у людей без глаз в рекламе телевизоров LG. Процесс разработки с точки зрения программиста заключается в наделении этого автомата чувствами, способностью точно реагировать на запросы пользователя. Именно тогда рождается лицо программы (или устройства), которое будет приятно на вид и на ощупь.