

Философия разработки пользовательских интерфейсов

Андрей Ковалев

Во вводной статье мы отметили, что разработка интерфейса пользователя в Java существенно упрощена. Рассмотрим место Java в гамме языков программирования и, в частности, место библиотеки Swing. Среди наиболее распространенных языков мы можем назвать в порядке возрастания затрат времени на создание приложения

Visual Basic	Java	Pascal	C++	Assembler
--------------	------	--------	-----	-----------

Объединяя языки в категории по принципу функциональности отдельной команды, получим следующую картину

Макроязыки и Скрипты	Языки высокого уровня	Языки низкого уровня
----------------------------	-----------------------------	----------------------------

Получаем, что в этой иерархии Java занимает левое крыло языков высокого уровня, то есть по простоте организации пользовательского интерфейса и по способу генерации объектного кода приближается к описательным макроязыкам или скриптам, а благодаря наличию эффективного синтаксиса программирования относится к языкам высокого уровня.

Как известно, развитие языков двигалось справа налево, от машинных кодов к макроязыкам. Поэтому Java стоит на высшей ступени развития. Такой уровень достигнут благодаря соблюдению 5 принципов:

1. **иерархии компонентов;**
этот принцип является стандартом со времен появления концепции объектного ориентирования;
2. **управления размещением;**
эта уникальная особенность обусловила простоту создания пользовательского

интерфейса на Java, сократив до минимума необходимость программного слежения за поведением компонентов в контейнерах.

3. обработка событий;

этот естественный для современных многозадачных систем принцип нашел оригинальное развитие в Java, где происходит реальное структурирование событий по типам.

4. рисование;

эта компонента пользовательского интерфейса выделена в отдельный класс с полным набором возможностей и фактически является отдельной библиотекой.

5. разделение данных и алгоритмов их обработки;

это свойство, добавленное в Java 2 предоставляет программисту универсальный способ работы с информацией, многократное использование алгоритмов и создание естественных структур данных в результате последовательных поисков.

Я не рассматриваю здесь многочисленные особенности Java, а выделил лишь интересующие нас в плане обозначенной тематики колонки.

С точки зрения последовательности разработки наши принципы располагаются следующим образом: разработка компонентов — размещение компонентов — написание реакций на события компонентов — генерация или ввод данных, их обработка. Проще говоря, цикл разработки включает в себя как создание пользовательского интерфейса, так и написание бизнес логики и Java позволяет сосредоточиться на этих двух задачах. Подчеркнем, что объектная концепция позволяет разделить эти две задачи между сотрудниками.

Таким образом мы будем вести обсуждение в соответствии с намеченными направлениями, отмечая как философские вопросы, так и практические подходы к их решению.