

УДК 658.512.011.56: 681.3.06

А.Е.Лосев (3 курс, каф. ИУС), Я.Н.Коршиков (5 курс, каф. ИУС), В.П.Котляров, к.т.н., проф.

## РАЗРАБОТКА БАЙТ-КОДА АНАЛИЗАТОРА ДЛЯ ЦЕЛЕЙ ТЕСТИРОВАНИЯ JAVA ПРИЛОЖЕНИЙ

Модульное тестирование является одним из неотъемлемых этапов разработки качественного продукта. Оно обеспечивает обнаружение ошибок на ранних этапах разработки и позволяет обнаружить ошибочный модуль.

Модульное тестирование в J2ME-приложениях обеспечивается путём создания нового тест-приложения (MIDlet), которое включает функциональность тестируемого приложения. При этом необходимо произвести модификации тестируемого кода: изменить атрибуты доступа к классам, методам и полям для того чтобы получить к ним свободный доступ и возможность задать тестовые входные параметры и проверить работу приложения при этих параметрах. Также необходимо изменить иерархию наследования классов исходного продукта, т.к. тестируемое приложение запускается внутри тестовой оболочки.

В языке JAVA законченный программный продукт представляет из себя jar-файл, в который упакован скомпилированный код приложения – class-файлы. Существуют декомпиляторы, которые позволяют восстановить исходный код приложения. Над полученным исходным кодом можно вручную произвести все необходимые модификации, но, к сожалению, они не всегда корректно декомпилируются, и полученный в результате декомпиляции код не компилируется.

Разработанная нами программа BCAnalyzer позволяет эффективно решать поставленные задачи. Она анализирует байт-код исходного class-файла без предварительной декомпиляции. Это означает, что исходный код программы не нужен, а необходим только jar-файл, т.е. BCAnalyzer работает с конечным продуктом. Наша программа распаковывает jar-файл приложения и модифицирует class-файлы: находит нужные байты, отвечающие за атрибуты доступа классов, методов и полей, изменяет эти атрибуты на необходимые нам. Так же BCAnalyzer позволяет изменить иерархию наследования, добавляя в неё тестовые модули. BCAnalyzer не изменяет функциональность тестируемого приложения, делает все необходимые модификации автоматически без вмешательства человека и без процесса повторной компиляции.

В результате работы мы получаем новые class-файлы, отличающиеся от исходных только байтами доступа и иерархией наследования. Эти файлы легко встраиваются в тестовое окружение и обеспечивают возможность протестировать любой модуль исходного продукта.

Также разработанная нами программа пригодна и для целей системного тестирования, т.к. позволяет задать тестовые значения переменных тестируемого приложения.

В процессе разработки использовалась среда MS Visual Studio 6.0, SUN JDK 1.4.2 и DJ decompiler.