

УДК 681.3.06

В.Е.Поддубный (3 курс, каф. ИУС), Ю.В.Юсупов (4 курс, каф. ИУС),  
А.В.Гаригин (6 курс, каф. ИУС), В.П.Котляров, к.т.н., проф.

## СИСТЕМА ДЛЯ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ СТУДЕНЧЕСКИХ ПРОГРАММНЫХ ПРОЕКТОВ ПРЕПОДАВАТЕЛЯМИ КАФЕДРЫ

Хорошо известно, что от молодых специалистов в современных условиях требуются как глубокие теоретические познания, так и хорошие практические навыки. На «компьютерных» специальностях под «хорошими практическими навыками» понимается, в первую очередь, умение в сжатые сроки создать качественный программный продукт, удовлетворяющий современным требованиям. Чтобы получить такие навыки, необходима практика, в рамках которой студенты должны выполнить множество лабораторных работ, результатами которых являются программы и все сопутствующие спецификации. Данный подход требует значительных затрат времени преподавателей на организацию проверки таких работ, так как требуется провести всестороннее тестирование каждой работы на соответствие техническому заданию.

В настоящее время всё большее распространение получают системы автоматизированного и автоматического тестирования программных продуктов. К такого рода системам относится, в том числе, система ТАТ, работающая с MSC-диаграммами как источником данных для тестирования. Однако непосредственное применение систем автоматизированного тестирования для проверки студенческих работ зачастую затруднено тем, что они не предоставляют возможности удаленного использования для проверки в автоматическом режиме большого количества *однотипных* программ, загружаемых студентами по сети.

Существуют также специализированные программные продукты для проверки студенческих работ, имеющие web-интерфейс. Примером свободно распространяемого программного продукта такого типа может служить система Microsoft Academic Tools (MSAT). Подобные системы зачастую прекрасно справляются с необходимостью обеспечения безопасности, конфиденциальности, автоматического запуска одинаковых тестов для работ на одну тему. Но к сожалению, имеющиеся в их распоряжении средства тестирования, как правило, сводятся к проверке сгенерированного программой вывода на соответствие образцу. При этом *малейшая* ошибка (например, орфографическая) приведёт к выставлению неудовлетворительной оценки, а значит, преподавателю придётся каждую программу перепроверять вручную.

Целью данной работы является интеграция MSAT и ТАТ. При этом возникают следующие основные трудности:

- система MSAT не рассчитана на запуск программ, отличных от присланных студентами (то есть нельзя напрямую запустить ТАТ);
- ТАТ требует передавать в качестве параметров данные, различающиеся для различных программ (например, путь);
- необходимо вызывать ТАТ таким образом, чтобы студенты не могли определить местоположение файлов, используемых при тестировании (чтобы программу нельзя было «подогнать» под ответ);
- преподаватель должен иметь возможность легко добавлять или удалять тесты по мере изменения требований к программе.

Кроме перечисленного, реализация проекта не должна принципиально зависеть от того, на каком языке написана студенческая программа. Для этой цели в качестве платформы для реализации выбрана Microsoft .NET. Это позволяет использовать её средства получения данных о программах и связывания их с необходимыми библиотеками только при необходимости во время выполнения. В том числе, имеется возможность вызова конкретных методов и функций, чтения значений полей класса без запуска всей программы. Кроме удобств при тестировании, подобная функциональность позволяет запустить ТАТ из студенческой программы, не влияя на функции программы в обычном (не тестовом) режиме и не информируя студента о конкретных используемых файлах.

Реализация данного проекта позволит в значительной степени сократить затраты преподавателя на проверку студенческих работ за счёт отсеивания заведомо неверных программ. Более того, проверка работ на отметку «зачёт/незачёт» сможет проводиться в полностью автоматическом режиме.

Само тестирование студенческих проектов будет осуществляться по составленным спецификациям на языке MSC. Это позволит провести значительно более глубокую проверку *важных* для конкретной задачи частей программы (например: корректность работы с базой данных), не заостряя внимание на особенностях конкретной реализации *менее значительных* частей (например: язык интерфейса, язык программирования).