

тестирования: статический, детерминированный, стохастический и в реальном масштабе времени.

Статическое тестирование является наиболее формализованным и автоматизируемым методом проверки корректности программ. В качестве эталона применяются правила структурного построения программных модулей и обработки данных; конкретизированные для проекта в целом. Проверка степени выполнения этих правил проводится без исполнения объектного кода программы путем формального анализа текста программы на языке программирования.

При детерминированном тестировании контролируется каждая комбинация исходных эталонных данных и соответствующая ей комбинация результатов функционирования программы. Это позволяет выявлять отклонение результатов от эталона с конкретным фиксированием все значений исходных и результирующих данных, при которых это отклонение обнаружено.

В сложных программах невозможно перебрать все комбинации исходных данных и проконтролировать результаты функционирования программы на каждой из них. В таких случаях применяется стохастическое тестирование, при котором исходные данные задаются множествами случайных величин с соответствующими распределениями и для сравнения полученных результатов используются распределения случайных величин.

В процессе тестирования в реальном масштабе времени проверяется исполнение программ, и обработка исходных данных с учетом времени их поступления, длительности и приоритетности обработки, динамики использования памяти и взаимодействия с другими программами и т.д.

#### Литература

1. Липаев В.В. Тестирование программ. — М.: Радио и связь, 1986.
2. Майерс Г. Надежность программного обеспечения. — М.: Мир, 1980.
3. Бахгизин В. В., Иблуду К. А., Савкин В. В. Методы тестирования и верификации программ. — М.: Машиностроение, 1984.

#### СИСТЕМА КОНТРОЛЯ ЗНАНИЙ

Тиунчик Д.В., Лобов С.Д.

Брестский государственный университет им. А.С.Пушкина

Брест, Беларусь

**Аннотация:** в работе рассмотрено построение пакета контролирующих программ в среде Windows 9x. Структура и особенности пакета, его возможно-

сти: создание тестов, проведение тестирования и анализ результатов тестирования.

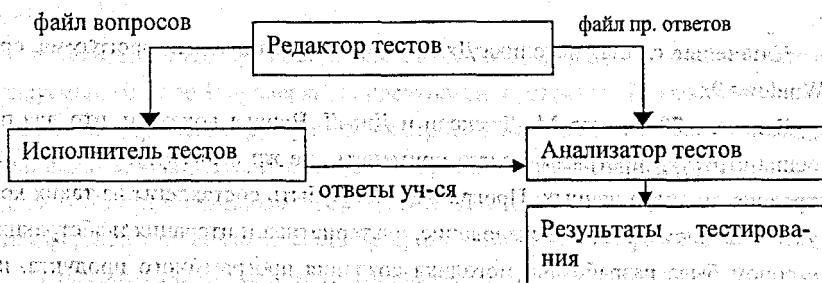
**Ключевые слова:** методика Джексона, контролирующая программа, среда Windows 9x.

В начале 70-х годов М. Джексон и Дж.-Д. Варнье доказали, что для построения структур программ можно применять те же конструкции, что и для построения структур данных. Программы могут быть составлены из таких конструктивных элементов, как следование, альтернатива и итерация и абстракция. Джексоном была разработана методика создания программного продукта, названная в последствии его именем. Эта методика основывается исключительно на процедурном подходе к программированию. Она позволяет максимально автоматизировать процесс программирования и свести к минимуму количество ошибок возникающих в процессе создания комплексов (пакетов) программного обеспечения.

Мы на практике применили методику Джексона при написании пакета тестирующих программ в одной из самых распространенных операционных средах – Windows 9x. При написании использовалась среда визуального программирования – Delphi.

Целью работы было создание оболочки для осуществления текущего и итогового контроля знаний тестируемого по самым различным учебным дисциплинам, а так же компьютерная реализация тестов из различных областей исследования (психология, социология, определение IQ и т.п.). Мы исходили из того, что оболочка должна отвечать следующим требованиям:

- удобный многооконный интерфейс;
- простота в использовании;
- доступная справочная система;
- поддержка различных типов вопросов, как с рисунками, так и без них;
- отсутствие доступа у тестируемого к ответам;
- наличие временных ограничений;
- автономные режимы создания тестов и анализа результатов тестирования.



Структура программы: пакет состоит из трех компонент: редактор тестов, исполнитель тестов и анализатор тестов.

Редактор тестов представляет собой многооконную среду для создания и редактирования тестов. Отметим основные особенности редактора тестов. С помощью редактора тестов создаются два файла: файл вопросов и файл правильных ответов. Файл вопросов имеет формат Paradox. В этом файле хранятся формулировки вопросов, рисунки к вопросам (если в них есть необходимость) и, если это необходимо, варианты ответов для выбора тестируемым. Файл правильных ответов редактор тестов создает автоматически. Файл правильных ответов — это также файл формата Paradox, в котором хранятся правильные ответы на вопросы теста из соответствующего файла вопросов. Такой способ подготовки тестов позволяет принять дополнительные меры по защите правильных ответов к тесту от несанкционированного доступа. Причем имеется возможность защиты данных паролем.

Исполнитель тестов служит для выполнения тестирования учащихся. При работе использует файл вопросов, созданный редактором тестов. Программа позволяет устанавливать различные параметры тестирования. Так с легкостью можно ограничить время тестирования, задать пароль для прерывания тестирования. Отметим также, что в режиме тестирования отключается ряд комбинаций "горячих" клавиш, в том числе и комбинация клавиш Ctrl+Alt+Del с целью предотвращения принудительного завершения работы исполнителя тестов. Программа позволяет проводить потоковое тестирование, в этом случае достаточно один раз указать файл вопросов и параметры тестирования в начальном диалоге и после этого «менять» тестируемых. По результатам тестирования программа автоматически создает файлы с ответами тестируемых. Эти файлы используют-

ся в последствии для анализа результатов тестирования с помощью анализатора тестов.

Анализатор тестов представляет собой программы анализа результатов тестирования. Требуется наличия файла правильных ответов и указания папки с ответами учащихся. Автоматически выполняет поиск файлов ответов учащихся и сравнение их с правильными ответами. В случае полного совпадения ответов выставляется «+», несовпадения ответов – «-». Если же ученик не смог ответить на поставленный перед ним вопрос, то выставляется знак «н/о» – «нет ответа», а в случае неполного, или избыточного ответа – «ч/о» – «частичный ответ». Все данные представлены в виде таблицы, в которой отображены индивидуальные данные о тестируемом, результаты ответа на каждый вопрос, а также общие результаты: количество правильных, неправильных и частичных ответов, а также количество вопросов, на которые ответа не последовало. Собирается также информация о суммарных результатах тестируемых, которая позволит проанализировать общие результаты тестирования всей группы сразу. Это позволит тестирующему проанализировать результаты, выяснив, например, степень усвоения материала по той или иной теме. Имеется возможность распечатки отчета о тестировании на любом принтере, установленном в системе.

Контекстно-зависимая справочная система приходит на помощь в любой ситуации. Поэтому любой неподготовленный пользователь программы успешно справляется с возникающими в процессе эксплуатации трудностями.

Далее поговорим о поддерживаемых параметрах вопросов. Как было отмечено выше, редактор тестов позволяет создавать, вопросы различных типов. Это, во-первых, вопросы альтернативного выбора, т.е. такие вопросы, при ответе на которые ученик выбирает правильный ответ или правильные ответы из нескольких предложенных ему альтернатив (от 2 до 5 вариантов ответа на вопрос). Во-вторых, редактор тестов с легкостью создает, такие вопросы, правильный ответ на которые необходимо вводить с клавиатуры. Для этого достаточно задать только один вариант ответа в редакторе тестов. В результате этого при выполнении теста будет отображена строка ввода, где и производится ввод правильного ответа с клавиатуры.

Литература

1. Кинг Д. Создание эффективного программного обеспечения: пер. с англ. - М.: Мир, 1991. - 228 с.

2. Тиунчик Д.В., Лобов С.Д. "Использование средств языка программирования DELPHI для написания приложений ведения баз данных" / Материалы Брестской региональной научно-практической конференции молодых ученых "Математика и ЭВМ-98", Брест, 1998. — С.44.

## **КОНСТРУКТИВНАЯ КОМПОЗИЦИЯ СТРУКТУР СЕТЕВЫХ МОДЕЛЕЙ ОРГАНИЗАЦИОННО-ТЕХНОЛОГИЧЕСКИХ СИСТЕМ**

Бортник Е.М., Ревотюк М.П.

Белорусский государственный университет информатики и радиоэлектроники  
Минск, ул. П.Бровки, 6, кафедра ИТАС

Рассматривается задача формального конструирования исполнительных версий сетевых интерпретируемых моделей в системах управления реального времени. Предлагаемый реляционный подход к структуризации моделей дискретных процессов оценивается как средство сокращения трудоемкости формализации и конструктивная основа объектно-ориентированного проектирования систем управления организационно-технологического уровня.

**Ключевые слова:** имитационное моделирование, сети Петри, реляционные операции.

Известно, что наиболее привлекательным видом моделирования дискретных процессов с регулярной структурой являются сети Петри и их расширения [1]. Ориентация сетей Петри на отражение свойства восприимчивости реальных систем к локальным изменениям переменных состояния весьма удобна как при формализации дискретных процессов со сложными асинхронными взаимодействиями, так и реализации технологий объектно-ориентированного проектирования и программирования [2]. Цель исследования — разработка формального аппарата структурных преобразований имитационных моделей на основе расширенных сетей Петри применительно к задачам управления системами организационно-технологического уровня.

Сущность задачи конструирования структурированных сетевых описаний поясним на примере моделирования робототехнической системы (РТС) с линейным размещением оборудования. Пусть моделируемая РТС, предназначенная для обработки транспортных партий деталей (ТПД), включает комплект обрабатывающего и накопительного оборудования и транспортный робот (ТР).

Предварительный анализ позволяет выделить в процессе функционирования РТС три качественно различные составляющие: прохождение ТПД по тех-