УДК 004.42; 004.438

Новиков М.Д.,
кандидат физико-математических наук,
старший научный сотрудник
лаборатории вычислительного практикума и
информационных систем факультета Вычислительной
математики и кибернетики МГУ им. М.В. Ломоносова

СИСТЕМЫ АВТОМАТИЧЕСКОГО ТЕСТИРОВАНИЯ ПРОГРАММ

Аннотация. *В статье описываются три системы, предназначенные для тестирования программ, написанных на языках Паскаль, Ассемблер и Си, соответственно. Даются характеристики каждой из трёх систем, их общие возможности и индивидуальные особенности. Приводится пример работы систем.*

Abstract. *Three systems destined for automatic testing of programs made in Pascal, Assembly and C programming languages are described in the article. Some properties and peculiarities of each of the systems are given. An example is presented.*

Ключевые слова: *программирование, тестирующие системы, обучение программированию, язык Паскаль, язык Ассемблера, язык Си.*

Key words*: programming, testing systems, teaching programming, Pascal programming language, Assembly programming language, C programming language.*

Введение

Системы автоматического тестирования программ разрабатываются на факультете Вычислительной математики и кибернетики МГУ им. М.В.

Ломоносова с 2017 года. Первой была создана система тестирования программ на Паскале, затем была создана система тестирования программ на Ассемблере. Разработка этих двух систем, в основном, завершена. Система же тестирования программ на Си начала создаваться в 2023 году и находится в стадии развития.

Все три системы автоматического тестирования (далее – Паскаль-система, Ассемблер-система и Си-система) предназначены для тестирования программ, выполняемых студентами младших курсов факультета ВМК в рамках практикума на ЭВМ. Необходимость создания подобных систем обусловлена трудностями проверки правильности программ, написанных студентами. Для этого требуется подбирать индивидуальные наборы исходных данных для каждой задачи и вручную вводить их с клавиатуры, что, учитывая сложность программ и большой набор исходных данных, занимает много времени, а также не гарантирует отсутствие ошибок при вводе и полноту тестирования. Системы же позволяют избежать этого трудоемкого процесса, автоматически тестируя программы на заранее подготовленных наборах исходных данных.

Общие возможности трёх систем

1) Каждая из трёх систем позволяет выбрать задачу из предлагаемого списка, задать имя файла с тестируемой программой и выполнить программу на наборах исходных данных, индивидуально подобранных для каждой задачи, встроенных в систему и обеспечивающих полную проверку правильности работы программы. Задачи для систем взяты из книг [1], [2], [3] и [4]. При этом Паскаль-система включает в себя задачи из [1] и [4], Ассемблер-система - задачи из [2] и [4], а Си-система – задачи из [1] и [3].

2) Все три системы позволяют тестировать как полные программы, так и их фрагменты, в частности, процедуры и функции. В последнем случае к тестируемой программе добавляется основной блок, дополняющий её до

полной программы. Для некоторых задач также требуется описать внутри программы процедуру или функцию с заданными свойствами (см., например, задачи серии 11.22. из [1]). В этом случае предусмотрено выделение из полной программы требуемой подпрограммы и отдельное её тестирование.

3) Системы проверяют выполнение требований к программам, указанным в [1]-[4]. Это, в основном, запрет, ограничение или, наоборот, обязательное использование каких-либо конструкций языка программирования – циклов, массивов, динамических структур и проч. Такие требования обусловлены тем, что студент должен не просто написать программу для решения задачи, но и продемонстрировать своё знание определённых приёмов программирования, конструкций языка, а также избежать неоптимальных – избыточных по времени выполнения и/или по объёму используемой памяти алгоритмов решения задач: например, использование вложенных циклов там, где можно обойтись одним циклом, использование массива там, где можно обойтись фиксированным числом переменных и т.п. Такая проверка производится путём анализа тестируемой программы – подсчётом количество вхождений в неё определённых слов – if, do, repeat, array и проч.

4) Для тестирования статических массивов (границы которых заданы именованными константами), создаются 4 копии тестируемой программы с разными значениями констант. Тем самым программа тестируется для массивов различных фиксированных длин, что позволяет выявить ошибки, связанные с использованием в программе числа вместо именованной константы (в основном, это случается при задании границ цикла).

5) Входные и выходные данные для программ отображаются на экране в удобном для пользователя виде. Матрицы выводятся по строкам – каждая строка матрицы на одной строке экрана. В случае ввода/вывода множества одинаковых объектов или разнородных объектов (см., например, задачу 11.61 из [1], где требуется ввести три вещественные матрицы и степень, в которую их надо возвести) все объекты ввода разделены и ввод снабжён

соответствующими комментариями. Деревья выводятся в виде линейной структуры: например, дерево поиска из четырёх вершин со значениями 5 (левый узел), 10 (корень), 20 (правый узел) и 15 (левый узел правого поддерева) выводится в виде T 10, TL 5, TR 20, TRL 15. Такой вывод оказался более удобным для восприятия, чем традиционный графический вывод.

Дополнительные проверки, выполняемые системами

Кроме описанных выше возможностей систем, каждая их них производит дополнительные проверки, связанные со спецификой конкретного языка программирования.

В Паскаль-системе дополнительно проверяется следующее [5], [6].

- Досрочный выход из цикла (break) после того, как становится известен результат вычисления.
- Использование ограниченных типов данных ([1], глава 7).
- Наличие рекурсии в задачах на рекурсивные алгоритмы.
- Корректность работы с файлами – закрытие всех файлов и удаление всех временных файлов по окончании выполнения тестируемой программы.
- Корректность работы с динамическими структурами – отсутствие висячих ссылок и утечек динамической памяти.
- Вид формальных параметров процедур и функций (value- или var-параметры).
- Производятся также другие необходимые проверки в соответствии с требованиями, предъявляемыми к программам.

Кроме упомянутых выше проверок, в системе производится специальный анализ ответов, выдаваемых тестируемой программой для задач, допускающих несколько правильных решений. Это, например, задача нахождения корней биквадратного уравнения ([1], 4.12д) – корни могут быть выведены в любом порядке, задача удаления элемента из дерева поиска ([1],

17.17м), задача расстановки между числами знаков операций так, чтобы получившееся выражение было равно некоторому числу S ([1], 18.36), и другие задачи. Для каждой такой задачи разработан специальный блок анализа ответа на предмет его правильности.

В Ассемблер-системе, дополнительно проверяется следующее [7].

- Отсутствие выхода за границу используемой памяти.
- Выполнение стандартных соглашений о связях между процедурой и основной программой ([8], глава 6): передача параметров через стек, очистка стека от параметров по окончании выполнения подпрограммы, отсутствие внешних объектов и сохранение значений всех регистров процессора.

Что касается Си-системы, она находится в стадии разработки; помимо упомянутых выше общих возможностей в ней планируется реализовать проверку корректности работы с файлами и с динамической памятью.

Апробация систем

Все три системы написаны в среде Delphi, могут выполняться на локальном компьютере и не требуют Интернета. К настоящему времени составлены тесты примерно к шестистам задачам на Паскале, трёмстам задачам на Ассемблере и также к трёмстам задачам на Си.

Использование систем показало их эффективность. До их создания студенты часто приносили на проверку неотлаженные программы, содержащие большое число логических ошибок. Теперь же студенты отлаживают свои программы самостоятельно, принося на проверку уже готовые версии.

Результаты были доложены на конференции «Математические основы информатики и информационно-телекоммуникационных систем» [9], и на конференции «Ломоносовские чтения» в 2021, 2023 и 2024 г.г. [10], [11], [12].

Пример:

В качестве примера рассмотрим тестирование программы для задачи 16.23е из [1]: надо написать процедуру, удаляющую из одностороннего списка первый отрицательный элемент (если такой есть).

Копии экрана представлены на рис. 1 и 2. Первый вариант программы содержит ошибку: она заключается в том, что при удалении из списка первого элемента (в случае, когда он отрицательный), не выполняется процедура освобождения памяти Dispose. Второй вариант программы – правильный (добавлена процедура Dispose).

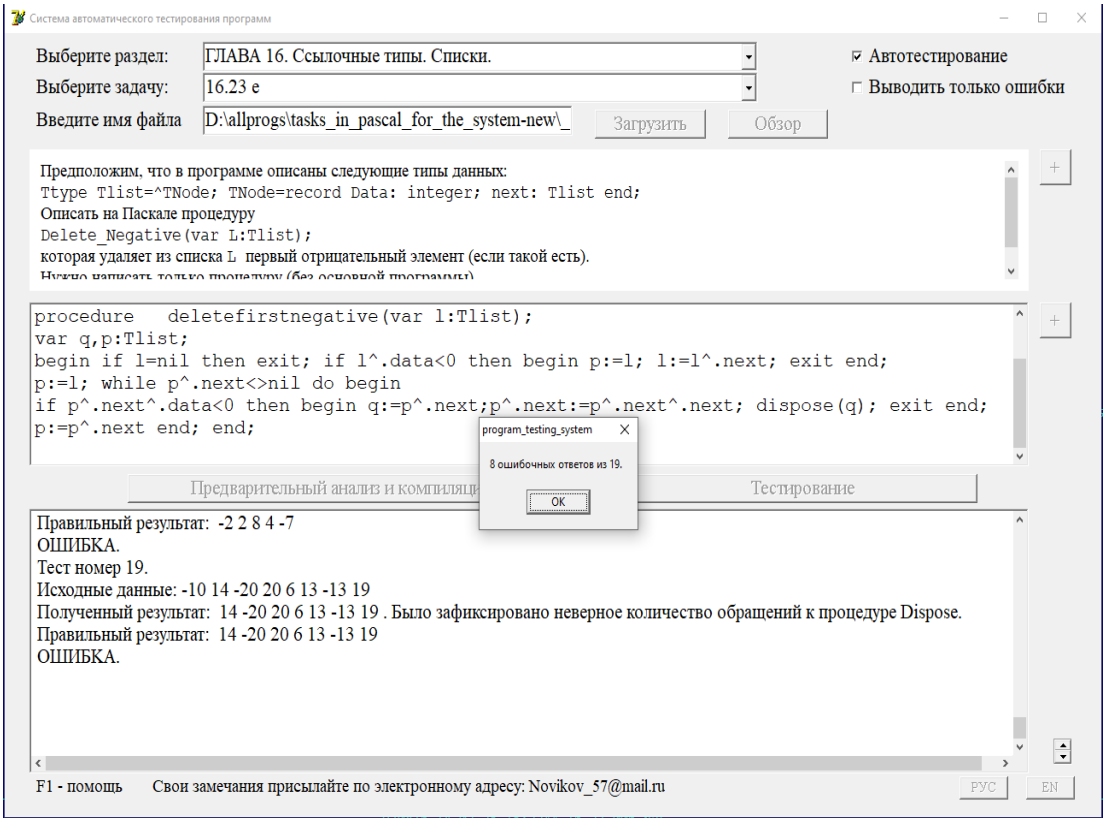

Рисунок 1. Результат тестирования программы, содержащей ошибку

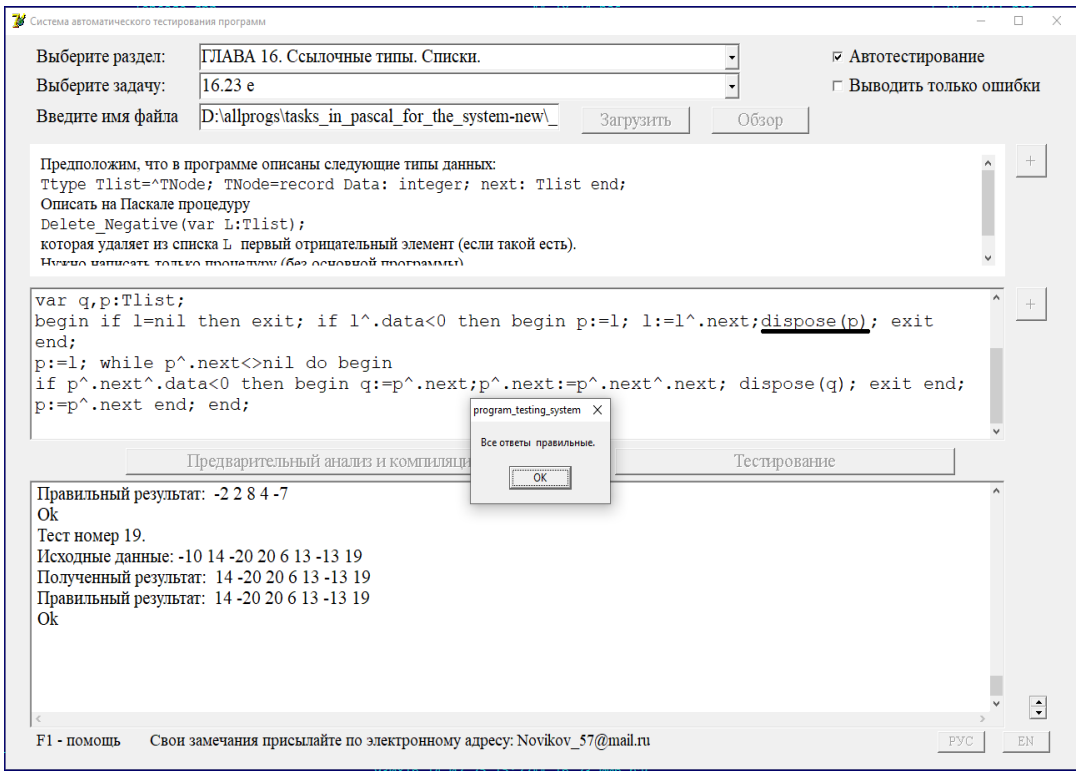

Рисунок 2. Результат тестирования исправленной программы

Использованные источники

1. Пильщиков В.Н. Язык Паскаль. Упражнения и задачи. М.: Научный мир, 2003, 224 с.
2. Бордаченкова Е.А. Задачи и упражнения по языку ассемблера MASM. Учебное пособие для студентов I курса. М.: Издательство Московского университета, 2023, 112 с.
3. Руденко Т.В. Сборник задач и упражнений по языку Си. Учебное пособие для студентов II курса. М.: Издательский отдел факультета ВМиК МГУ им. М.В. Ломоносова, 1999, 80 с.
4. Бордаченкова Е.А., Панфёров А.А. Задания практикума. 1 курс. Учебное пособие для студентов 1 курса. М.: Издательство «МАКС Пресс», 2016, 48 с.
5. Новиков М.Д. Автоматическое тестирование сложных программ на языке Паскаль. В журнале «Национальная ассоциация учёных».

Екатеринбург: Издательство «Евразийское научное содружество», т.1, №88, 2023, с. 56-59.

6. Новиков М.Д. Тестирование программ на языке Паскаль, использующих динамическую память. В журнале «Eastern European Scientific journal», т.1, №3(67), 2021, с. 45-47.

7. Автоматическое тестирование процедур на языке Ассемблера. В журнале «Международный журнал гуманитарных и естественных наук», №3-2 (66), 2022, с. 144-147.

8. Баула В.Г. Введение в архитектуру ЭВМ и системы программирования. Учебно-методическое пособие. – [электронный ресурс]. – URL: http://arch64.cs.msu.su (дата обращения – 14.04.2025).

9. Новиков М.Д. Автоматическое тестирование студенческих программ. В сборнике Всесоюзная научная конференция «Математические основы информатики и информационно-телекоммуникационных систем». Сборник трудов. Тверь.: издательство Тверской государственный университет, 2021, с. 235 – 240.

10. Новиков М.Д. Система автоматического тестирования программ на языке ассемблера. В сборнике Ломоносовские чтения-2021, научная конференция, факультет ВМК МГУ, тезисы докладов, серия «Вычислительная математика и кибернетика», издательство Московского университета (Москва), тезисы, с.116-117.

11. Новиков М.Д. Автоматизированный практикум для студентов первого курса. В сборнике Ломоносовские чтения-2023, научная конференция, факультет ВМК МГУ, тезисы докладов, серия «Вычислительная математика и кибернетика», издательство «МАКС Пресс» (Москва), тезисы, с.26-27.

12. Новиков М.Д. Программное обеспечение поддержки практикума на младших курсах факультета ВМК. В сборнике Ломоносовские чтения-2024, научная конференция, факультет ВМК МГУ, тезисы докладов, серия «Вычислительная математика и кибернетика», издательство «МАКС Пресс» (Москва), тезисы, с.186-187.