

3-е изд., стереотип. – М.: Дрофа, 2002.

Н. М. Субоч, А. В. Карпов, Е. А. Святушенко, М. А. Ройтберг  
Пушино, Институт математических проблем биологии РАН, Пушкинский  
государственный университет, Гимназия Пушкино

Проект: Кумир <http://www.infomir.ru>, <http://lpm.org.ru/kumir>

## Методы тестирования в разработке системы обучения программированию КуМир

### Аннотация

В докладе обсуждается методика тестирования, использовавшаяся при разработке системы программирования КуМир. Описана структура корпуса тестов, методика создания этого корпуса и средства поддержки тестирования. Используемая методика может быть полезна при разработке других обучающих систем.

### Введение

В ходе разработки системы КуМир (как и при разработке всех достаточно сложных систем) мы столкнулись с двумя проблемами. Во-первых, хотя для системы было разработано достаточно подробное описание, не все функциональные возможности оказывались реализованными. Во-вторых, при развитии системы могут перестать правильно работать уже отлаженные компоненты. Чтобы преодолеть эти трудности, нами был создан корпус тестов и разработан протокол тестирования, включающий как автоматизированную, так и интерактивную составляющие. Идея примененного подхода — в подготовке максимально подробного описания системы и создании корпуса тестов в соответствии с этим описанием. Такой подход не гарантирует отсутствие ошибок, но, как показал наш опыт, существенно повышает эффективность разработки.

### Автоматизированное тестирование

В КуМире предусмотрен так называемый пакетный (*batch*) режим выполнения, при котором графический интерфейс системы не загру-

жается, а результаты выполнения программы выводятся в файлы. При этом создаются три файла вывода, хранящие содержимое трех областей главного окна системы КуМир:

- *Поле с текстом программы.* Оно также содержит информацию о подчёркивании синтаксических ошибок в программе.
- *Поле диагностики.* Содержит тексты сообщений о синтаксических ошибках.
- *Поле ввода-вывода.*

Нами был создан набор тестовых программ для выполнения в пакетном режиме и набор эталонных файлов вывода каждого теста. Наличие несоответствий между полученными выходными файлами, полученными при выполнении теста, и эталонными выходными файлами, говорит о найденных ошибках. Различия в выходных файлах первого типа означают ошибку в подчеркивании синтаксических ошибок; в файлах второго типа — неверную диагностику ошибок; в файлах третьего типа — нарушение процесса выполнения программы.

Для того чтобы автоматизировать процесс тестирования, был написан *скрипт тестирования*, который последовательно обрабатывает все тесты и сравнивает полученные данные с эталонными. Найденные несоответствия записываются в файл отчета. Файл отчета скрипта тестирования представляет собой последовательность пар различающихся строк из выходных файлов. Для каждой пары указывается имя выходного файла и номера строк.

## Как создаются эталонные тесты

Мы разделяем тестовые примеры на три типа:

**Выполнимые** — не содержат синтаксических ошибок и ошибок выполнения, выполняются нормально.

**С ошибками разбора** — содержат синтаксические ошибки.

**С ошибками выполнения** — не содержат синтаксических ошибок, но их выполнение завершается аварийно.

При разработке тестов мы рассматриваем все формальные правила описания языка и создаем примеры, следуя этим правилам. Составляя правильные конструкции, мы получаем выполнимый тест. Перебирая различные варианты ошибочного написания рассматриваемого правила, мы получаем набор тестов с ошибками. Таким образом, мы можем покрыть всю функциональность языка программирования.

## Тестирование интерфейса

Описанный выше метод позволяет тестировать только разбор и выполнение программ, в то время как необходимо также выполнять тестирование интерфейса. Для этого была написана система так называемых *интерфейсных тестов*, охватывающая функциональность всего интерфейса системы. Каждый такой тест представляет собой инструкцию для тестера с алгоритмом действий, которые ему необходимо совершить. Тестер должен выполнить требуемые действия и сравнить результат с ожидаемым.

Для создания системы интерфейсных тестов нами было разработано по возможности формальное описание интерфейса системы КуМир. Мы старались описать все состояния системы, действия пользователя в этих состояниях и реакции системы. Каждый раздел описания стал основой для написания интерфейсного теста или группы тестов. Такое описание, естественно, оказалось неполным, поэтому система интерфейсных тестов расширялась по мере отладки системы.

## Выводы

Даже в производственных системах, которые используются много лет, находятся ошибки[2]. То есть долговременная эксплуатация не может быть заменена никакими специальными системами тестирования. Тем не менее, при разработке системы обучения программированию КуМир хорошо себя зарекомендовали следующие методы тестирования.

- Создается корпус тестов, состоящий из тестов автоматического выполнения и интерфейсных тестов. Тесты обеих групп создаются на основе формальных описаний. Тесты автоматического выполнения создаются на основе описания языка КуМир, включая описания диагностики ошибок. Интерфейсные тесты создаются на основе специально подготовленного описания интерфейса.
- Тесты автоматического выполнения делятся на три группы: 1) Тесты синтаксических ошибок; 2) Тесты ошибок выполнения; 3) Тесты правильных программ. Результатами выполнения каждого такого теста являются текстовые файлы, содержащие диагностику синтаксических ошибок, диагностику ошибок выполнения, а также вывод программ. В перспективе будет преду-

смотрена возможность протоколирования всех промежуточных действий. Эталонные значения файлов прилагаются к каждому тесту автоматического выполнения. Для выполнения этих тестов в системе КуМир предусмотрен специальный пакетный режим и созданы средства тестирования. Эти средства позволяют выполнить все тесты в пакетном режиме, отметить тесты, результат выполнения которых отличается от эталонных, и указать, в чем состоит расхождение с эталоном. Это тестирование запускалось регулярно в течение последнего года разработки системы КуМир.

- Для тестирования интерфейса было создано описание возможных состояний системы КуМир с точки зрения пользователя, действий пользователя в этих состояниях и реакций системы. На основе этого описания был создан корпус интерфейсных тестов. Каждый тест по возможности включает максимально точные инструкции для человека, который ведет тестирование.

Такой протокол тестирования позволил избежать повторного внесения ошибок при развитии системы и существенно ускорил ее отладку.

## Литература

- [1] *Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н.* Информатика, 7–9 классы.
- [2] *Дейкстра Э.* Дисциплина программирования.