

УДК 004.415:378

## АВТОМАТИЗАЦИЯ ПРОВЕРКИ ВЫПОЛНЕНИЯ ГРАФИЧЕСКИХ ЗАДАЧ В САД СИСТЕМЕ

Подберезкин А.С., Подберезкин А.С., Колядко С.В.  
Научный руководитель – Полозков Ю.В., к.т.н., доцент

Контроль выполнения заданий является неотъемлемым и весьма трудоёмким этапом процесса обучения, особенно связанного с САД моделированием. При автоматизации проверки выполнения графических задач важными представляются два основных аспекта: во-первых, непосредственное соответствие заданного и построенного графических образов и, во-вторых, соблюдение методики графических построений. Тогда могут быть реализованы два способа автоматизации проверки задач: контроль итогового решения задачи и контроль процесса (мониторинг) решения задачи. Контроль итогового решения задачи носит статичный характер, а мониторинг решения характеризуется динамичностью, т.к. требует контроля действий, выполняемых пользователем в различные моменты времени. Однако для программной реализации мониторинга решения задач основу составляют алгоритмы контроля итогового решения.

1. Для разработки алгоритмов, требуемых для контроля итогового решения, прежде всего, выполняется анализ методики графических построений. В ходе этого анализа выявляются основные шаги построения графического образа. Каждый шаг построений представляет собой отдельную подзадачу. Подзадачи могут включать одно или несколько элементарных действий. В первом случае разрабатывается алгоритм, отвечающий за контроль соответствующего действия, который возвращает булево значение параметра правильности решения, значение полученного решения, значения перечислений верного и неверного решений, предназначенных для формирования комментариев по результатам контроля. В этом случае алгоритм может быть отнесен к категории **базового**, т.е. обеспечивающего контроль отдельной части задачи, не зависимой от правильности выполнения других шагов. Эти шаги и, соответственно, базовые алгоритмы являются обязательными для включения в состав сценариев решения задач независимо от интерпретации описательных условий.

2. Во втором случае разрабатывается один или несколько алгоритмов, отвечающих за контроль последовательности действий, выполняемых в рамках подзадачи. Такой алгоритм относится к категории **основных**, т.е. обеспечивающих контроль отдельной части задачи, которая может выполняться только после правильного исполнения шагов, контролируемых базовыми алгоритмами. Основной шаг может представлять собой отдельную часть задачи как зависимую, так и не

зависимую от правильности выполнения других основных шагов. Этот шаг и, соответственно, основной алгоритм не является обязательным для включения в состав сценариев решения различных однотипных задач.

В конечном итоге разрабатывается множество алгоритмов, позволяющих контролировать все шаги решения задачи по построению графического образа. Применяя определенные правила кодирования и систематизации, данные алгоритмы могут быть представлены в виде нижне-треугольной матрицы сценариев решения задач, последняя строка которой определяет совокупность требуемых алгоритмов для решения всей задачи – **производный** алгоритм. Производный алгоритм, представляет собой совокупность базовых, основных и других производных алгоритмов для контроля решения всех частей задачи. По сути, производный алгоритм описывает сценарий решения задания, т.е. набора задач. В ходе отработки каждого алгоритма этой совокупности может быть получен соответствующий отклик для формирования динамически или статически выдаваемого списка ошибок.

В качестве примера можно рассмотреть задачу построения звезды по заданным длинам лучей и выполнением лучей тонкой линией, контура – основной. Методикой обучения графическим построениям предусматривается выполнение следующих основных шагов:

- расчет углов между лучами, в частности, между короткими и длинными;
- построение длинных лучей, в частности, в количестве трех;
- построение коротких лучей, в частности, в количестве трех;
- построение отрезков контура звезды;
- контроль и редактирование типов линий лучей и контура.

Построение лучей производится путем векторного задания отрезков на основе заданной длины, расчетного угла и правил векторного построения в САД системе. Начальные точки всех лучей находятся в одной центральной точке звезды. Задание типов линий лучей и контура может производиться как в процессе задания соответствующих отрезков, так и после их построения.

Декомпозиция методики построения звезды позволяет выделить следующие необходимые алгоритмы контроля подзадач:

- контроль существования отрисованных графических примитивов (B1);
- контроль типов отрисованных графических примитивов, в данном случае отрезков (B2);
- контроль количества отрисованных отрезков с общим началом (B3);
- контроль количества отрезков и целостности контура, т.е. совпадения конечной точки текущего отрезка с начальной точкой следующего (B4);
- контроль количества отдельных графических примитивов (B5)

- контроль длины длинных лучей (O1);
- контроль длины малых лучей (O2);
- контроль углового положения длинных лучей (O3);
- контроль углового положения малых лучей (O4);
- контроль длины отрезков контура (O5);
- контроль принадлежности концевых точек лучей концевым точкам контура (O6);
- контроль типа линии лучей (O7);
- контроль типа линии отрезков контура (O8).

В целом решение данной задачи можно представить в виде множества алгоритмов указанных в последней строке матрицы сценариев решения задач (таблица 1).

Таблица 1. Матрица сценариев решения задач построения звезды (P1)

	B1	B2	B3	B4	B5	O1	O2	O3	O4	O5	O6	O7	O8	P1
B1	1													
B2	1	1												
B3	1		1											
B4	1			1										
B5	1				1									
O1	1	1	1			1								
O2	1	1	1				1							
O3	1	1	1					1						
O4	1	1	1						1					
O5	1	1	1	1						1				
O6	1		1	1						1	1			
O7	1		1									1		
O8	1			1									1	
P1	1	1	1	1	0	1	1	1	1	1	1	1	1	1

Следует отметить, что с использованием алгоритмов, входящих в состав данной матрицы можно вырабатывать новые сценарии решения подобных задач. Например, строить графический образ не из отрезков, а из дуг и т.п. Траектории обучения должны учитывать статистические данные по решаемости задачи и успеваемости обучающегося. По количеству правильных и неправильных ответов можно судить о наиболее легких и проблематичных алгоритмах, задачах и учебных тематик для конкретно взятого обучающегося. На основе полученных данных можно корректировать траекторию обучения путем переходов между задачами или переходов к теоретическим материалам.