

Г. Л. МУРАВЬЕВ, С. В. МУХОВ, А. Н. КЛИМОВИЧ
БрГТУ (г. Брест, Беларусь)

**СРЕДСТВА ОБУЧЕНИЯ КОНСТРУИРОВАНИЮ ПРОГРАММ
И ТРЕБОВАНИЯ К ИХ ХАРАКТЕРИСТИКАМ**

Развитие индустрии программных систем характеризуется значительным усложнением процессов разработки. Необходимый уровень обучения проектированию наряду с использованием систем программирования требует и специальных средств, автоматизирующих обучение.

Наиболее характерные направления и представители средств обучения представлены ниже. Это системы визуального программирования типа Алгоритм [1], использующие в качестве входного языка исполнимые схемы алгоритмов, что обеспечивает трассировку графических спецификаций и анализ корректности программ. Языковые системы типа русифицированной среды ЛЕГО-Лаборатория с языком Лого (диалектом языка ЛИСП) [2], близким по синтаксису к естественному (фирма Logo Computer Systems Inc), с процедурным стилем программирования. Система программирования КуМир (Комплект Учебных МИРов) [3, 4], базирующаяся на русифицированном алголо-подобном языке с возможностью исполнения программ в редакторе-ком-пиляторе версии Е-практикум. Версия 2000 (НИИСИ РАН РФ) использует библиотеку Qt и работает в ОС GNU/Linux и Windows [4]. Конструкторы типа «Библиотеки алгоритмов» [5], «Конструктора блок-схем» [6] обеспечивают дистанционное структурное проектирование алгоритмов, пошаговую детализацию с исполнением спецификаций в режиме анимации, интерпретации. Отдельное направление – тренажеры, web-визуализаторы тренажеров [7], реализуемые как web-приложения, исполняемые сервером или на стороне клиента, позволяющие управлять и визуализировать исполнение спецификаций.

Результаты сравнительного анализа базовых характеристик инструментов редактирования, структурирования, комментирования спецификаций, отладки, средств генерации, тестирования и т. п. систем, автоматизирующих обучение, представлены в таблице. Здесь в первой графе приведены требования к перспективной системе, во второй – характеристики системы КуМИР, в третьей – характеристики типовой системы обучения, полученные по «усредненным» данным таких систем как [1–7] и другим системам, в четвертой графе приведены характеристики типовой системы программирования. Символ «+» означает наличие средства, инструмента в системах 2–4 типов и необходимость средства для систем первого типа, «*» – желательность средства для систем первого типа и частичность наличия в системах остальных типов. Системы 2–4 отличаются использованием данных средней типизации, для системы первого типа достаточно использование стандартных скалярных и структурных типов, в том числе, имеющих математические аналоги. В системах 2–4 используются строчные и фрагментарные комментарии, а в системе первого типа необходимы также комментарии «встраиваемые» в текст. Исполнимость систем 2–3 основана, как правило, на интерпретации спецификаций (И), что достаточно и для систем первого типа, где также перспективно использование виртуальной машины (В).

Таблица – Характеристики систем

Характеристика	система			
	1	2	3	4
Средства прототипирования, нисходящего проектирования, спецификации:	+	-	-	*
- данных, потоков данных	+	-	-	-
- задач, сценариев использования	+	-	-	-
- спецификация модулей	+	*	-	*
- спецификация иерархий модулей	+	-	-	-
- спецификация интерфейсов	+	-	-	-
Язык спецификаций текстов:	+	+	+	+
- модульность, сегментирование	+	+	+	+
- структурирование, отступы, сдвиги	+	+	-	*
- автоструктурирование текстов	+	-	-	-
- русифицированность текстов	+	+	+	*
- склоняемость имен, идентификаторов	+	-	-	-
- синонимичность команд	+	-	-	-
- читаемость команд	+	+	-	-
- «традиционные» комментарии	+	+	+	+
- выделение «синтаксиса» текстов	+	+	+	*
Средства редактирования:	+	+	+	+
- многооконность среды	+	+	-	+
- графический интерфейс пользователя	+	+	+	+
- удаленный доступ, дистанционность	*	-	*	-
- русифицированность средств	+	+	+	-
Средства автогенерации:	+	*	*	+
- генерация интерфейсов приложений	+	-	-	*
- генерация каркасов приложений	+	-	-	+
- получение ехе-модуля программы	*	*	-	+
- генерация ЯВУ-текста программы	+	-	-	-
- исполнимость спецификаций	В	И	И	К
Средства тестирования:	+	*	-	-
- «прокрутка» каркасов приложений	+	-	-	+

Характеристика	система			
	1	2	3	4
- библиотека эталонных примеров	+	+	*	-
- тестирование на базе эталонов	*	*	-	-
- автотестирование программ	*	-	-	-
Средства верификации:	*	-	-	-
- для типовых конструкций языка	*	-	-	-
- для проектных спецификаций	*	-	-	-
Средства документирования проектных решений, протоколы тестирования	+	*	*	*

Обучение конструированию программ предполагает выработку специальных навыков работы, начиная с формализации требований к проекту, его прототипирования, модульной разработки и заканчивая спецификацией алгоритмов решаемых задач, кодированием их на языках программирования и т. д. Однако в рассмотренных системах и средах обучение конструированию программ в части работы со спецификациями, архитектурой, модульного проектирования, прототипирования, автогенерации (включая генерацию каркасов программ) поддерживаются слабо. Как правило, отсутствуют отладчики спецификаций, нет средств генерации текстов на языках высокого уровня (ЯВУ). Оценка корректности проектных решений, спецификаций, поиск и локализация ошибок, тесно связанные с обучением конструированию программ, разработке алгоритмов, как правило, производятся вручную. Кроме этого, используемые лингвистические средства, инструменты обеспечивают недостаточную документированность принимаемых проектных решений и читаемость текстов, спецификаций.

В работе выполнена классификация средств, автоматизирующих обучение, приведены результаты сравнительного анализа их базовых характеристик. Сформулированы требования к характеристикам и составу перспективных средств обучения. Показана целесообразность их построения в виде компьютерных сред, базирующихся на принципах прототипирования программ, структурного подхода к разработке, исполнимости спецификаций, генерации проектных решений.

ЛИТЕРАТУРА

1. Миленький, А.В. Обучение основам алгоритмизации – новые возможности / А.В. Миленький // Информационные технологии в образовании: материалы XI междунар. конференции-выставки [Электронный ресурс]. – 2001. – Режим доступа: <http://www.ito.su/2001/ito/I/1/I-1-14.html>.
2. Организация изучения основных алгоритмических конструкций в среде ЛогоМиры [Электронный ресурс]. – 2010. – Режим доступа – <http://www.edu-zone.net/show/47800.html>.
3. [Электронный ресурс]. – 2011. – Режим доступа: <http://lpm.org.ru/kumir>.
4. Яшина, Е.Ю. Система КуМир для WINDOWS / Е.Ю. Яшина, А.Г. Леонов // [Электронный ресурс]. – 2009. – Режим доступа: <http://www.ito.su>.
5. Конструктор программ на сайте «Библиотека алгоритмов» [Электронный ресурс]. – 2010. – Режим доступа: <http://alglib.chat.ru>.
6. Кузин, С.Г. Компьютерная технология обучения основам алгоритмизации / С.Г. Кузин, Р.О. Митин, И.С. Скрибловский // [Электронный ресурс]. – 2008. – Режим доступа: www.unn.ru/vmk/graphmod.
7. Якименко, О.В. Применение программ-тренажеров в обучении программированию / О.В. Якименко // [Электронный ресурс]. – 2007. – Режим доступа: yakimenkoov@tspu.edu.ru.