кументе 2) в строке математической формулы Y координата нижних левых граничных пикселей символов имеет в разы большую дисперсию по сравнению с той же величиной строки текста, ввиду того, что технических и научных публикациях используются «моновысотные» шрифты. Таким образом, сперва выделяются области, обрамлённые широкими пустотами, далее, анализируется дисперсия Y координаты самого левого нижнего пикселя каждого символа.

Полученные результаты позволяют сделать вывод о хорошем качестве выделения символов и возможности использования данного подхода для решения основной задачи – лексического анализа текста и выражений в математической нотации.

Библиографический список

1. Fateman R.J., Tokuyasu T. Progress in recognizing typeset mathematics // Proceedings of SPIE The International Society for Optical Engineering. – 1996. – Vol. 2660. – P. 37–50.

2. Garain, U., Chaudhuri, B.B., A Syntactic Approach for Processing Mathematical Expressions in Printed Documents // International Conference on Pattern Recognition'00. – Spain, 2000, vol. IV, P. 523-526.

3. Южиков В.С. Сегментация изображения страницы с текстом с текстом // Вестник ИжГТУ. – 2007. – №3. – С. 84-47.

4. Issam El-Naqa, Yongyi Yang, Miles N. Wernick. Support vector machine learning for detection of microcalcifications in mammograms. [Электронный ресурс]. – Режим доступа: http://www.ipl.iit.edu/IPL-Conferance_papers/isbl.pdf - Загл. с экрана.

5. Lee S-W., Lee D.-J., Park H.-S., A New Methodology for Gray-scale Character Segmentation and Recognition // IEEE Transactions on pattern analysis and machine intelligence. – IEEE, 1996, vol. 18, P. 1045-1050.

Анализ возможностей объектной системы CLOS[1]

О.Н. Половикова

АлтГУ, г. Барнаул

Функциональный язык LISP был на пике своей популярности в 70-80 годах прошлого столетия, когда активно использовался для написания программных систем в области искусственного интеллекта. На сегодняшний день растет интерес к этому языку, при этом исследова-

[1] Работа выполнена при поддержке аналитической ведомственной целевой программы «Развитие научного потенциала высшей школы (2009-2011 годы)» (код проекта №2.2.2.4/4278).

тели замечают, что индекс его популярности растет довольно быстро. В опубликованном компания TIOBE Software рейтинге популярности языков программирования за прошедший год (2010) язык Lisp с 23 мета (апрель 2010 года) передвинулся на 15 место (апрель 2011 года) [1].

Промышленным стандартом языка Лисп является язык Common Lisp, который вытеснил большинство других диалектов. Одним из немаловажных фактов, которые способствуют делать выбор для написания современных прикладных систем в пользу этого язык, являются возможности объектной системы Common Lisp (CLOS). Следует заметить, что данная объектная система предлагает достаточно отличающуюся от других языков реализацию принципов ООП [2, 3]. Для организации поведения экземплярами класса применяется множественная диспетчеризация, в отличие от Си++ и Java, где для этих целей используется передача объектам сообщений. Поэтому в CLOS поведение классов определяется моделью обобщенных функций, а не жестко закрепляется набором методов, которые объявляются только при создании класса.

Отличительный от многих других объектно-ориентированных языков подход к реализации принципов полиморфизма и инкапсуляции классов имеет свои преимущества и недостатки. Поэтому целью данного исследования является анализ возможностей системы CLOS, а также формирование базовых требований и условий к практическим задачам, для реализации которых следует использовать данную объектную систему.

Основными преимуществами объектной системы Common Lisp перед аналогами:

1) динамическое комбинирование действующего метода реализации вызова функции для объекта класса;

2) независимость объявления методов обобщенных функций от места и времени объявления классов;

3) методы обобщенной функции можно использовать в качестве аргументов для функций высшего порядка.

Рассмотрим представленные выше преимущества.

Каждому вызову функции для объекта определенного класса может соответствовать несколько методов реализаций. Причем, в процессе работы с программой можно добавлять, удалять или модифицировать их реализации. Построение так называемого эффективного (действующего, рабочего) метода, который и будет отвечать за выполнение указанной функции, выполняется также динамически и зависит от выбранного или созданного комбинатора. Следует заметить, что дан-

ное преимущество обеспечивается не только механизмами множественной диспетчеризации методов, но и функциональностью языка Лисп.

Такой подход к организации поведения сущностей предоставляет широкий спектр возможностей для наиболее адекватного проецирования модели предметной области на реализацию функциональной части экземпляров. Выделим подзадачи, для решения которых необходимо использовать описанный выше способ реализации поведения классов.

1. Неопределенность особенностей выполнения или алгоритм исполнения разрешается только в момент работы с программой.

2. Приемлемые итоговые методы класса строятся на каком-либо способе комбинирования методов классов родителей.

3. Используется сложная выборка реализаций родительских методов, не отвечающая линейной комбинации по иерархии наследования классов.

Независимость объявления методов обобщенных функций от места и времени объявления классов позволяет размещать в разных программных блоках реализации методов и классов. В процессе работы интерпретатора можно модифицировать поведение классов (изменяя функциональную составляющую методов), при этом не требуется корректировать классы или объекты. Учитывая то, что методы являются функциями, то их можно использовать как аргументы для функций высшего порядка, таких как `MAPCAR и др.`

В заключение, следует отметить, что представленные в статье преимущества CLOS, не перекрывают весь объём возможностей множественной диспетчеризации, используемой для организации поведения объектов системы. Явное отсутствие спецификаторов, позволяющих управлять видимостью атрибутов и методов класса при его построении, относят к изъянам при поверхностном анализе конструкций языка. При этом, как правило, не учитывается, что отсутствие привычной функциональности по инкапсуляции хранимых в классе данных и поведения можно частично перекрыть средствами сокрытия символов в текущем пакете.

Прямым назначением пакетов является возможность создания для своей программы собственного пространства имён, чтобы не возникало конфликтов между одноименными символами в различных программах, загруженных в одну Lisp-среду. Использование пакетов не позволяет контролировать доступ к объектам, их слотам или к методам обобщенных функций, они явно не предоставляют механизмов для ограничения видимости символов, будь это имя переменной или имя функции. Но при этом есть возможность в рамках одного пакета де-

лать недоступными символы из других пакетов, разрешать конфликты при наследовании одинаковых символов из разных пакетов, составлять публичные API библиотеки, для их использования во внешних программных продуктах.

Библиографический список

1. TIOBE Programming Community Index for April 2011 April Headline: Lua is approaching the top 10 of the TIOBE index [Электронный ресурс]. URL: [http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (дата обращения 01.05.2011).

2. Kent M. Pitman. Condition handling in the lisp language family. 2001.

3. Peter Seibel. Practical Common Lisp. 2006.

Место системы управления качеством образовательной услуги в ИАИС ВУЗа

Ф.А. Попов, Е.С. Молодцова

БТИ АлтГТУ, г. Бийск

Широкий спектр направлений деятельности представляет ВУЗ как сложное в управлении учреждение. В последние годы особое место занимает деятельность по управлению качеством образовательной услуги (ОУ). Наиболее эффективным и комплексным подходом к управлению качеством ОУ является подход, связанный с внедрением в управленческую культуру ВУЗа систем менеджмента качества (СМК), основанных на требованиях международных и национальных стандартов и моделей систем качества, чаще всего – стандартов серии ИСО 9001.

Анализ применения информационных технологий в сфере образования показывает, что одной из основных тенденций их развития является все более полный охват направлений деятельности ВУЗа в рамках интегрированной автоматизированной информационной системы (ИАИС), состоящей из комплекса функционально-ориентированных подсистем, обеспечивающих решение основных задач ВУЗа. Однако задачам управления качеством в рамках СМК при проектировании и создании ИАИС практически не уделяется внимания.

Очевидно, что для эффективного функционирования СМК в рамках подсистемы управления качеством автоматизированы должны быть все задачи, возникающие на этапах разработки, внедрения и поддер-