

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ

УДК 004.432.2

Т. Н. Лебедева, Л. С. Носова

ФОРМАЛИЗАЦИЯ ДАННЫХ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ 1С

Будущий инженер-программист должен владеть навыками разработки малых и больших программных комплексов с использованием современных языков и систем программирования. Для того чтобы создать качественный программный продукт, необходимо продумать логику построения модулей, программный интерфейс, описать соответствующие алгоритмы взаимодействия системы. Важно правильно построить диалог пользователя с программным комплексом. В связи с этим студент должен иметь представление о формальных системах и грамматиках, позволяющих детально понять внутреннее состояние используемых алгоритмических конструкций и описать их на практике. Несмотря на то, что язык программирования 1С не включен в рейтинг языков программирования, используемых при разработке программного обеспечения, он является популярным для разработки корпоративных информационных систем. В связи с этим встает вопрос о его изучении в вузовском курсе. Отмечены особенности языка программирования 1С и трудности, возникающие при его изучении. Описаны средства формализации данных в языке программирования 1С. Поскольку наиболее глубокое понимание семантических особенностей языка программирования, правильное использование конструкций наблюдаются при использовании в учебном процессе синтаксических диаграмм, рекомендуется включать в изучение языков программирования элементы теории языков и формальных грамматик.

Ключевые слова: язык программирования, формализация данных, синтаксические диаграммы.

Введение

Состояние проблемы. В настоящее время, в эпоху рыночных отношений, предъявляются высокие требования к профессиональной компетентности специалиста. Инженер-программист должен обладать достаточными теоретическими знаниями в области программирования и должен уметь применять их при решении поставленных производственных задач.

Любой конкурентоспособный программный продукт должен быть оснащен удобным, интуитивно понятным интерфейсом и специальным языком, предназначенным для описания входных и выходных данных. Для этого программист-разработчик должен предусмотреть эффективный контроль правильности входных фраз и их однозначную интерпретацию, не допускающую двусмысленности. Входные данные передаются в прикладную систему с помощью естественного языка, заданных цепочек символов описываемой предметной области. С помощью описания данных в текстовом формате можно описать любую структуру всевозможных документов, трехмерный виртуальный мир, задать графический интерфейс пользователя, а также других объектов некоторых моделей реального мира. Для корректного составления и описания текстовых данных с их последующим процессом распознавания и интерпретации используются специальные методы их формализации, анализа и преобразования. К ним мы можем отнести синтаксис и семантику языка. В основе этих методов лежит теория языков и формальных грамматик [1].

Теоретико-методологическую основу исследования составляют работы в области теоретической информатики (Ф. Л. Бауэр, М. Брой, В. М. Глушков, Х. Роджерс, В. А. Успенский и др.), а также теории разработки языков программирования и автоматов (А. Ахо, Ф. Вайнгартен, М. Зелковиц, Т. Пратт, В. А. Серебряков, Дж. Ульман и др.).

Новые языки программирования и их трансляторы появляются ежегодно. По данным Tiobe Software [2], в настоящее время насчитывается около 250 языков программирования. Ка-

ждый вновь создаваемый язык отличается целым набор функциональных возможностей, таких как строгая типизация, абстракция, формализуемость, объектность, кроссплатформенность, масштабируемость и т. д. Это обусловлено быстрым развитием аппаратно-программных систем, необходимостью решения все более сложных задач, поэтому процесс создания новых сред и систем программирования в этой области не прекращается.

Любой язык программирования характеризуется набором правил: лексических, синтаксических и семантических. Именно они определяют внешний вид программы и действия исполнителя программы под ее управлением. Если обратиться к определению языка программирования, то он выступает как формальная знаковая система, предназначенная для записи компьютерных программ [3].

Мы считаем целесообразным включать изучение вопросов формализации в процесс освоения студентами языка программирования 1С.

Характеристика языка программирования 1С

Язык программирования 1С в настоящее время не имеет официального названия и является встроенным в технологическую платформу 1С:Предприятие 8. Его изучение не представляет сложности, т. к. его синтаксис прост и понятен, однако разработчикам на 1С необходимо знать предметную область, для которой создается приложение, и именно здесь возникают трудности в освоении встроенного языка.

Еще к одной особенности встроенного языка 1С относят его двуязычный синтаксис: написание команд возможно как на английском языке (латиница), так и на русском (кириллица). В связи с трудностями перевода терминов предметной области разработчики-программисты 1С используют в своей деятельности код на кириллице. Это было сделано также для улучшения восприятия алгоритмов. Опыт практической деятельности показывает, что многие начинающие программисты-студенты при программировании часто применяют правила русского языка: осуществляют склонение идентификаторов, используют при программировании окончания и синонимы. Например, вместо «Сообщить» используют «Сообщение», «Показать». Или «Для каждой ПриходнойНакладной» вместо «Для каждого ПриходнаяНакладная», или «Цены» и «Цена» и т. д. Скорее всего, данные ошибки возникают, когда студент не имеет четкого представления об именовании переменных, служебных слов и т. п.

Почти все зарезервированные слова, имена типов значения, свойств, событий, методов встроенного языка имеют два имени: английское и русское. Исключение – слова, не имеющие аналогов в русском (например, Null). Одновременно в коде программы можно использовать русские и английские слова. При этом компилятор нечувствителен к регистру.

Синтаксические диаграммы языка программирования 1С

Требования, предъявляемые к выбору идентификатора переменной, объявленной пользователем процедуры или функции, в системе 1С аналогичны требованиям выбора идентификаторов в других языках программирования. В общем виде синтаксическая диаграмма идентификатора приведена на рис. 1.

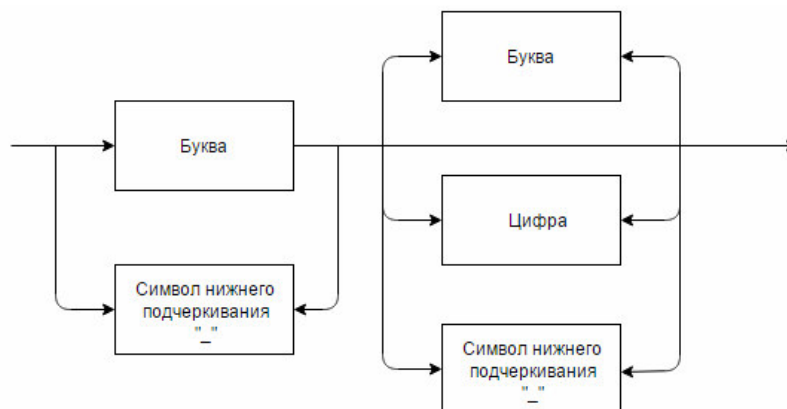


Рис. 1. Диаграмма идентификатора

Встроенный язык IC поддерживает примитивные типы данных:

Число: любое десятичное число. Для разделения целой и дробной части используется точка. Максимально допустимая разрядность числа – 38 знаков. Пример диаграммы целых чисел приведен на рис. 2, вещественных – на рис. 3.

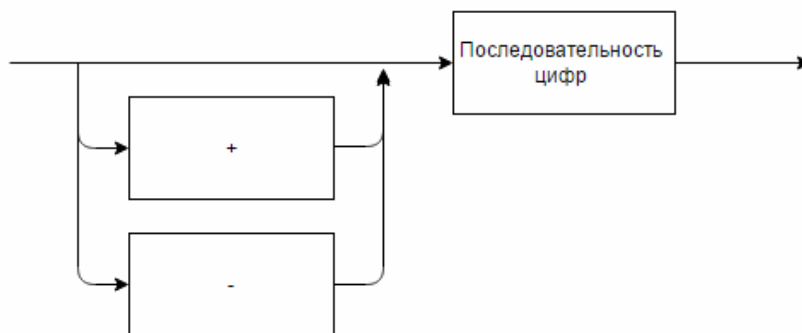


Рис. 2. Диаграмма целых чисел



Рис. 3. Диаграмма вещественных чисел

Строка: значения типа содержат строку произвольной длины в формате Unicode, можно использовать слова на различных языках. Набор символов заключен в двойные кавычки (“”). Диаграмма строки показана на рис. 4.



Рис. 4. Диаграмма строки

Дата: неразделимо хранятся дата и время в диапазоне от 1 января 1 года 0 часов 0 минут 0 секунд до 31 декабря 9999 года 23 часов 59 минут 59 секунд. Допустимо использовать различные разделители, например «/», «:», «-» и др.

Булево: принимает только два значения: Истина или Ложь, которые задаются соответствующими литералами – True или False соответственно (результат вычисления логических операций).

Неопределенно (Undefined): это значение определяет пустое значение с незадачным типом, т. е. не принадлежащее ни к одному другому типу. Представлено пустой строкой и используется для очищения существующих переменных и контейнеров.

Null: пустое значение с незадаанным типом в базе данных. Используется исключительно для определения отсутствующего значения при работе с базой данных. Представлено пустой строкой.

Тип: значение типа используется для идентификации типов значений, например для определения и сравнения типов. Не имеет литералов, возвращается функциями встроенного языка «ТипЗнч» и «Тип».

Для работы с переменными во встроенном языке определен оператор присваивания. Система 1С:Предприятие поддерживает неявное определение переменной, т. е. в момент присвоения переменной значения. Для передачи постоянного значения в переменную используется оператор присваивания (=). При описании алгоритма операторы разделяются точкой с запятой (;). Диаграмма оператора присваивания приведена на рис. 5.

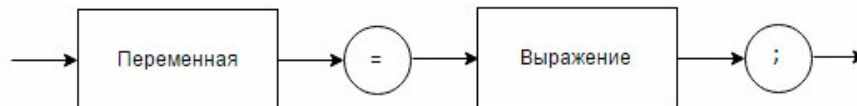


Рис. 5. Диаграмма оператора присваивания

Для построения алгоритмов во встроенном языке 1С определены синтаксически конструкции – операторы (рис. 6).

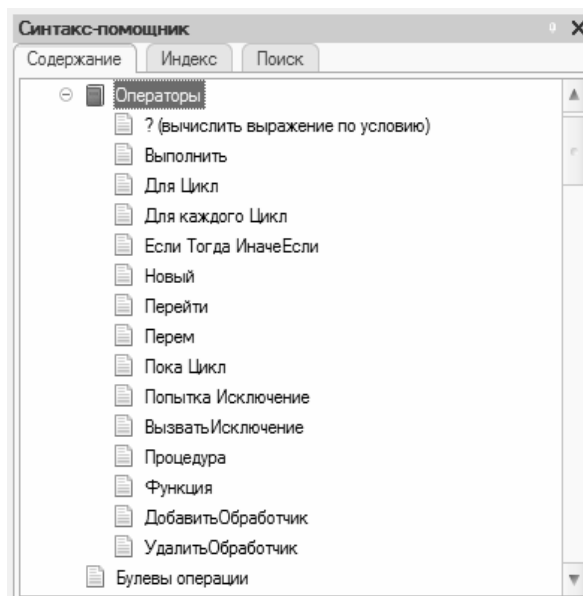


Рис. 6. Содержание справочной информации

Рассмотрим некоторые операторы и их синтаксические диаграммы.

Условный оператор *Если*, как и в любом языке программирования, представлен двумя формами – полной и неполной. Его синтаксис в русскоязычном и англоязычном описании представлен в табл. 1.

Таблица 1

Синтаксис оператора *Если*

Русскоязычный	Англоязычный
Если <Логическое_выражение> Тогда //операторы; [ИначеЕсли <Логическое_выражение> Тогда] //операторы; [Иначе] //операторы; КонецЕсли;	If <Логическое_выражение> Then //операторы; [Elsif <Логическое_выражение> Then] //операторы; [Else] //операторы; EndIf;

Синтаксическая диаграмма выглядит следующим образом (рис. 7).

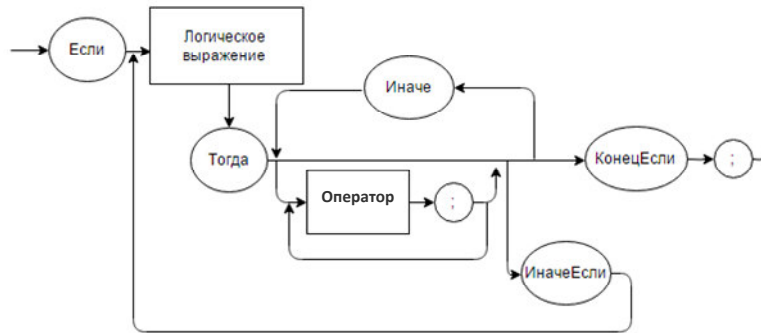


Рис. 7. Диаграмма условного оператора

Оператор цикла *Пока* имеет следующий синтаксис (табл. 2).

Таблица 2

Синтаксис оператора *Пока*

Русский	Англоязычный
Пока <Логическое_выражение> Цикл //операторы; КонецЦикла;	While <Логическое_выражение> Do //операторы; EndDo;

Синтаксическая диаграмма оператора приведена на рис. 8.

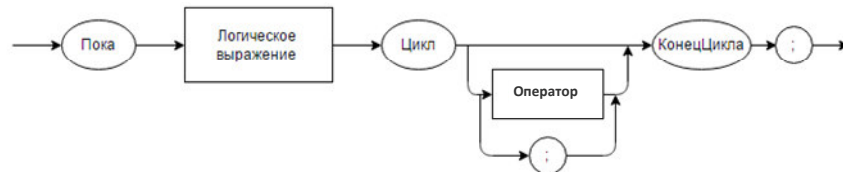


Рис. 8. Диаграмма оператора цикла

Возможности редактора кода 1С:Предприятие 8

Разработчики технологической платформы стараются, чтобы редакторы кода соответствовали современным тенденциям развития редакторов программного кода. Начиная с версии 1С:Предприятие 8.3 реализованы возможности автоматического выделения цветом идентификаторов, конструкций, находящихся под текстовым курсором. Например, выделяются начало и конец процедуры, начало и конец цикла, парные скобки и т. п. [4].

Для снижения вероятности ошибок и организации синтаксически правильных программ редактор кода позволяет выбирать операторы и конструкции с помощью механизма шаблонов:

1. Набор слова и нажатие Ctrl+Q. Например, при наборе слова «Проц» и вызова шаблона редактор предоставит следующие варианты (рис. 9).

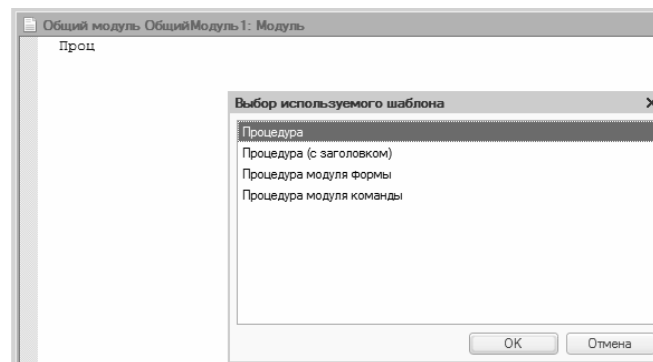


Рис. 9. Механизм шаблонов

После выбора шаблона получится следующее (рис. 10).

```

// <Описание процедуры>
//
// Параметры
// <Параметр1> - <Тип.Вид> - <описание параметра>
// <продолжение описания параметра>
// <Параметр2> - <Тип.Вид> - <описание параметра>
// <продолжение описания параметра>
//
Процедура МояПроцедура ()
|
КонецПроцедуры // МояПроцедура ()

```

Рис. 10. Код, полученный с помощью механизма шаблонов

2. В пункте «Шаблоны текста» контекстного меню в редакторе кодов имеется возможность выбора шаблонов (рис. 11).

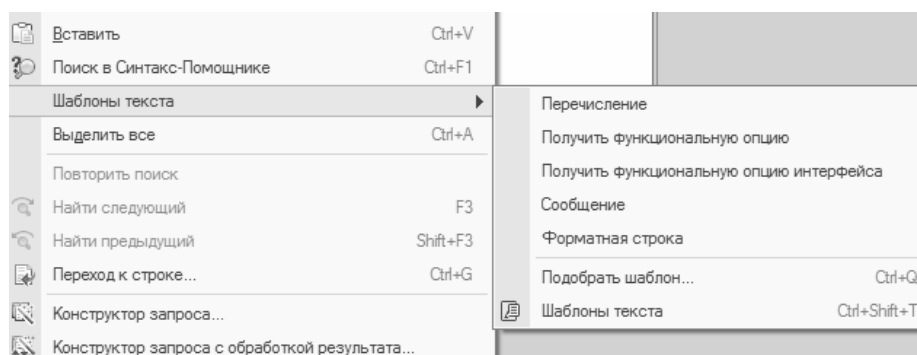


Рис. 11. Шаблоны текста

Автодополнение кода также осуществляется по нажатию `Ctrl+'`.

В отличие от других систем программирования, технологическая платформа позволяет создавать лишь объекты, а их визуальное расположение на форме происходит автоматически, что не отвлекает разработчика на построение интерфейса пользователя и проводит к стандартному виду всех форм прикладного решения.

Механизм бизнес-процессов технологической платформы 1:Предприятие 8

Еще одним средством формализации может выступать один из прикладных механизмов технологической платформы – механизм бизнес-процессов.

Данный механизм создан для описания, создания и управления выполнением бизнес-процессов. Он предназначен для автоматизации цепочек связанных операций, направленных на достижение цели, обычно в контексте организационной структуры, которая задает функциональные роли и связи [5].

Механизм предлагает готовую стратегию автоматизации деятельности организации. Для описания простого бизнес-процесса визуально задается схема маршрута с указанием условий ветвления в узловых точках. Все остальное система выполняет автоматически. Для сложных бизнес-процессов необходима тесная их увязка с функциями прикладного решения [5].

Вся логика бизнес-процессов визуально отражается в карте маршрута. Карта представляет собой связанный граф. С помощью этого графа легко описываются алгоритмы условных переходов, реакции процесса на различные события. Создается такой граф с помощью встроенной графической схемы. Примеры карт маршрутов представлены ниже: линейный алгоритм (рис. 12), условный (рис. 13) и циклический (рис. 14).

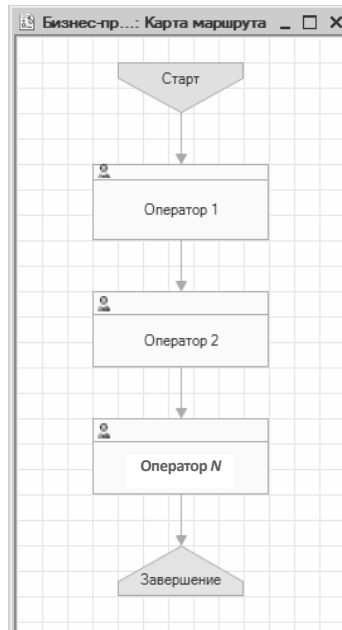


Рис. 12. Линейный алгоритм

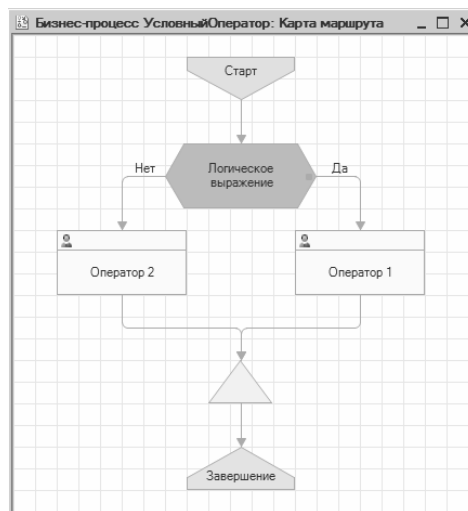


Рис. 13. Условный алгоритм

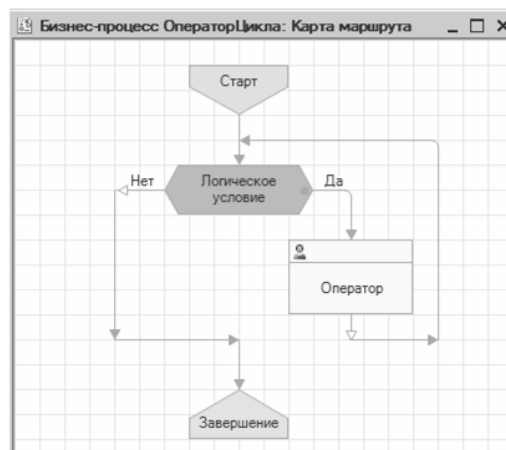


Рис. 14. Циклический алгоритм

Данный механизм можно использовать для формализации процессов предприятия, создания блок-схем, алгоритмов действий, последовательности применения определенных документов, информационных массивов данных, действий и процедур и т. д., направленных на получение конечного результата. Применение карт маршрутов бизнес-процессов способствует повышению когнитивности (понимаемости), эргономичности представленной информации, позволяет абстрагироваться от несущественных факторов, мыслить в терминах рассматриваемых объектов.

Методические аспекты изучения встроенного языка 1С

В методическом аспекте отдельно особенности встроенного языка программирования 1С не рассматривались. Его изучение происходит во время работы с технологической платформой 1С:Предприятие 8. Знакомство с платформой проводится для студентов разных направлений, однако для всех них встроенный язык уже не является первым языком программирования. В этой связи его освоение является достаточно быстрым, легким и по большей части самостоятельным. Практическая работа основывается на выполнении заданий по шаблону, а также по методу проектов. Каждой группе выдается задание на разработку информационной базы в выбранной предметной области: образование, туризм, жилищное хозяйство, банковская сфера и др. Студенты должны распределить роли в группе, определить структуру справочников, изучить примеры документов, определить бизнес-процессы и т. д. [6].

Полученный опыт позволил нам выделить следующие особенности освоения встроенного языка программирования 1С. При первом знакомстве с алгоритмами, написанными на кириллице, студенты выражают неоднозначное мнение. Однако затем происходит привыкание, кроме того, с помощью русскоязычного кода легче описывается предметная область. Изучение конструкций языка осуществляется самостоятельно с помощью Синтаксис-помощника (Если то, Для каждого, Процедура и т. д.). В случае возникновения вопросов по встроенному языку предлагается изучить содержание синтаксического помощника. При создании обработок, внутренних и внешних, студенты отмечают простоту и удобство их разработки, что обеспечивается внутренними механизмами платформы. В отличие от других систем программирования, технологическая платформа позволяет создавать лишь объекты, а их визуальное расположение на форме происходит автоматически, что не отвлекает разработчика на построение интерфейса пользователя и приводит к стандартному виду все формы прикладного решения. Использование встроенных настроек системы для написания синтаксически правильного кода позволяет снизить количество ошибок, но не освобождает от ошибок семантических.

Выводы

Как показывает практика, использование встроенных возможностей среды программирования (автодополнение, шаблоны, выделение кода цветом) помогает программисту избежать синтаксических ошибок при программировании. Однако наиболее глубокое понимание семантических особенностей языка программирования, правильное использование конструкций приходит при использовании в учебном процессе синтаксических диаграмм. В связи с этим мы считаем необходимым включение в изучение языков программирования элементов теории языков и формальных грамматик.

СПИСОК ЛИТЕРАТУРЫ

1. *Лебедева Т. Н.* Особенности изучения теории формальных языков и грамматик в вузе / Т. Н. Лебедева // Методология и методика информатизации образования в многоступенчатой структуре высшей школы: материалы Всерос. науч.-практ. конф. (Смоленск, 7–8 декабря 2009 г.). Смоленск: Изд-во СмолГУ, 2009. С. 61–66.
2. *Tiobe Software* // URL: <http://www.tiobe.com/index.php/content/company/Home.html/> (дата обращения: 15.04.2015).
3. *Могилев А. В.* Информатика: учеб. пособие для студ. пед. вузов / А. В. Могилев, Н. И. Пак, Е. К. Хеннер. М.: Академия, 2012. 848 с.
4. *Новые возможности подсветки синтаксических конструкций* // URL: // URL: <http://v8.1c.ru/o7/201308ME/index.htm/> (дата обращения: 15.04.2015).
5. *Бизнес-процессы, механизм* // URL: http://v8.1c.ru/overview/Term_000000270.htm/ (дата обращения: 11.06.2015).
6. *Носова Л.С.* Организация работы студентов инженерных специальностей с технологиями «1С» / Л. С. Носова // Информатика и образование. 2015. № 1 (260). С. 20–23.

Статья поступила в редакцию 6.05.2015,
в окончательном варианте – 15.06.2015

ИНФОРМАЦИЯ ОБ АВТОРАХ

Лебедева Татьяна Николаевна – Россия, 454084, Челябинск; Южно-Уральский институт управления и экономики; канд. пед. наук, доцент; доцент кафедры «Информационные технологии и системы»; lebedevatn@mail.ru.

Носова Людмила Сергеевна – Россия, 454084, Челябинск; Южно-Уральский институт управления и экономики; канд. пед. наук; доцент кафедры «Информационные технологии и системы»; nosovals@mail.ru.



T. N. Lebedeva, L. S. Nosova

FORMALIZATION OF DATA IN THE PROGRAMMING LANGUAGE 1C

Abstract. The future software engineer must have the skills of programming simple and complex software systems, using modern programming languages and systems. In order to create a quality product, it is necessary to consider the logic of building up the modules, software interface, to describe the interaction of the corresponding algorithms. It is important to build up a dialogue between a user and software package. In this regard, the student must have an idea of the formal systems and grammars, allowing fully understand the internal state of the used algorithmic structures and describe them in practice. Despite the fact that the programming language 1C is not included in the rank of the programming languages, used during the software development, it is popular for the development of the corporate information systems. So the question of studying this program language is becoming actual. The paper states the features of the programming language 1C and difficulties the students face while its studying. The means of formalizing the data in the programming language 1C are described. Since the most profound understanding of the semantic features of the programming language and the correct use of the structures are observed while using syntax diagrams in the classroom, it is recommended to include the elements of the theory of languages and formal grammars into the study of the programming languages.

Key words: programming language, data formalization, syntax diagrams.

REFERENCES

1. Lebedeva T. N. Osobennosti izucheniia teorii formal'nykh iazykov i grammatik v vuze [The features of studying the theory of formal languages and grammars at the university]. *Metodologiya i metodika informatizatsii obrazovaniia v mnogostupenchatoi strukture vysshei shkoly. Materialy Vserossiiskoi nauchno-prakticheskoi konferentsii (Smolensk, 7–8 dekabria 2009 g.)*. Smolensk, Izd-vo SmolGU, 2009. P. 61–66.
2. *Tiobe Software*. Available at: <http://www.tiobe.com/index.php/content/company/Home.html/> (accessed: 15.04.2015).
3. Mogilev A. V., Pak N. I., Khenner E. K. *Informatika* [Information science]. Moscow, Akademiia Publ., 2012. 848 p.
4. *Novye vozmozhnosti podsvetki sintaksicheskikh konstruksii* [New possibilities of highlighting of the syntax structures]. Available at: <http://v8.1c.ru/o7/201308ME/index.htm/> (accessed: 15.04.2015).
5. *Biznes-protsessy, mekhanizm* [Business processes, mechanism]. Available at: http://v8.1c.ru/overview/Term_000000270.htm/ (accessed: 11.06.2015).
6. Nosova L. S. Organizatsiia raboty studentov inzhenernykh spetsial'nostei s tekhnologiyami «1S» [Organization of the students' work of the engineering specialties with 1C technologies]. *Informatika i obrazovanie*, 2015, no. 1 (260), pp. 20–23.

The article submitted to the editors 6.05.2015,
in the final version – 15.06.2015

INFORMATION ABOUT THE AUTHORS

Lebedeva Tatyana Nikolaevna – Russia, 454084, Chelyabinsk; South Ural Institute of Management and Economics; Candidate of Education Science, Assistant Professor; Assistant Professor of the Department "Information Technologies and Systems"; lebedevatn@mail.ru.

Nosova Lyudmila Sergeevna – Russia, 454084, Chelyabinsk; South Ural Institute of Management and Economics; Candidate of Education Science; Assistant Professor of the Department "Information Technologies and Systems"; nosovals@mail.ru.

