На сегодняшний день, необходимо учесть всевозможные отклонения в системе, уметь адаптироваться к внезапным авариям, утечкам, отказам насосных систем, путем создания универсальной модели системы ППД [5].

Для повышения энергоэффективности необходимо предугадывать состояние системы при внесении каких-либо изменений, использования такой модели, позволит оптимизировать весь процесс закачки, что повысит надежность системы и точность соответствия режимов закачки воды в, исходя из требований в тот или иной момент времени.

Основной посыл для возможности реализации этих задач, состоит в достаточно высокой управляемости системы ППД, что позволяет модернизировать и более качественно автоматизировать систему.

1. Стрекалов, А.В. Системный анализ и моделирование гидросистем поддержания пластового давления. – Тюмень.: ИФ «Слово», 2002. – 324 с.
2. Зуев, К.И. Автоматизация систем водоснабжения и водоотведения// Зуев, К.И. Издательство ВлГУ, 2016. – 224 с.
3. Прокопчина, С.В. Автоматизация управления водоснабжающим предприятием на основе процессного подхода //С.В. Прокопчина, д.т.н., профессор//Программные продукты и системы, Финансовый университет при Правительстве РФ, г. Москва, 2014, С. 201-204
4. Ивановский, В.Н., Сабиров, А.А., Деговцов, А.В. Системы очистки воды для нужд поддержания пластового давления и промышленной подготовки нефти// Территория «НЕФТЕГАЗ». – 2014. С. 54-59
5. Барская, И.В., Лукашевич, О.Д. Водоснабжение, канализация, строительные системы охраны водных ресурсов// Барская, И.В., Лукашевич, О.Д// ТГАСУ, Томск С. 1-7

Меренкова О.В.

Язык 1С как язык программирования

*ФГБОУ ВО «Елецкий государственный университет им. И.А. Бунина»
(Россия, Елец)*

doi: 10.18411/sr-10-06-2020-08

idsp: sciencerussia-10-06-2020-08

*Научный руководитель
Корниенко Д.В.*

Аннотация

Встроенный язык 1С программирования является популярным для разработки корпоративных информационных систем. Он достаточно гибок и позволяет разработчику подстраиваться под определенные условия. Он имеет базовые конструкции функционирования, поддерживает написание кода на двух языках. Программный модуль визуально понятен, что облегчает работу с кодом.

Ключевые слова: Язык программирования, синтаксис, тип, значение, разработка, программное обеспечение, система, модуль, процедура.

Программа «1С: Предприятие» на данный момент является лидером в сфере автоматизации бизнес-процессов, представляющая собой платформу, на базе которой работают отдельные подсистемы семейства 1С. Несмотря на то, что в момент появления программы «1С: Предприятие» на рынке бухгалтерских программ уже существовало множество других ПО для ведения бухгалтерского учета («Инфобухгалтер», «Парус», «Финансы без проблем» и другие), 1С прочно заняла свои позиции в этой нише. Главным преимуществом являлся встроенный макроязык программирования. В данной программе язык не был «жестким», то есть он позволял вносить правки, изменения и корректировки в программный код, программа имела возможность приспосабливаться к разным условиям. Самая первая версия программы

была разработана под DOS. Для этой платформы позднее было выпущено еще несколько версий. В версиях 5.0 и 2.0 ПРОФ, которые выпустили одновременно, впервые был внедрен встроенный язык формирования печатных форм и бухгалтерских проводок. Рассмотрим подробнее функции и возможности языка 1С. [1]

Язык 1С нельзя назвать полноценным языком программирования, так как среда 1С не может создавать отдельные исполняемые файлы, в отличие от других языков программирования. Разработчик не имеет прямого доступа к таблицам базы данных, используя так называемый псевдоязык, состоящий из операторов, которые можно написать на русском или английском языках.

Серьезные изменения в коде произошли с выпуском версии 8. Перенос программного кода из версии 7.7. в новую версию стал невозможен.

Встроенный язык программирования 1С: Предприятие – язык программирования, использующийся в семействе программ «1С: Предприятие». Это важная часть платформы, так как позволяет описать алгоритмы работы прикладного решения. Данный язык допускает конструкции, позволяющие создавать новые подпрограммы. В версиях 7.7 и старше возможностью интерпретации была частичной, то есть интерпретация текста программного модуля в байт-код выполнялась в момент обращения к этому модулю в процессе работ. В версиях 8.2 и выше модули компилируются. Встроенный язык 1С походит на языки Pascal, Java Script, но в то же время он является самостоятельным языком.

Встроенный язык 1С имеет несколько значимых особенностей:

- предварительная компиляция — перед исполнением модули, содержащие текст на встроенном языке, преобразуются во внутренний код;
- использование быстродействующей памяти для хранения скомпилированных модулей;
- возможность изменения типа переменной в процессе работы;
- возможность использования базовых объектов конфигурации и объектов, созданных системой при визуальном конструировании прикладного решения.

Еще одной особенностью встроенного языка является двуязычный синтаксис: написание команд может осуществляться как на русском, так и на английском языках. Но чаще всего разработчики используют для написания кода кириллицу, что связано прежде всего с труднопереводимыми терминами предметной области. В свою очередь, язык кода не подчиняется обычным правилам русского языка. Иными словами, отсутствуют склонения идентификаторов, применение окончаний и использование синонимов, при непосредственном написании кода. [3]

Большинство зарезервированных слов, имена типов значения, свойств, событий, методов имеют названия и на русском, и на английском языках, причем, оба языка можно использовать одновременно. Исключениями являются слова, которые не имеют аналогов в русском языке (например, Null). Язык не чувствителен к регистру.

Для работы прикладного решения 1С: Предприятия не обязательно прописывать все в виде программного кода. Большая часть прикладного решения возможно сконструировать визуально, используя объекты конфигурации, их свойства, взаимосвязи, формы представления. Одной из функций встроенного языка является создание новых алгоритмов обработки, отличных от типовых.

В связи с этим, модули, содержащие текст на встроенном языке, используются в заранее известных ситуациях, которые возникают в процессе работы и называются событиями. Например, ниже на рисунке 1 представлены события, связанные с функционированием объекта конфигурации Справочник.

Рис. 1. События объекта Справочник

Рассмотрим событие ПередЗаписью. Как видно из названия события, оно возникает перед записью элемента справочника в базу данных. Встроенный язык предоставляет возможность описать алгоритм, например, для корректной записи данных. Разместив этот алгоритм в нужном модуле, будет выполняться проверка правильности написания данных.

Можно сделать вывод, что встроенный язык - это скриптовый язык. Выполнение модулей (модуль – объект, содержащий программный код) происходит при возникновении определенного события в процессе функционирования прикладного решения.

Огромным плюсом в редактировании модулей является выделение цветом синтаксических конструкций и их группировка. На рисунке 2 представлено обозначение начала и конца процедуры, которое выделено красным цветом.

```

Процедура ПриУстановкеНовогоНомера (СтандартнаяОбработка, Префикс)
Префикс = Обмен.ПолучитьПрефиксНомера ();
КонiecПроцедуры

```

Рис. 2. Выделение синтаксических конструкций

Группировка синтаксических конструкций (представлена на рисунке 3) позволяет сворачивать и разворачивать их, что облегчает просмотр длинных конструкций, переносить и копировать группы целиком. Свернутый текст обозначается маркером.

```

Процедура ПриУстановкеНовогоНомера (СтандартнаяОбработка, Префикс) ...

```

Рис. 3. Группировка синтаксических конструкций

Платформа 1С позволяет работать с различными типами данных. Перейдем непосредственно к рассмотрению кода платформы 1С.

Исполняемый код на языке 1С содержится в программных модулях и предназначен для того, чтобы платформа определенным образом реагировала на действия системы или пользователя, когда визуальных средств разработки недостаточно. В программных модулях можно описать собственные методы и функции.

Разберем классический программный модуль. Чаще всего он состоит из трех основных частей:

- область объявления переменных;
- область описания процедур и функций;
- основной текст программы.

Область объявления переменных размещается в самом начале программного модуля, предшествует первому исполняемому оператору. В этом разделе находятся только операторы объявления переменных (рисунок 4).

```
//объявление переменной
Перем фамилия Экспорт; //глобальная переменная
Перем Имя, Отчество; //переменная модуля
Перем ФИО; //переменная модуля, к которой можно обращаться
//из любой процедуры модуля
```

Рис. 4. Объявление переменных

Область описания процедур или функций располагается от первого оператора Процедура или Функция до оператора, исполняемого вне тела этой процедуры или функции.

```
//область описания процедур и функций
Процедура ПроцедураЗаписиФИО()
    Перем Итог ;//переменная процедуры
    Итог="фамилия"+"Имя"+"Отчество";
КонецПроцедуры
функция функцияВозврат()
    //операторы функции
    Возврат ("фамилия"+"Имя");
Конецфункции
```

Рис. 5. Область описания процедур и функций

Область основной текст программы размещается от первого оператора вне тела процедуры или функции до конца модуля (рисунок 6). В раздел включены только исполняемые операторы. В конкретном программном модуле какая-либо из этих областей может отсутствовать.

```
//основной текст программы
фамилия="Меренкова";
Имя="Оксана";
Отчество="Владимировна".
```

Рис. 6. Область основной текст

Программные модули необходимы в том случае, когда в конфигурации необходимо описать алгоритмы функционирования, отличающиеся от стандартных. Алгоритмы выполняют в виде процедур или функций, которые будут вызываться самой системой, например, при открытии справочников, при изменении объектов и так далее. Такие ситуации предусматриваются заранее.

Программные модули воспринимаются системой как единое целое, в связи с чем все функции и процедуры выполняются в едином контексте. Модуль может выполняться либо на клиенте, либо на сервере, но некоторые программные модули могут быть скомпилированы как на стороне сервера, так и на стороне клиента.

Выше было рассмотрено объявление переменной в 1С до процедуры и функции, но это также возможно осуществить и внутри их. Переменная объявленная в начале программного модуля считается глобальной. Для обеспечения к переменной, доступа из другого модуля необходимо объявить ее экспортной (рисунок 7).

```
Перем Склад, Подразделение Экспорт;
```

Рис. 7. Экспорт переменной

Переменные во встроенном языке 1С имеют мягкую типизацию, то есть в одном и том же программном коде переменной могут присваиваться разные типы данных.

Как говорилось ранее, программный модуль - это комбинация различных операторов, выполняющих заданное действие. Между собой операторы разделяются знаком «;» (точка с запятой). Описать переменную можно явно и неявно. Явный способ описания переменной используется если переменную необходимо передать в функцию. Также в названии переменных нельзя употреблять зарезервированные слова, которые указаны в Синтаксис-помощнике, например, такие как Цикл, КонецПроцедуры и другие. [4]

Рассмотрим несколько основных операторов.

Оператор присваивания используется для инициализации переменных, элементов массивов или реквизитов объекта некоторыми значениями, а также для предопределения их значений при последующих упоминаниях. Источниками значений могут быть константа, имя переменной, элемент массива. На рисунке 6 приведен пример оператора присваивания.

Условный оператор или оператор «Если» применяется при наличии какого-либо условия. После условия прописывается слово «Тогда» и выполняются последующие операторы. Затем можно задать ключевое слово «Иначе» и серию других операторов. В том случае, когда условий несколько используют «ИначеЕсли». Завершается этот оператор «КонецЕсли», после чего следует точка с запятой (рисунок 8). [5]

```
Если ВыборкаРезультата.НайтиСледующий(СтруктураНомер) Тогда
Если ВыборкаРезультата.Норма = 0 Тогда
Сообщение = Новый СообщениеПользователю;
Сообщение.Текст = "Вид расчета: Оклад - Нет рабочих дней в заданном периоде";
Сообщение.Сообщить();
ЗаписьРегистра.Результат = 0;
Иначе
// Рассчитать оклад по фактическому периоду и исходным данным
ЗаписьРегистра.Результат = (ЗаписьРегистра.ИсходныеДанные/ВыборкаРезультата.Норма)
```

Рис. 8. Пример условного оператора

Циклические операторы используют для задания цикла по счетчику или условию. Цикл завершается командой «КонецЦикла». Для цикла по счетчику задается фиксированное число повторений. Условный цикл выполняется до тех пор, пока заданное условие цикла истинно. [3]

```
Для Каждого ЗаписьРегистра Из НаборЗаписейРегистра Цикл
СтруктураНомер = Новый Структура("НомерСтроки");
СтруктураНомер.НомерСтроки = ЗаписьРегистра.НомерСтроки;
ВыборкаРезультата.Сбросить();
```

Рис. 9. Использование условного цикла

Несмотря на то, что встроенный язык 1С предоставляет множество возможностей для решения разнообразных задач, он также имеет и некоторые недостатки. Он обладает рядом ограничений, которые не характерны для языков общего назначения. Например, на этом языке нельзя реализовать интерфейс, не поддерживаемый платформой. С другой стороны, необходимое для решения прикладных задач успешно реализовано в объектах платформы, которые доступны из встроенного языка. Смысловая нагрузка объектов определена на уровне платформы, благодаря чему обеспечивается их поддержка: расчет итогов, целостность данных, представление в интерфейсе и другое. С течением времени встроенный язык 1С почти

не меняется, но происходит расширение встроенных объектов и дополнительных функций.

1. Борькина, А.А. История возникновения «1С: Предприятие» / А.А. Борькина, А.К. Конищева // Сборник материалов VIII всероссийской, научно-практической конференции молодых ученых с международным участием "Россия молодая". – Кемерово: Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2016.
2. Корниенко, Д.В. Реализация механизма сложных периодических расчетов при разработке прикладного решения на базе 1С: Предприятие 8: монография / Д.В. Корниенко, В.В. Корниенко. – Елец: Елецкий гос. ун-т им. И. А. Бунина, 2016. – 61 с.
3. Лебедева, Т.Н., Носова Л.С. Формализация данных в языке программирования 1С // Вестник астраханского государственного технического университета. – 2015. – №3. – С. 113-121.
4. Операторы и синтаксические конструкции [Электронный ресурс]: сайт. – URL: <https://master1c8.ru/platforma-1s-predpriyatye-8/rukovodstvo-razrabottchika/glava4-vstroenny-yazk/4-6-operator-i-sintaksicheskie-konstruktsii-tchasty-2/> (дата обращения: 30.04.2020)
5. Переменные и операторы [Электронный ресурс]: сайт. – URL: <https://курсы-по-1с.рф/articles/переменные-и-операторы/> (дата обращения: 01.05.2020)
6. Требования к написанию и оформлению программного кода 1С [Электронный ресурс]: сайт. – URL: <https://programmist1s.ru/trebovaniya-k-programmnomu-kodu-1s/> (дата обращения: 20. 04. 2020).

Наумов И.И., Тарасюк М.А., Моторин Д.Е. Бондаренко А.А. Владимиров В.В.
Обзор роботизированных платформ для дезинфекции помещений и общественных мест

*Донской государственный технический университет в г. Шахты
 (Россия, Шахты)*

doi: 10.18411/sr-10-06-2020-09

idsp: sciencerrussia-10-06-2020-09

Аннотация

В текущей эпидемической ситуации, когда пандемией коронавируса (COVID-19) охвачены все страны мира, появилось множество компаний, которые занимаются разработкой эффективных способов дезинфекции.

В данной статье производится анализ технологически ориентированные решения, которые обеспечивают общественную безопасность и здоровье для миллионов людей во всем мире. Разработки UVD Robots, Xenex Disinfection Services, Nevoa, Neolix и XAG - одни из лучших, на которые стоит обратить внимание.

Ключевые слова: Робот дезинфектор, Обзор, UVD Robots, Xenex Disinfection Services, Nevoa, Neolix, XAG.

В текущей эпидемической ситуации, когда пандемией коронавируса (COVID-19) охвачены все страны мира, появилось множество компаний, которые занимаются разработкой эффективных способов дезинфекции.

Для предотвращения распространения коронавирусной инфекции в общественных местах, необходимо проводить регулярную дезинфекцию поверхностей. Процесс ручной очистки опасен, учитывая возможность заражения. Перспективными в решении данной проблемы становятся автономные дезинфекционные роботы.

В данной статье мы проанализировали технологически ориентированные решения, которые обеспечивают общественную безопасность и здоровье для миллионов людей во всем мире. Разработки UVD Robots, Xenex Disinfection Services, Nevoa, Neolix и XAG - одни из лучших, на которые стоит обратить внимание.

UVD Robots - робот для дезинфекции ультрафиолетом

Ультрафиолетовое излучение клинически доказано как эффективное дезинфицирующее средство. Он убивает болезни, вирусы, бактерии и другие виды