ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.415.2.052.03

Р. Н. Воронов, И. Е. Воронова

АНАЛИЗ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ ДЛЯ СОЗДАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, ПРИМЕНЯЮЩЕГОСЯ В ХИМИЧЕСКОЙ ПРОМЫШЛЕННОСТИ

На предприятиях химической промышленности, как сложных технических системах, необходимо соблюдение требований безопасности. Эти требования распространяются на грамотное управление технологическим оборудованием, чтобы не допустить сбоев. Безотказность работы оборудования зависит от многих факторов, одним из которых является качество программного обеспечения (далее ПО). Качество ПО напрямую зависит от грамотности программистов и от языков программирования, примененных при его создании.

Цель статьи — проанализировать существующие языки программирования и выделить среди них наиболее оптимальные для создания ПО, управляющего технологическими процессами на предприятиях химической промышленности.

Языки программирования подразделяемя на несколько групп по различным признакам:

- по семантике: структурные (императивные), объектно-ориентированные, функциональные или мультипарадигменные;
- по типизации данных: с динамической или статической типизацией. Языки со статической типизацией, в свою очередь, делятся на типобезопасные и не типобезопасные.

Объектно-ориентированные языки программирования делятся на два типа: чисто-объектно-ориентированные (Smalltalk, Eiffel) и структурные, имеющие объектную надстройку (C++, Objective-C, Oberon, Ada).

Для программирования программно-аппаратных комплексов чаще всего используются статически-типизированные языки, в связи с тем, что большая часть ошибок в программе обнаружится на этапе компиляции. В типобезопасных языках не допускается автоматическое приведение типов данных. К таким языкам относят, например, Ada, Oberon, Modula-2/3, Haskell и т. п. Данные языки разрабатывались с упором на безопасность, как системные языки высокого уровня.

Обычно в качестве стандартных языков для разработки управляющего ПО служат процедурные или гибридные языки программирования. Наиболее часто используется язык C, стимулирующий написание потенциально-опасных программ, поскольку принуждает программиста запоминать множество ссылок на участки памяти, которые необходимо выделять и освобождать,. В отличие от C и C++, современные языки программирования Pascal-семейства такие как Oberon/Oberon-2, Ada позволяют обойтись без явного выделения памяти, имея возможность передавать аргументы в процедуры по ссылке. Благодаря этому, например, язык Ada часто используется для написания бортового ПО для самолётов и поездов. Языки Oberon-2 и Modula-2 — для написания бортового ПО орбитальных спутников и самолётов. Язык Modula-3 разрабатывался для управления производственными процессами. На данный момент он наименее распространён.

Другим потенциально-опасным местом в программе является наличие глобальных переменных - чем меньше их в программе, тем меньше труднообнаруживаемых ошибок.

В отличие от процедурных языков, чистые функциональные языки вообще не позволяют создавать ни глобальных, ни локальных переменных, и не имеют средств управления памятью, доступных программисту. Все манипуляции по выделению и освобождению памяти производятся автоматически, незаметно для пользователя. Отсутствие переменных компенсируется поддержкой хвостовой рекурсии. Таким образом, программа, написанная на функциональном языке, выглядит как математическая нотация какого-либо процесса. Также, у некоторых функциональных языков (Standard ML, Ocaml, Haskell) имеется система автоматического вывода типов данных Хиндли-Милнера [1], что заметно сокращает текст программы и позволяет обнаружить множество ошибок на стадии компиляции программы. Отсутствие глобальных переменных облегчает поиск ошибок на стадии выполнения программы, так как ошибки в этом случае могут появиться только в какой-либо конкретной функции и никак не воздействуют на работу программы вцелом. Язык Haskell обладает следующими возможностями:

- наиболее распространённый компилятор языка Haskell (Glasgow Haskell Compiler, GHC) полностью поддерживает параллельное программирование, выгодно использует ресурсы современных многоядерных и многопроцессорных систем;
- поддерживает параллельное управление памятью (каждый процесс самостоятельно освобождает ресурсы, в которых больше не нуждается), что позволяет использовать его в системах реального

Таблица 1. Анализ наиболее распространённых языков программирования в управлении производственными процессами

Язык	Семантика	Типизация	Управление памятью	Поддержка параллельного программирования	Признак безопасности
C	Структурный	Статическая (допускается автоматическое приведение типов данных)	Ручное	Только при наличии специальной библиотеки	Нет
C++	Гибридный (структурный с объектно-ориентированной надстройкой)		Ручное	Только при наличии специальной библиотеки	Нет
Objective-C			Ручное или автоматическое	Только при наличии специальной библиотеки	Нет
Modula-2	Структурный, модульный	Статическая (автоматическое приведение типов не допускается)	Ручное	Только при наличии специальной библиотеки	Да
Modula-3	Гибридный (структурный с возможностью объектно-ориентированного программирования), модульный		Автоматическое	Да (имеется поддержка в стандартной библиотеке)	Да
Oberon/ Oberon-2			Автоматическое	Только при наличии специальной библиотеки	Да
Ada			Ручное, полуавтоматическое	Да (имеется поддержка в языке)	Да
Haskell	Функциональный	Статическая (автоприведение типов не допускается). Развитая системой типов	Автоматическое	Да (имеется поддержка в стандартной библиотеке)	Да

времени;

- поддерживает отложенные вычисления, что позволяет без затрат памяти создавать бесконечные структуры данных;
- обладает наиболее развитой, в сравнении с другими языками, системой типов.

Исходя из сравнительной таблицы, можно сделать вывод, что наиболее оптимальными языками программирования для управления производственными процессами на химических предприятиях являются языки, обладающие следующими свойствами типобезопасность, поддержки параллельного программирования, автоматического управление памятью (сбор мусора).

Данным требованиям удовлетворяют Haskell; Oberon/Oberon-2, Modula-3, Ada, Modula-2 (управление памятью ручное).

Оптимален из перечисленных языков Haskell. Благодаря совершенной системе типов, модели вывода типов, поддержке параллельных вычислений, а также, автоматическому управлению памятью и краткости, этот язык наиболее подходит для создания программных комплексов, управляющих технологическими процессами. В данный момент некоторые зарубежные предприятия и организации используют Haskell в своих разработках. Так, американская компания Amgen, работающая в сфере биотехнологий, использует Haskell для быстрой реализации математических моделей; международный банк ABN AMRO в Амстердаме использует ПО, написанное на языке Haskell, которое отличается высокой надёжностью.

СПИСОК ЛИТЕРАТУРЫ

1. Hindley Milner Type Inference [Электронный ресурс]. – Режим доступа: http://c2.com/cgi/wiki?HindleyMilnerTypeInference, свободный.

Авторы статьи:

Воронов Роман Николаевич - ассистент каф. процессов, машин и аппаратов химических производств. E-mail: vrn.pmahp@kuzstu.ru

Воронова Ирина Евгеньевна - ассистент каф. процессов, машин и аппаратов химических производств. E-mail: vie.pmahp@kuzstu.ru