

О КРИТЕРИЯХ ВЫБОРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Н.А. Атрощенко

Кафедра информационных технологий и высшей математики, Минский университет управления

Минск, Республика Беларусь

E-mail: {natasha}@atroshenko.by

Для начинающих и перманентно совершенствующих свои знания опытных программистов, для компаний, связанных с IT-разработками немаловажным является рейтинг языков программирования и прогноз на дальнейший их вес в IT-технологиях в условиях постоянно меняющейся конъюнктуры рынка.

ВВЕДЕНИЕ

Основными критериями выбора языка программирования для разработчика при перспективе создать проект в сотни тысяч строк кода принципиальными являются два момента: «удобство» написания кода (поддерживаемые парадигмы программирования, возможности ООП, подключаемых библиотек и др.) и корректность его работы, заключающаяся в возможности избежать несанкционированных ошибок: язык программирования должен быть полностью лишён «дыр», приводящих к аварийному завершению приложения с потерей данных с аварийным завершением работы ОС. Для сравнения можно взять языки промышленного программирования: Java и C++.

JAVA или C++?

Критические ошибки могут возникнуть при: обращении к чему-либо несуществующему. Это может быть ещё не созданный объект или несуществующий элемент массива. В Java гарантируется, что при обращении к несуществующему элементу при индексации массива генерируется соответствующее исключение, которое можно легко отловить. В языке Си не только нет проверки при индексации массива, но и само существование их невозможно, так как для обработки передаётся в процедуры только ссылка на начало массива, а не его длина. C++ унаследовал это от Си, с той лишь разницей, что можно обойти это несовершенство, создав в качестве массива `StaticArray<int, 10> myarray;` вместо `int myarray[1]`. Частой ошибкой также является обращение через указатель к несуществующему объекту, и, если в Java нет нужды программисту заниматься удалением объекта (в Java, к примеру, этим занимается JVM), то в C++ за удаление неиспользованных объектов отвечает только сам программист, и на один и тот же объект можно создать, к примеру, два указателя, причём через один указатель этот объект удалить, а через другой ещё работать с тем местом, где он находился. Дополнительные усилия со стороны программиста, однако, могут пресечь проблему удаления объекта, в котором находится исполнение путём коррекции потока управления, - не находится ли

он в удаляемом объекте, а после удаления объекта навсегда выйти из процедуры. Другой «неприятной особенностью» в языке может стать неинициализированная переменная. В C++ чрезвычайно важно проинициализировать все поля класса при их создании, иначе при запуске программы вы рискуете получить невоспроизводимые ошибки [1]. Даже одна неинициализированная переменная в конструкторе может привести к критической ошибке в самом начале запуска приложения, ведь на начальной стадии она может инициализироваться любым значением, и это может быть наименьшее или наибольшее возможное значение в данном типовом диапазоне. От такой ситуации застрахована Java, так как все поля зануляются при их создании, например, переменная типа `int` изначально равна нулю, а `boolean=false`. Обойти эту проблему в C++ можно на стадии операции `new` путём её переопределения. В Java, в отличие от C++, является обязательной обработка исключительных ситуаций, что позволяет избежать не только двусмысленностей, но и критических ошибок, приводящих к аварийному завершению приложения типа «файл не найден». Кроме того, умело написанный Java-код может сам обрабатывать все ошибки во время исполнения, не говоря уже о таких мощных Java-инструментах, как Maven, Apache Ant и JUnit. В C++ гарантируется уничтожение всех локальных объектов при выходе из процедуры (через `return` или через исключение) при вызове деструкторов, если в конструкторе прописать открытие ресурса, а в деструкторе – закрытие [2]. Так называемый циклический импорт, когда два класса взаимозависимы (к примеру, один класс вызывает методы второго, а первый – методы первого: в этом случае встаёт вопрос, какой из них кого должен импортировать) не является проблемой для Java: классы увидят друг друга. В C++ запрещён циклический импорт, и какой-то класс обязательно нужно объявлять первым, но есть обходные пути путём наследования от интерфейсного класса с использованием виртуальных функций. Использование Java в ракурсе многопоточного программирования основано на высокоорганизованном элегантном решении многофункциональной синхронизации, с помощью которой можно создавать высокоэф-

фективные интерактивные системы, в то время как поддержка многопоточности в C++ сопряжена со значительными трудностями. Java – это не только язык, а целая платформа, которая постоянно, без материальных (freeware) и особых технических затрат постоянно расширяется за счет создания новых стандартных классов и библиотек. Необходимо признать, что в C++ стандартная библиотека шаблонов (STL) и набор шаблонных классов (ATL) в свете расширяемости и совершенствования довольно консервативны. К тому же, библиотеки, интегрированные программные оболочки для Java-программ требуют меньше затрат, чем аналогичные разработки для C++. Быстро получившая своё распространение система непрерывной интеграции Jenkins в Java дала новые преимущества разработки крупных приложений, это: настраиваемое непрерывное тестирование, прогон тестов, деплой кода на тестовый сервер и проверку code style, хранение собранных deb-пакетов, отчёты о тестах, управление сборкой проектов в зависимости от событий или установленного времени, поддержка инструментариев для работы с разными системами контроля версий, включая CVS, Subversion, Mercurial и др. В рейтинге языков программирования голландской компании TIOBE Software BV [3] (автора расчета рейтинга популярности языков программирования с 2001 года), при сравнении 150 языков программирования Java уже много лет является лидером (в 2014 году впервые на первое место вышел C). Исследования составляются на основе данных самых популярных в мире онлайн-сервисов, (Google, MSN, Yahoo!, Wikipedia) по целому списку показателей, среди которых основным является количество поисковых запросов в сети Интернет, а значит, заинтересованность разработчиков и популярность цитирования. Лидирующее место в других рейтингах также достаточно прочно занимает C++ [4], где-то опережая, где-то уступая Java [5] возможно, потому, что система пользователя Java – приложений должна обладать достаточно мощными вычислительными ресурсами при использовании приложений со сложными пользовательскими интерфейсами, а также потому, что за кро-

сплатформенностью Java кроется не очень высокая производительность, поэтому считается, что вычислительные процессы лучше писать на C++. В рейтинге языков, по мнению известной компании Jobs Tractor [6], занимающейся отслеживанием вакансий для IT-разработчиков, составляющей ежемесячный рейтинг ведущих языков программирования, Java отводится второе место (август 2014г.) [7], такие же показатели у RedMonk [8], составляющей свой рейтинг популярности языков программирования раз в квартал, а C++ - девятое. Тем не менее, C++ по-прежнему наиболее предпочтительный язык для реализации математических моделей, написания кодов для компьютерных игр, сложных вычислительных промышленных разработок (многие операционные системы написаны на C++).

В любом случае, при выборе языка для создания приложения (проекта) нужно руководствоваться целесообразностью, расширяемостью, масштабируемостью, ресурсоёмкостью, интеграцией с другими языками и платформами и другими критериями.

1. Николас А. Солтер, Скотт Дж. Клеппер C++ для профессионалов /Николас А. Солтер, Скотт Дж. Клеппер // Компьютерное издательство – Диалектика, –2005. –912 с
2. А. Хортон – Visual C++ 2005 базовый курс // Компьютерное издательство Диалектика, 2007. – 1152 с
3. TIOBE Programming Community Index [Electronic resource] – Mode of access: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> – Date of access: 10.08.2014.
4. Рейтинг языков программирования [Электронный ресурс] / Режим доступа: <http://dou.ua/lenta/articles/language-rating-jan-2014/> – Дата доступа: 10.08.2014.
5. Анализ популярности языков программирования [Электронный ресурс] /Режим доступа: <http://www.opennet.ru/opennews/art.shtml?num=37408/> – Дата доступа: 10.08.2014.
6. Jobs Tractor [Electronic resource] – Mode of access: <http://jobstractor.com/> – Date of access: 10.08.2014.
7. Jobs Tractor language trend [Electronic resource] – Mode of access: <http://jobstractor.com/monthly-stats/> – Date of access: 10.08.2014.
8. redmonk.com [Electronic resource] – Mode of access: <http://redmonk.com/> – Date of access: 10.08.2014.