

JAVA ДОСТОИНСТВА И НЕДОСТАТКИ JAVAPROSANDCONS

Гаврюшов Егор Дмитриевич

Научный руководитель: Куценко А.А.

преподаватель

НАН ЧОУ ВО «Академия маркетинга и социально-информационных технологий»

Академический колледж, г. Краснодар

Аннотация. Описать в общих чертах основные особенности Java как язык, так и технологию.

Ключевые слова: Язык программирования, безопасность

С одной стороны, Java — объектно-ориентированный язык программирования, с другой — технология, разработанная компанией Sun Microsystems. Как язык, так и технология официально существует с 23 мая 1995 года.

Код компилируется в байт-код и работает на специальной Java-машине. Это сглаживает как недостатки возникающие при чистой компиляции так и недостатки возникающие при чистой интерпретации.

При чистой компиляции на основе исходного текста программы составляется набор машинных команд, соответствующих алгоритму программного кода. То есть, после компиляции текста в набор команд, мы получаем как результат, некую программу, для запуска которой не требуется ничего кроме той операционной системы, для которой была написана программа. Так как, при компиляции переход осуществляется в набор команд, понятных конкретной машине (компьютеру) и конкретной среде (ОС), то и оптимизация этого кода исходит из параметров и конфигураций этой самой машины. Поэтому такая программа будет работать максимально стабильно и быстро, будет затрачивать минимум ресурсов.

Всем известно, принципы Java – простота языка, безопасность и переносимость кода.

Одно из крупнейших преимуществ языка заключается в богатой библиотеке. С одной стороны громадность библиотеки является одной из преград к изучению языка, и как следствие высокий порог вхождения. Но благодаря высокому порогу вхождения популярность языка понизилась, в то время как востребованность языка осталась прежней/увеличилась. Это дает нам такие плоды как высокая зарплата Java-программистов, относительно малая конкуренция, возможность бесплатного обучения Java заинтересованными фирмами и последующие сертификация и трудоустройство, силами этих же фирм.

Богатая библиотека сводит тривиальную часть работы программиста к поиску готового решения. Что освобождает программиста от рутины и предоставляет ему возможность более серьезно вложиться в творческий аспект, сэкономить время и силы.

Одна из проблем Java. И причины здесь можно выделить две: жесткая политика Объектно-Ориентированного Подхода (далее ООП), отсутствие функций первого класса (first-classfunction).

Политика ООП введена в Java с целью избавления от некоторых опасностей, возникающих при создании крупного проекта. В Java же невозможно создавать функции, не являющиеся методами какого-либо класса. К примеру, в C++ такого запрета нет. Из-за этого в C++, при проектировании больших проектов, возникает целый ряд опасностей. Использование функций "не методов" приводит к тому что:

- может возникнуть конфликт при именовании функций, когда имя функции с таким же набором параметров и такого же типа данных уже имеется.
- проект в целом становится более неповоротлив
- проект становится труден для осмысления, не только новопришедшему программисту, но и тому кто давно в проекте

Запрет на функциональный подход избавляет от подобных опасностей и учит хорошему тону программирования. В итоге, на одной чаше весов многословность с громоздкостью, а на другой надежность работы проекта и безопасность.

Функции первого порядка по-сути очень сходны с классами первого порядка. Это функции, которые можно передавать как параметры. В C++ это решается при помощи поддержки указателей на функции. В Java все же есть способы передачи ссылки на функцию, но обычно там где в C++ используется указатель на функцию, в Java принято использовать Интерфейсы. Это более многословная технология, но при этом она избавляет от опасностей при использовании указателей.

Java, несмотря на различные способы оптимизации, всё же довольно ресурсоёмка и медлительна. Причины в следующем:

- автосборка мусора
- компиляция "налету" (Just In Time compilation)
- отказ от таких опасных механизмов как: арифметика указателей, неявное преобразование типов с потерей точности, функции первого класса

Но это всё делает язык более платформонезависимым и безопасным. Автосборка мусора с одной стороны освобождает программиста от заботы освобождения памяти, с другой стороны известна проблема когда автосборка мусора не срабатывала и происходила серьёзная утечка памяти.

Программы, написанные на языке Java, после однократной трансляции в байт-код могут быть исполнены на любой платформе, для которой реализована виртуальная Java-машина.

Наиболее эффективно возможности реального компьютера может использовать только

программа, написанная с использованием «родного» машинного кода.

Список литературы:

1. Берд, Барри Программирование на Java для чайников / Барри Берд. - М.: Диалектика, 2013. - 384 с.
2. Васильев, А. Н. Самоучитель Java с примерами и программами / А.Н. Васильев. - М.: Наука и техника, 2016. - 368 с.
3. Герберт, ШилдтJava 8. Руководство для начинающих / Шилдт Герберт. - М.: Диалектика / Вильямс, 2015. - 899 с.
4. Герберт, ШилдтJava. Руководство для начинающих / Шилдт Герберт. - М.: Диалектика / Вильямс, 2014. - 104 с.