



# Урок № 1

## Курс: «Разработка программного обеспечения на Java»

Тема: Введение в язык программирования “Java”

### План

1. Вступление
  1. История и этапы развития языка “Java”
  2. Сравнительный анализ языка ”Java” с другими языками программирования
  3. Что такое виртуальная машина?
  4. Что такое байт-код?
  5. Что такое Java
2. Алгоритм
  1. Понятие алгоритма
  2. Примеры использования алгоритмов в реальной жизни
  3. Типы алгоритмов. Линейный, разветвлённый, циклический

## 1. Вступление

### 1.1 История и этапы развития языка “Java”

Java - один из самых популярных языков программирования. По мнению Герберта Шилдта, основная причина успеха Java - его быстрое развитие от версии к версии и "способность" быстро подстраиваться под современные нужды программистов.

#### **История создания. Начало**

Всё началось в 1991 году, когда группа инженеров из компании Sun под руководством Патрика Нотона и Джеймса Гослинга занялась разработкой небольшого языка, который можно было бы использовать для программирования бытовых устройств, например, контроллеров для переключения каналов кабельного телевидения.

Поскольку такие устройства не потребляют много энергии и не имеют больших микросхем памяти, то язык должен был быть маленьким и генерировать очень компактные программы. Кроме того, поскольку разные производители имели разные архитектуры процессоров, было важно обеспечить "выполняемость" программ, написанных на данном языке на разных машинах. Проект получил кодовое название "Green".

#### **Идея от Pascal**

Стремясь изобрести небольшой, компактный и машиннезависимый код, разработчики возродили модель языка, использованную при реализации первых версий языка Pascal. Никлаус Вирт, создатель языка Pascal, в свое время разработал машиннезависимый язык, генерирующий промежуточный код для некоей гипотетической машины.

Этот язык стал коммерческим продуктом под названием UCSD Pascal. (Такие гипотетические машины часто называются виртуальными — например, виртуальная машина языка Java, или JVM.)

Промежуточный код можно выполнять на любой машине, имеющей соответствующий интерпретатор. Инженеры, работавшие над проектом "Green", также использовали виртуальную машину, что решило их основную проблему.

#### **Идея от C++**

Несмотря на модель языка Pascal, сотрудники Sun имели опыт работы с UNIX и знали, что такое "промышленная разработка", потому в основу синтаксиса Java они положили именно C++, а не Pascal. В частности, они сделали язык объектно-, а не процедурно-ориентированным.

## **Происхождения названия**

Сначала Джеймс Гослинг решил назвать его "Oak" (рус. «Дуб»). Однако потом сотрудники компании Sun узнали, что слово Oak уже используется в качестве имени ранее созданного языка программирования, и в 1995 изменили название на Java.

Язык назван в честь марки кофе Java, которая, в свою очередь, получила наименование одноимённого острова (Ява), поэтому на официальной эмблеме языка изображена чашка с парящим кофе:



Существует и другая версия происхождения названия языка, связанная с аллюзией на кофе-машину как пример бытового устройства, для программирования которого изначально язык создавался.

## **Становление языка**

Одним из главных факторов роста популярности Java в первых версиях (по мнению Г. Шилдта) был взрывообразный рост популярности сети Интернет - и, как следствие, появления потребности создания платформонезависимых приложений.

В 1993 разработчики Java (тогда ещё Oak) понимают, что Интернет, как и микроволновки (и кофемашины) требует кроссплатформенных решений. Именно Интернет обеспечит в дальнейшем успех Java в мире программистов и, как следствие, - пользователей.

## **Влияние на интернет**

Java предлагает переложить ряд "обязанностей" с сервера на клиент с помощью использования апплетов, которые выполняются во внутренней среде Java и, таким образом, контролируется доступ к иным данным на машине клиента (в целях обеспечения безопасности).

Затем появляются сервлеты (сервлет является классом Java, реализация которого расширяет функциональные возможности сервера. Взаимодействует с клиентами посредством принципа запрос-ответ), таким образом, java становится еще более универсальным языком.

## **Мобильная разработка**

Наступает эра распространения мобильных устройств. Разработчики языка и здесь идут в ногу с прогрессом и ещё до массовых продаж мобильных устройств с полноценной ОС, Java начинает распространяться по карманом пользователей -

позволяя играть в игры и выходить в интернет с помощью упрощённых браузеров. Эта тенденция только наращивает обороты с выходом смартфонов под операционной системой Android. Теперь большинство людей ежедневно пользуются миллионами приложений, написанных на java с использованием Android SDK.

## **1.2 Сравнительный анализ языка "Java" с другими языками программирования**

Язык Java во многом похож на язык C или C++, но в действительности это только внешнее сходство. Когда вы впервые взглянете на Java-код, вы увидите, что базовый синтаксис похож на C или C++. Но на этом сходство и заканчивается. Язык Java ни в коем случае не является прямым потомком C или следующим поколением C++. Если вы сравните свойства этих языков, вы увидите, что у Java в действительности больше общего с динамическими языками высшей степени, такими как Smalltalk и Lisp. В действительности реализация Java настолько далека от реализации C++, насколько вы можете себе представить.

Если вы знакомы с современной панорамой языков, вы заметите, что популярный язык C# отсутствует в этом сравнении. C# в целом — это ответ Microsoft на Java, имеющий много преимуществ при поверхностном рассмотрении. При общем дизайне и подходе (например, использование виртуальной машины, байт-код, «песочница» и др.) платформы по существу не отличаются в показателях скорости или характеристиках безопасности. C# теоретически является таким же портативным, как Java, но сегодня он поддерживается на гораздо меньшем количестве платформ. Как и Java, C# много позаимствовал из синтаксиса C, но в действительности является близким родственником динамических языков. Многие разработчики Java считают, что понять C# достаточно легко, и наоборот. Большая часть времени, которое тратится при переходе с одного языка на другой, уходит на изучение стандартной библиотеки.

Поверхностное сходство языков, однако, ничего не стоит. Язык Java берет многое от синтаксиса языка C и C++, так что вы увидите краткие языковые конструкции, включая изобилие фигурных скобок и точек с запятой. Язык Java подписывается под философией C, заключающейся в том, что хороший язык должен быть компактным, другими словами, он должен быть достаточно легковесным и регулярным, так что программист сможет сразу запомнить все его возможности. Как и в случае с расширяемостью библиотек языка C, пакеты с классами языка Java могут добавляться в центральные компоненты языка для расширения его словаря.

Язык С успешен, поскольку предоставляет программную среду со значительным содержанием свойств, с высокой производительностью и приемлемой долей портативности. Язык Java также старается балансировать между функциональностью, скоростью и портативностью, но совсем другим образом. Язык С выдает функциональность за портативность, язык Java изначально выдавал скорость за портативность. Язык Java также решает проблемы безопасности, которые язык С не решает (хотя в современных системах многие из этих проблем сейчас решаются операционной системой и аппаратным обеспечением).

В самом начале, еще до динамической и адаптивной компиляции, язык Java был медленнее, чем статически компилируемые языки, и казалось, что он никогда не достигнет достойного уровня. Но, разработчики языка проделали большую работу, и сейчас производительность Java сравнима с С или С++ в эквивалентных задачах, и эта критика в общем сошла на нет. Движок видеоигры с открытым кодом Quake2, созданной ID Software, был переписан на Java. Если язык Java является достаточно быстрым для военной игры от первого лица, то он определенно достаточно быстр для бизнес-приложений.

### **1.3 Что такое виртуальная машина?**

Виртуальная машина Java (JVM) — это часть платформы Java, которая даёт нам возможность разрабатывать и запускать приложения в различных операционных системах и на разных аппаратных платформах, то есть реализует кроссплатформенность.

Большинство языков программирования при компиляции образуют нативный для текущей платформы код, который, затем, непосредственно выполняется центральным процессором компьютера. Такой подход являлся наиболее очевидным и простым, он существовал со времен изобретения первых компиляторов. Всё бы хорошо, вот только взять и поделить свою программу на С++ с другим пользователем, у которого другая операционная система, или другая аппаратная платформа, просто так не получится.

В Java эта проблема решена кардинальным образом. Вместо того, чтобы заточивать приложение под большое количество разных процессоров и ОС, разработчики платформы Java написали приложение, которое являлось виртуальным процессором и было промежуточным звеном между физическим процессором и исполняемым кодом. Байт-код Java выполняется именно виртуальной машиной Java. Java Virtual Machine предоставляет приложению на Java доступ к памяти, файловой системе и всем функциям операционной системы.

Благодаря политики открытого кода, существует несколько реализаций реализаций JVM. Самые популярные из существующих виртуальных машин Java:

HotSpot — основная Java-машина, которая входит в официальные Java Runtime Environment и JDK, на данный момент разрабатывается Oracle.

Apache Harmony — свободная реализация виртуальной машины Java от Apache, доступная под их же лицензией.

CASAO — виртуальная Java-машина, разработанная в Венском техническом университете.

Dalvik — регистровая реализация Java-машины, является основной частью платформы Android.

## **1.4 Что такое байт-код?**

Байт-код Java — набор инструкций, исполняемых виртуальной машиной Java. Каждый код операции байт-кода — один байт.

У одного байта существует 256 возможных значений, поэтому всего 256 возможных кодов операций в байт-коде. Некоторые коды не используются и являются зарезервированы языком.

Для программирования на языке Java или других JVM-совместимых языках, знание особенностей байт-кода не обязательно. Однако, как следует из публикации в журнале IBM developerWorks, «понимание байт-кода и понимание механизмов его генерации компилятором Java помогает Java-программисту так же, как и знание языка ассемблера помогает программисту, пишущему на Си или C++».

## **1.6 Что такое Java?**

# **2. Алгоритм**

## **2.1 Понятие алгоритма**

Алгоритм — набор инструкций, описывающих порядок действий, приводящих к однозначному решению поставленной задачи (за конечное число действий).

Запись алгоритма на формальном языке называется программой. Иногда само понятие алгоритма отождествляется с его записью, так что слова «алгоритм» и «программа» — почти синонимы. Небольшое различие заключается в том, что под алгоритмом, как правило, понимают основную идею его построения. Программа же всегда связана с записью алгоритма на конкретном формальном языке. Разницу между программой и алгоритмом еще можно пояснить следующим образом. Алгоритм — это метод, схема решения какой-то задачи. А программа — это конкретная реализация алгоритма, которая может быть скомпилирована и выполнена на компьютере. Алгоритм, в свою очередь, является реализацией идеи решения. Это можно проиллюстрировать следующей схемой:

Идея решения → Алгоритм → Программа

Стрелка означает переход к следующему этапу решения задачи с повышением уровня подробности описания метода решения.

Алгоритмы обладают свойством детерминированности (определённости): каждый шаг и переход от шага к шагу должны быть точно определены так, чтобы его мог выполнить любой другой человек или механическое устройство. Кроме детерминированности, алгоритмы также должны обладать свойством конечности и массовости:

*Конечность* - алгоритм всегда должен заканчиваться за конечное число шагов, но это число не ограничено сверху.

*Массовость* - алгоритм применяется к некоторому классу входных данных (чисел, пар чисел, набору букв и тому подобному).

## 2.2 Примеры использования алгоритмов в реальной жизни

Человек постоянно сталкивается с понятием алгоритм в различных сферах своей деятельности (кулинарные книги, инструкции по использованию различных приборов, правила решения математических задач и т.д.). Обычно, мы, выполняем привычные действия, не задумываясь, механически. Например, вы хорошо знаете, как открывать ключом дверь. Однако, чтобы научить этому

малыша, придется четко разъяснить и сами эти действия и порядок их выполнения:

1. Достать ключ из кармана.
2. Вставить ключ в замочную скважину.
3. Повернуть ключ два раза против часовой стрелки.
4. Вынуть ключ.

Если вы внимательно оглянитесь вокруг, то обнаружите множество алгоритмов, которые мы с вами постоянно выполняем. Мир алгоритмов очень разнообразен.

### **2.3 Типы алгоритмов. Линеиный, разветвлённый, циклический**

Несмотря на большое разнообразие алгоритмов (в зависимости от цели, начальных условий задачи, путей решения), удастся выделить общие свойства алгоритмов и классифицировать их.

*Линейный алгоритм* — набор команд, выполняемых последовательно во времени друг за другом.

*Разветвляющийся алгоритм* — алгоритм, содержащий хотя бы одно условие, в результате проверки которого может осуществляться разделение на несколько параллельных ветвей алгоритма.

*Циклический алгоритм* — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторого условия.