**Интерфейс командной строки**

Интерфейс командной строки - это набор команд, которые операционная система (непосредственно или через дополнительное ПО) предоставляет для выполнения административных или пользовательских задач.

Команды можно разделить на внутренние команды ОС и внешние

* Внутренние команды – это часть ОС, для выполнения которых не нужно устанавливать никакого дополнительного ПО. Например: “ipconfig –all”, “path”, “cd”, “route print”;
* Внешние команды – это команды предоставляемые любым пользовательским ПО. Например: “javac HelloWorld.java”

Изначально интерфейс командной строки был слабо развит в ОС Windows. Использование командных (\*.bat и \*.cmd) файлов нельзя даже сравнивать с возможностями Unix-подобных систем.

С появлением Windows NT были добавлены следующие технологии и инструменты:

* **Windows Script Host -** один из элементов Microsoft Windows. Позволяет запускать сценарии, написанные с помощью скриптовых языков **VBScript**/**JScript**. Исполнение возможно в графической среде (wscript.exe) или в консоли (cscript.exe);
* **Windows Management Instrumentation -** расширенная и адаптированная под Windows реализация стандарта [WBEM](https://ru.wikipedia.org/w/index.php?title=WBEM&action=edit&redlink=1) принятого многими компаниями, в основе которого лежит идея создания универсального интерфейса мониторинга и управления различными системами и компонентами;
* **Windows PowerShell** - расширяемое средство автоматизации от Microsoft, состоящее из оболочки с интерфейсом командной строки и сопутствующего языка сценариев

На мой взгляд возможности современного интерфейса командной строки Windows не уступают Unix-подобным системам. Но определяющим является то, что для Unix-подобных систем эти возможности были изначально заложены в архитектуру, а в Windows они были добавлены в процессе эволюции. Этот факт влияет на масштабируемость решений и потенциал развития.

**Виртуальная машина Java, JRE, JDK**

Можно выделить такие основные причины возникновения и популярности Java и других интерпретируемых языков:

* Потребность обеспечить независимость кода от аппаратной реализации;
* Безопасность кода, максимальная изоляция пользовательской программы от ОС с целью повышения устойчивости;
* Возникновение Интернета как среды в которой живут самые разнообразные устройства;

Чтобы решить описанные проблемы были созданы среды-интерпретаторы кода. Текст программы преобразуется компилятором в промежуточный байт-код (компилятор может быть частью среды, а не отдельной программой) и исполняется интерпретатором.

В случае языка Java такой интерпретатор называется Java Virtual Machine (JVM). Следует отметить что JVM умеет также генерировать машинно-зависимый код во время выполнения программы. Благодаря этому снижение производительности не является значительным.

JRE - Java Runtime Environment включает в себя:

* Минимальную реализацию виртуальной машины, необходимую для исполнения Java-приложений, без компилятора и других средств разработки.
* Библиотеку Java-классов;
* Плагин для исполнения в браузере;

JDK – включает в себя:

* Компилятор Java (javac);
* Стандартные библиотеки классов Java;
* Примеры, документацию, различные утилиты
* JRE;

В состав JDK не входит интегрированная среда разработки на Java,