NORDIC INTERNATIONAL UNIVERSITY

Java dasturining asosiy tushunchalari.

O'qituvchi: Sofoyeva Fotima

Mavzular:

1. Dasturlash nima? | Asosiy tushunchalar.
2. Java nima? | tarixi va asosiy xususiyatlari.
3. Kompilyator (Compiler) va Interpretator (Interpreter).
4. “Bytecode” va “Machine code”.

Dasturlash nima?

- **Dasturlash** – bu kompyuterga ma'lum bir vazifani bajarish uchun buyruqlar ketma-ketlikni yozish usuli;
- **Kompyuter dasturi** – bu kompyuter uchun buyruqlar ketma-ketligi;
- **Dasturlash tili** – Dastur yozish uchun foydalaniladigan belgilar to'plami;
- **Dasturchi** – Dastur yozuvchi odam.

Dasturlash tillari 2 ta katta guruhga bo’linadi:

1. **“High-level programming language”- Yuqori darajali dasturlash tillari.** Odatda dasturlash yuqori darajali dasturlash tillari (Java, C++, C#, Python, Clojure, …) vositasida amalga oshiriladi. Bu dasturlash tillarining semantikasi odam tiliga yaqinligi tufayli dastur tuzish jarayoni ancha oson kechadi. Ular yordamida kod insonga "tushunarli“ tilda yoziladi. Ingliz tilini yaxshi biluvchilar programma kodini qiynalmasdan tushunishlari mumkin.
2. **“Low-level programming language”- Quyi darajarli dasturlash tillari.** Quyi darajali dasturlash tili ancha murakkab bo'lib ular juda maxsus sohalarda ishlatiladi va ularningmutaxassislari ham juda kam. Chunki quyi dasturlash tillari (masalan: assembler) ko'pincha mikroprotsessorlar bilan ishlashda kerak bo'lishi bo’ladi. Asosan turli dasturlash ishlari uchun yuqori darajali dasturlash tilidan keng foydalaniladi.

Qanday Dasturlash tillari bor?

1. Python
2. Java
3. JavaScript
4. PHP
5. SQL
6. Rust
7. C#
8. Go

Eng mashxur dasturlash tillari: https://www.stackscale.com/blog/most-popular-programming-languages/

Dasturlash tillarining farqlari

- Tezlik
- Xavfsizlik
- Xotira tejamkorligi
- Yozish qulayligi
- Platformasi (foydalanish maqsadi)
- Jamiyat va qo'llab-quvvatlash
- Kutubxonalar
- …

Java tarixi.

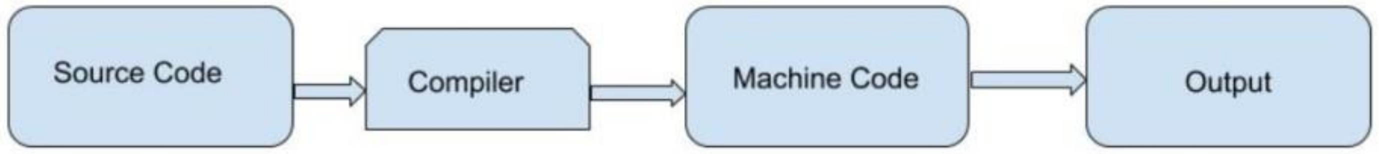
- Java platformasining dastlabki versiyasi 1995 yilda Sun Microsystems (2010 yil yanvaridan Oracle korporatsiyasining bir qismi) tomonidan chiqarilgan. Java dasturlash tilini ishlab chiqish 1991 yilda boshlangan. Dastlab bu til Oak deb nomlangan va u televizorlar uchun pristavkalarda foydalanish uchun mo'ljallangan edi.
- Java 1991 yilda Sun Microsystems kompaniyasida Jeyms Gosling, Patrik Naughton, Chris Warth, Ed Frank va Mayk Sheridan tomonidan ishlab chiqilgan. Java ning birinchi ishchi versiyasini ishlab chiqish uchun 18 oy kerak bo’lgan.
- Java - Indoneziyadagi orol bo'lib, u erda birinchi marta kofe ishlab chiqarilgan, java kofesi deb nomlangan.

Javaning xususiyatlari.

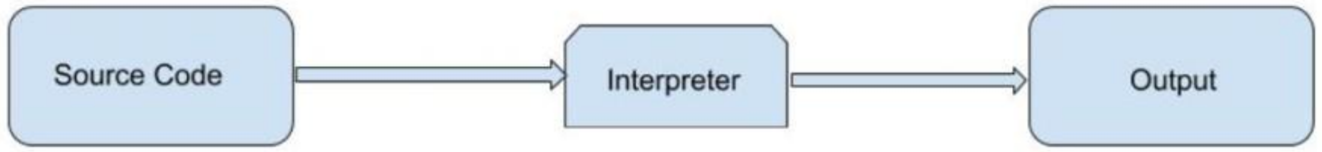
- **Object Oriented** − Javada hamma narsa obyekt shaklidadir. Javani obyekt modeliga asoslanganligi tufayli osongina kengaytirish mumkin.
- **Platform Independent –** Javada kompilyatsiya boshqa dasturlash tillaridagi kabi (C, C++ ..) aynan biror-bir platforma uchun emas balki platformalarga bog’liq bo’lmagan bayt-kodga kompilyatsiya bo’ladi. Baytkod esa JVM o’rnatilgan ixtiyoriy qurilmada ishga tushishi mumkin.
- **Simple –** Java o’rganishga oson qilib yaratilgan. Agarda siz OOP ning asosiy konsepsiyalarni bilsangizjavani o’zlashtirish siz uchun qiyin bo’lmaydi.
- **Secure –** Java Internetningtaqsimlangan muhitini qo'llab-quvvatlaganligi sababli, u bir nechta xavfsizlik xususiyatlarini ham ta'minlaydi. Soxtalashtirish, taqlid qilish va virus tahdidlari kabi xavfsizlikmuammolarini Internetda Java-dan foydalanish orqali bartaraf etish yoki kamaytirish mumkin.
- **Robust -** Robustning ma'nosi "kuchli". Bu shuni anglatadiki, Java dasturlari kuchli, chunki ular C yoki C++ dasturlari kabi osonlikcha ishdan chiqmaydi. Mustahkam bo'lish uchun asosan uchta sabab bor.

- **Multithreaded –** Java supports multi-threading programming that allows to write programs to do several works simultaneously. A thread is an individual process to execute a group of statements.
- **Interpreted –** During compilation, Java compiler converts the source code of the program into byte code. This byte code can be executed on any system machine with the help of Java interpreter in JVM.
- **High-Performance –** With the use of Just-In-Time compilers, Java enables high performance.
- **Distributed –** Java is designed to support the distributed environment of the Internet because it handles TCP/IP and UDP protocols. It facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications. This feature of Java makes us able to access files by calling the methods from any machine on the internet.
- **Dynamic –** Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.
- **Portability** - If a program gives the same result on every system machine, that program is called portable. Java programs are portable.

Kompilyator (Compiler) va Interpretator (Interpreter).



How Compiler Works



How Interpreter Works

Bytecode vs Machine code

Parameters	Byte Code	Machine Code
Definition and Meaning	A byte code acts as an intermediate code present between a machine code and a source code.	It basically refers to a set of various instructions that a machine can read and understand directly.
Level of Code	It is an intermediate-level code.	It is a low-level code.
Type of Instructions	It consists of hexadecimal, binary, and macro instructions such as swap, add, new, etc.	It consists of instructions in the binary language. Thus, the instructions are present in the codes of 0s and 1s.
CPU Understandable	A CPU cannot understand it directly.	Any CPU can directly understand as well as process this type of code.
Generation and Execution	We generate a byte code after source code compilation. But a CPU cannot directly run it. It entirely depends on an interpreter for its execution.	It is basically machine language. Thus the CPU can process it. It is present in binary format and does not need separate interpretation or compilation.
Role of Virtual Machine	The virtual machine first executes the bute code, and only then the CPU can process it.	We don’t need a virtual machine for the execution of machine code. The CPU can do that directly.
Machine-Specific	The byte code is suitable for execution by virtual machines and other software. But it is not very specific (directly) towards a machine.	The machine code is machine-specific in all aspects.
Platform Dependency	This type of code is platform-independent. It depends entirely on the virtual machine and also on the systems that already have an inbuilt virtual machine. Its execution can occur directly irrespective of the platform.	This type of code is platform-dependent. It is because we cannot run the object code generated out of one platform on the same OS. The objects vary a lot depending on the native instructions of the machine and its overall system architecture.
Conversion of Source Code	We don’t need to convert every source code into a byte code for its CPU execution. The source code that we write in a high-level language needs to get converted into byte code for its ultimate conversion into an object code. This way, the CPU can execute them.	Every source code ultimately needs to get converted into a machine code for CPU execution. It is the final step, irrespective of high-level or low-level language

Tez yozish uchun qo’llanma:

- https://www.typing.com/ - mashq qilish uchun web sayt.
- https://10fastfingers.com/typing-test/english – tezlikni test qilish uchun web sayt.
- https://monkeytype.com/ - tezlikni monitoring qilib boruvchi web sayt.