Programovací jazyky a Java

Program

- Program je algoritmus zapsaný jazykem srozumitelným pro cílový systém, který jej bude schopný vykonávat
  - příklad: algoritmus zapsaný pro člověka nebo pro stroj
- V počítačích programovací jazyky na různé úrovni abstrakce – podle toho i míra detailu algoritmu
  - strojový kód – blíže stroji - velmi detailní, ale rychlý
  - vyšší programovací jazyk – blíže člověku - snadněji použitelný, ale obvykle pomalejší

Programy a programovací jazyky

- Programovací jazyky
  - strojově orientované
    - strojový jazyk = jazyk fyzického procesoru (tj.: finální kód spustitelného programu)
    - asembler (jazyk symbolických adres)
  - vyšší jazyky
    - imperativní (příkazové, procedurální)
    - neimperativní (např. funkcionální - LISP)
    - pojem virtuální stroj (hypotetický procesor, který provádí příkazy vyššího programovacího jazyka

```
00000     di
00001     xor   a
00002     ld    de,65535
00005     jp    4555

00008     ld    hl,(23645)
00011     ld    (23647),hl
00014     jr    83


00000     di
00001     xor   a

A:000 00000000 .   DI   (SP):45043
B:175 ▒                    65297
C:255 @   BC:45055 SP:00000 50175
D:001 .   DE:00257 IX:00000 04555
E:001 .   HL:00000 IY:23610 23850
H:000 .
L:000 . R:000 NZ NC PO P T:00000
 UNIVERSUM Control ON  Call NON
```

http://www.fi.muni.cz/usr/jkucera/pv109/2002/xcimbur.html

```
C COMPUTE POSITIVE ROOT OF A
C QUADRATIC EQUATION
      READ INPUT TAPE 3, 201, A, B, C
  201 FORMAT (3I5)
      IF (A) 300,400,400
  300 STOP 1
  400 R=-B+SQRT(B*B-4*A*C)/(2*A)
      WRITE OUTPUT TAPE 4, 501, R
  501 FORMAT (F5.3)
      STOP
      END
```

http://www.root.cz/clanky/programovaci-jazyky-pouzivane-na-mainframech/

Hlavní rysy imperativních jazyků

- Např. C, C++, **Java,** Pascal, Basic, ...
- Zpracovávané údaje mají formu datových objektů různých typů, které jsou v programu reprezentovány pomocí proměnných resp. konstant

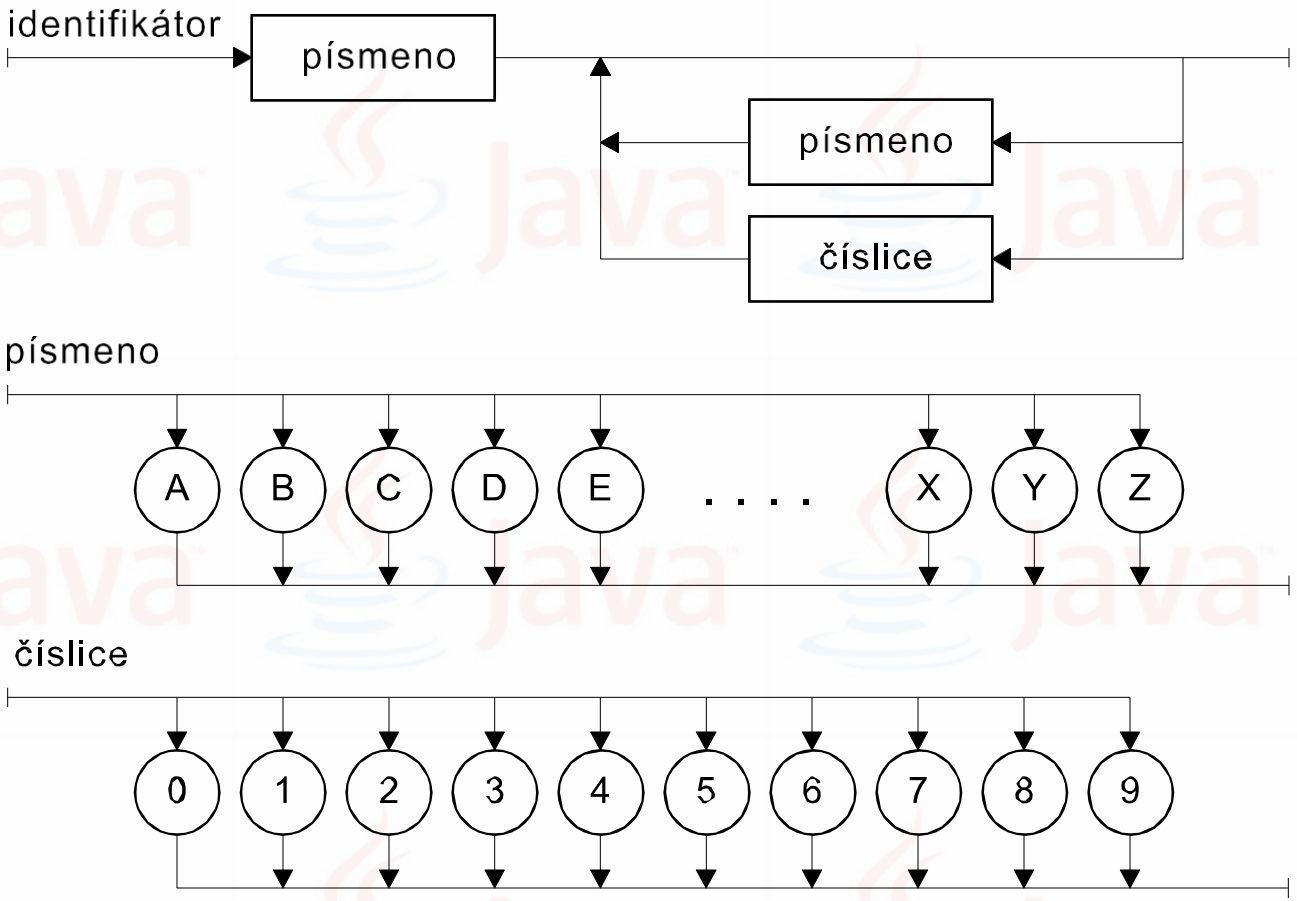
- Program obsahuje deklarace a příkazy
  - deklarace definují význam jmen (identifikátorů)
  - příkazy předepisují akce s datovými objekty nebo způsob řízení výpočtu

Vlastnosti programovacích jazyků

- Syntaxe
  - souhrn pravidel udávajících přípustné tvary dílčích konstrukcí a celého programu
  - tvoří tzv. gramatiku jazyka
- Sémantika
  - udává význam jednotlivých konstrukcí
- Prostředky pro popis syntaxe
  - syntaktické diagramy
  - různé varianty Backus-Naurovy formy
- Sémantika
  - význam zápisu
  - je obvykle popsána slovně

Syntaktické diagramy

Příklad: "identifikátor je posloupnost písmen a číslic začínající písmenem"



Rozšířená BNF

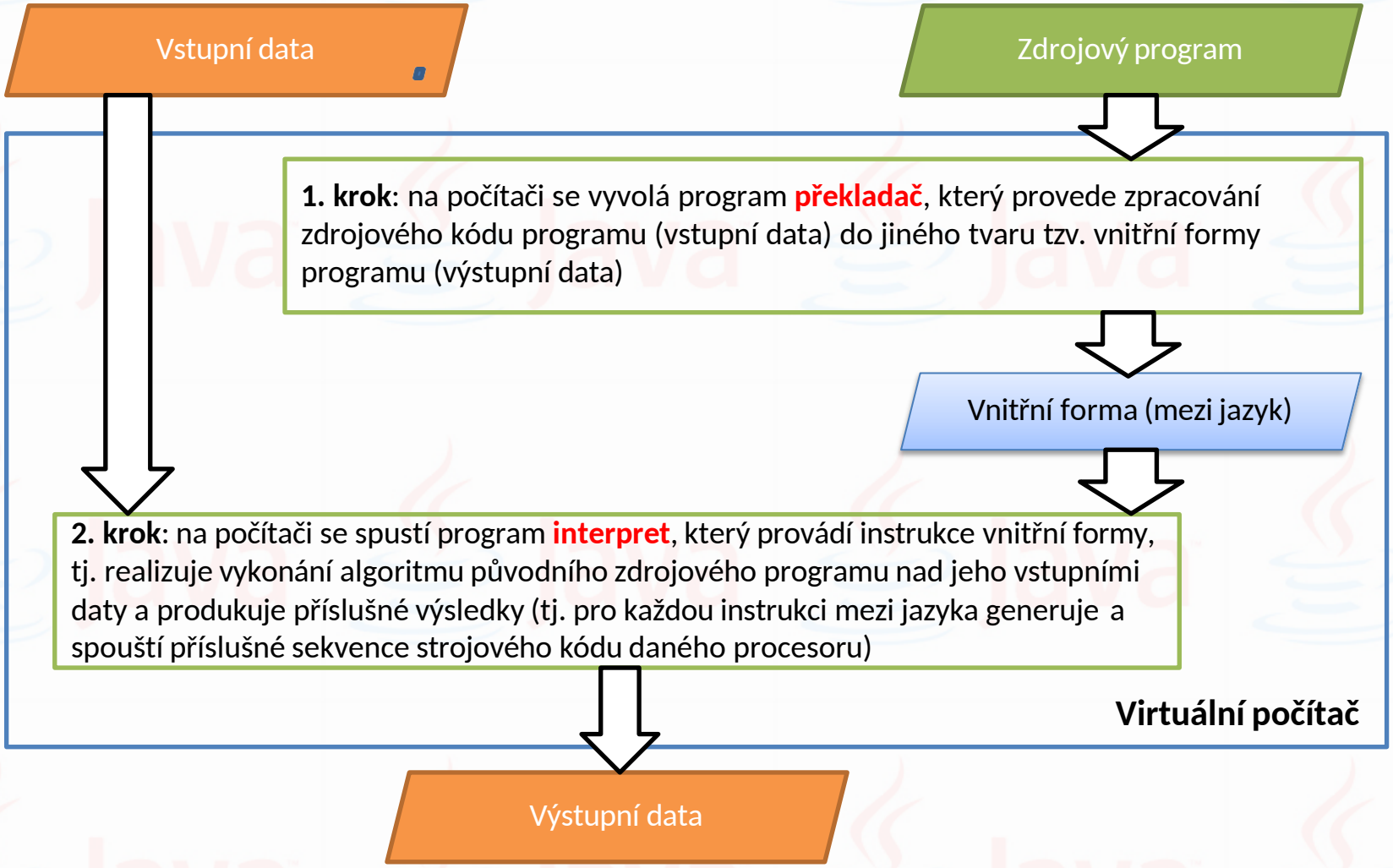
- Rozšířená Backus-Naurova forma (EBNF):
- Příklad: syntaktická definice pojmu *identifikátor*:
  - *identifikátor = písmeno {písmeno | číslice}*
  - *písmeno = 'A' | 'B' | 'C' | 'D' | ... | 'X' | 'Y' | 'Z'*
  - *číslice = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '9'*
- Neterminály:
  - *identifikátor*, *písmeno*, *číslice*
- Terminály:
  - *'A'* , *'B'*, ...
- Význam metasymbolů:

{x}	žádný nebo několik výskytů symbolu ***x***
x | y	jeden výskyt symbolu ***x*** nebo ***y***
[x]	žádný nebo jeden výskyt symbolu ***x***

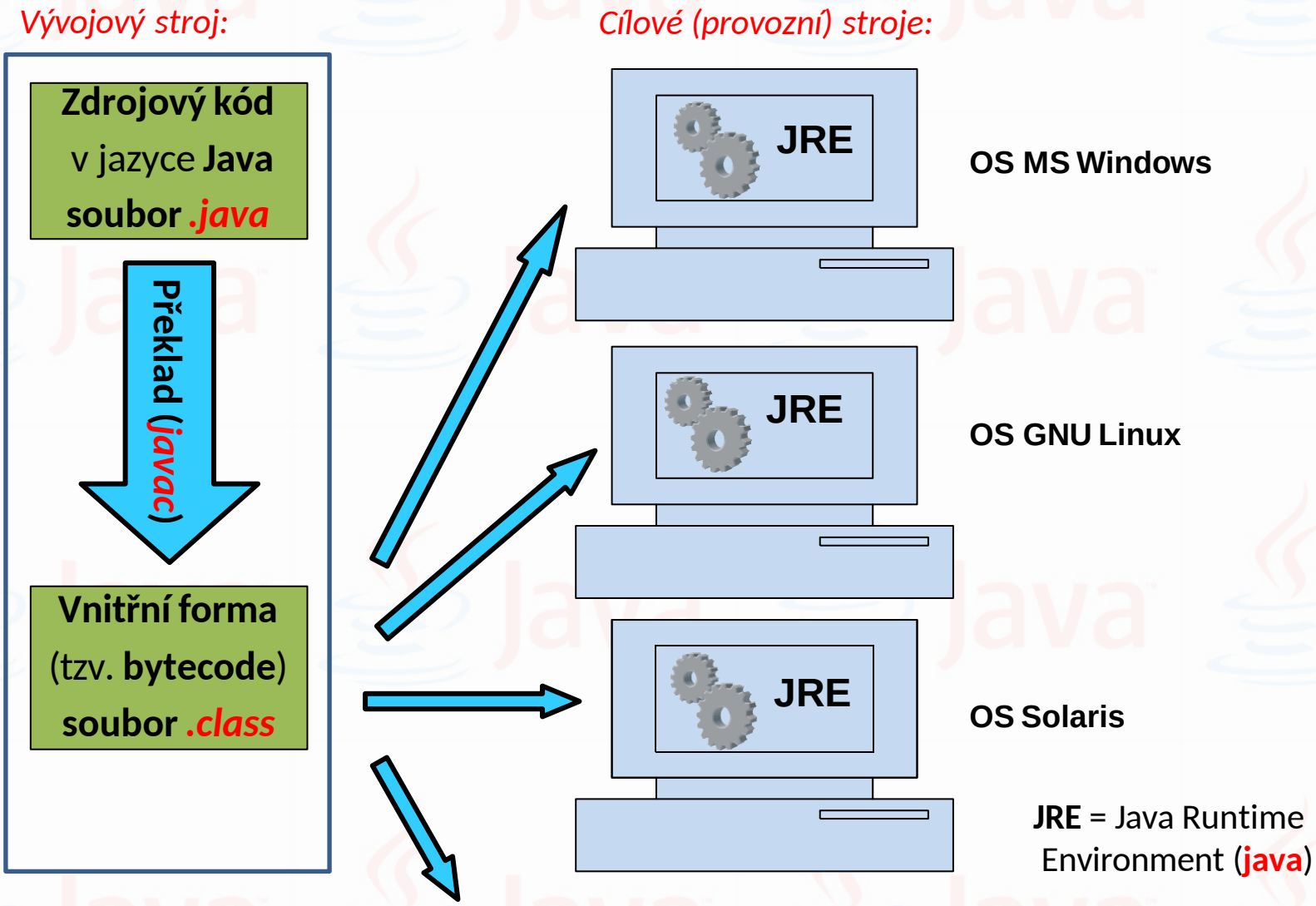
Výpočetní proces a paměť počítače

- Výpočetní proces je posloupnost akcí nad daty uloženými v paměti počítače (rozlišujeme hw úroveň a různé úrovně virtualizace)
  - data jsou v paměti reprezentována posloupnostmi bitů (bit má hodnotu *0* nebo *1*)
- Připomeňme:
  - paměť je tvořena řadou 8-mi bitových paměťových míst nazývaných bajt (z angl. byte, česky též slabika)
  - rozlišujeme vnitřní (operační) paměť a vnější paměť (např. disk)
  - každé paměťové místo vnitřní paměti má svou adresu (nezáporné celé číslo), která slouží pro jeho identifikaci
  - kapacita paměti se udává v KB (1 KB = 2^{10}B = 1024B), MB (1 MB = 2^{20}B) nebo GB (1 GB = 2^{30}B)
  - Pozn: nová terminologie "KiloBiByte"
- Instrukce strojového jazyka
  - předepisují aritmetické, logické a jiné operace s posloupnostmi bitů (bez ohledu na to, jaká data posloupnost bitů reprezentuje)

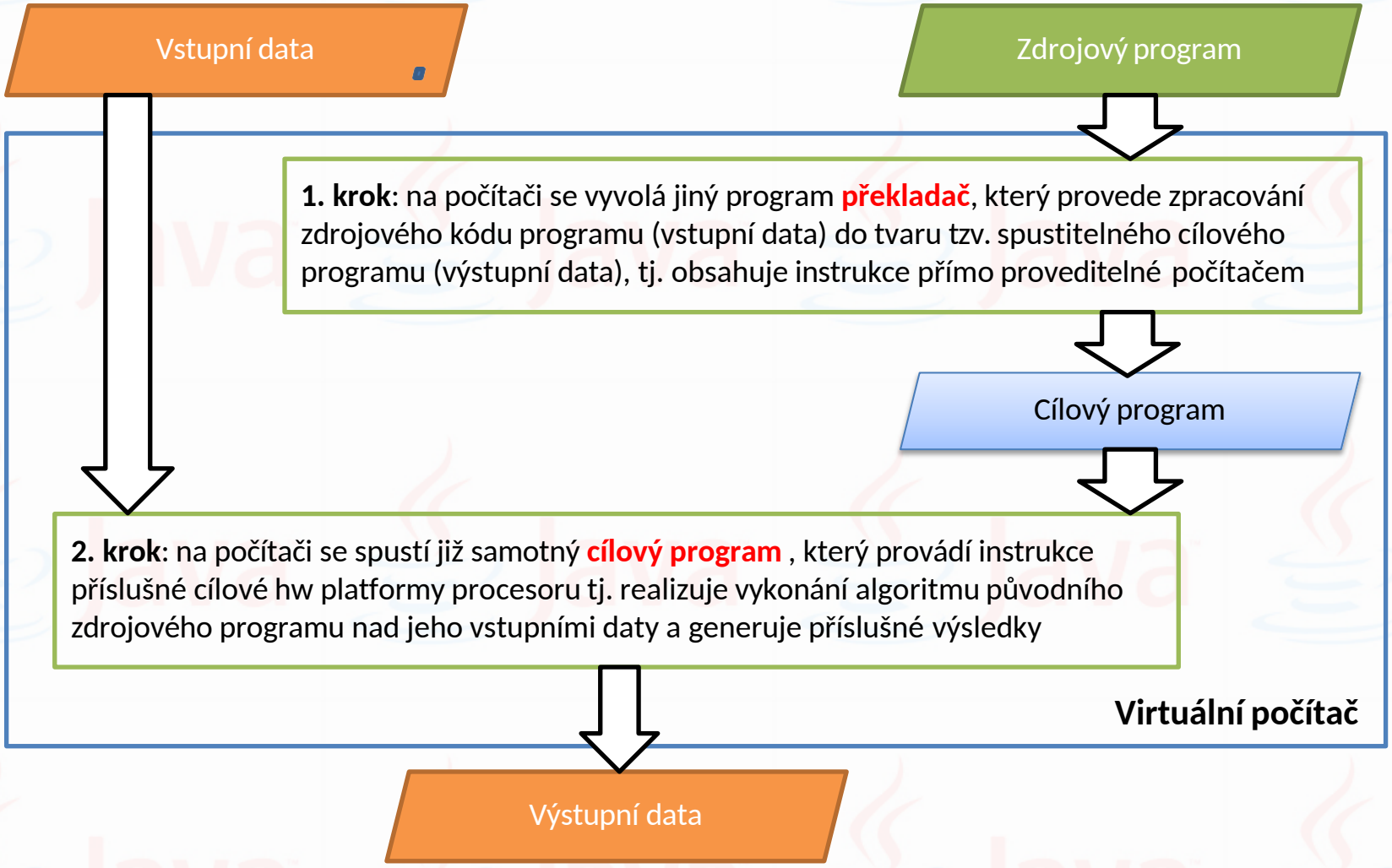
Implementace programovacích jazyků interpretační metoda



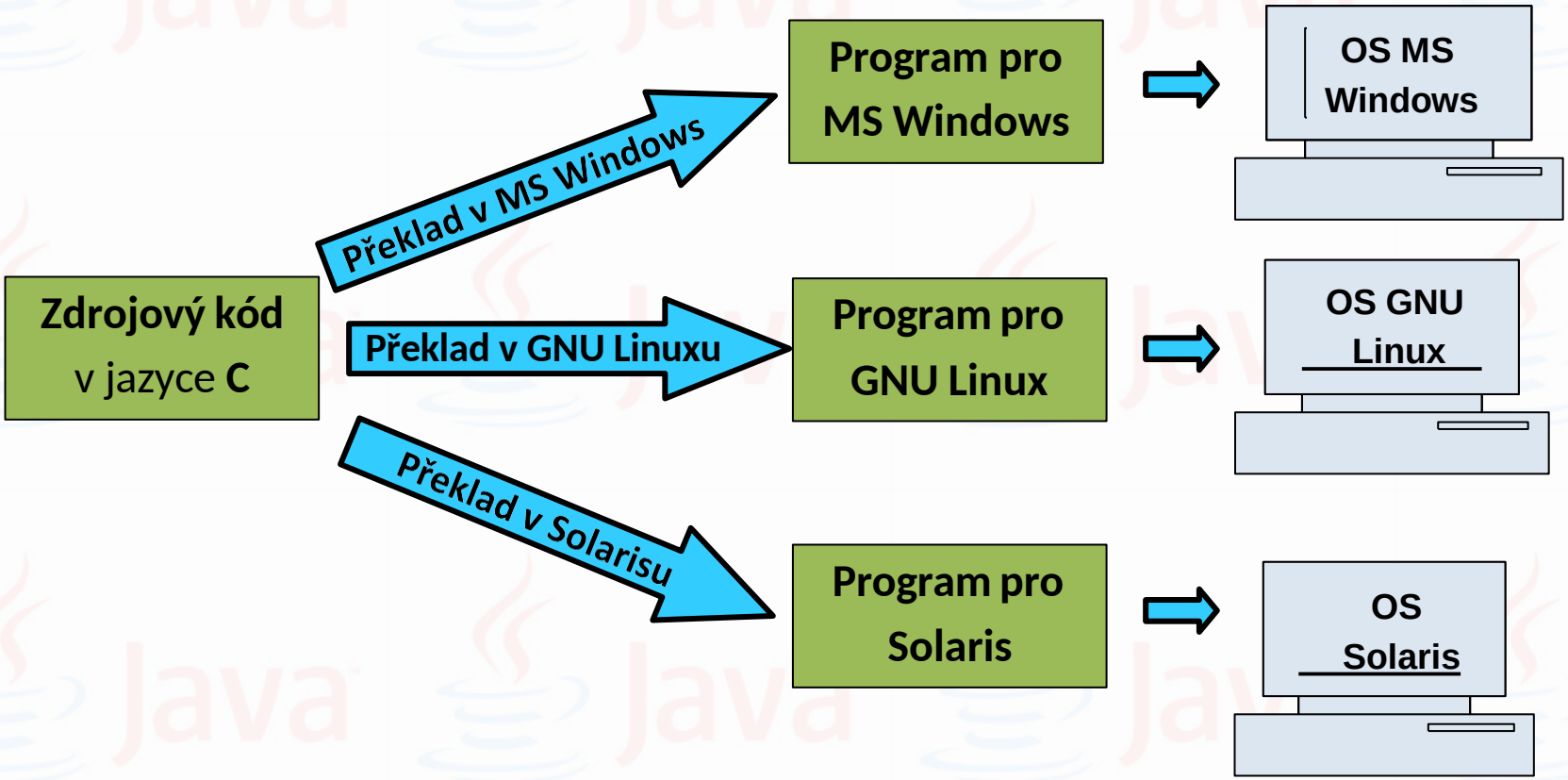
Interpretační metoda - jazyk Java - přenositelnost



Implementace programovacích jazyků kompilační metoda



Kompilační metoda - jazyk C



Důsledek: Pro každé různé cílové provozní prostředí (fyzický procesor/OS)je nutno provést kompletní překlad programu až do spustitelné formy (navíc není zajištěno identické chování těchto programů – rozdíly v implementaci atd.)

Vyšší programovací jazyky

- Podle přístupu k návrhu
  - procedurální – čistě sekvenční struktura kódu
  - objektové – objektová struktura s vnořenými sekvenčními kroky
    - bližší lidskému vnímání
  - funkcionální – přenos funkcionality jako argumentu
  - programování řízené událostmi

- Podle zaměření – orientační členění
  - systémové
    - používají kompilátor – rychlý proveditelný kód
  - aplikační
    - přístupnější syntaxe
    - kompilátory i interprety
  - webové
    - určeny do speciálního prostředí
    - většinou interpretované
  - speciální
    - pro speciální úkoly
    - umělá inteligence, IoT, …
  - vizuální nadstavby

IDE

- Integrated Development Environment
- Vývojové prostředí usnadňující vývoj SW aplikací
  - integrace všech potřebných funkcí pro tvorbu programového kódu
  - nápověda a kontrola syntaxe
  - testování kódu
- Přímá vazba na konkrétní programovací jazyk
- Volba podle preferencí programátora nebo standardů firmy
- Pro Javu např. Netbeans, IntelliJ Idea

Java

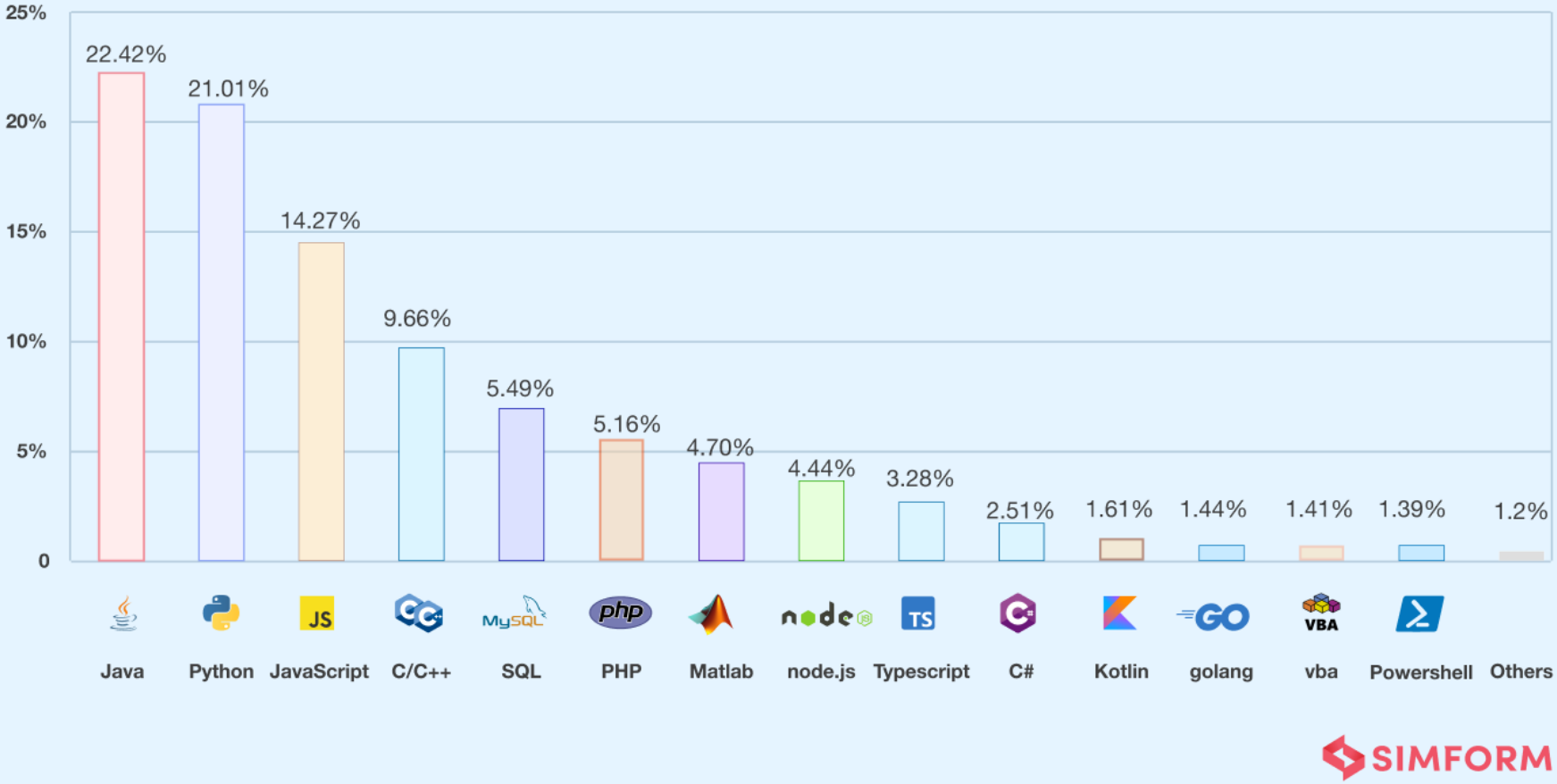
Úvod

- Volba programovacího jazyka:
  - pro prezentaci, návrh a ověřování algoritmů budeme používat jazyk Java
- Proč se věnovat jazyku Java?
  - jde o moderní vyšší, obecně použitelný programovací jazyk s vysokým stupněm zabezpečení
  - je objektově orientovaný, umožňuje však i klasické procedurální programování
  - vytvořené programy jsou zcela portabilní (Windows, Linux, Android, …)

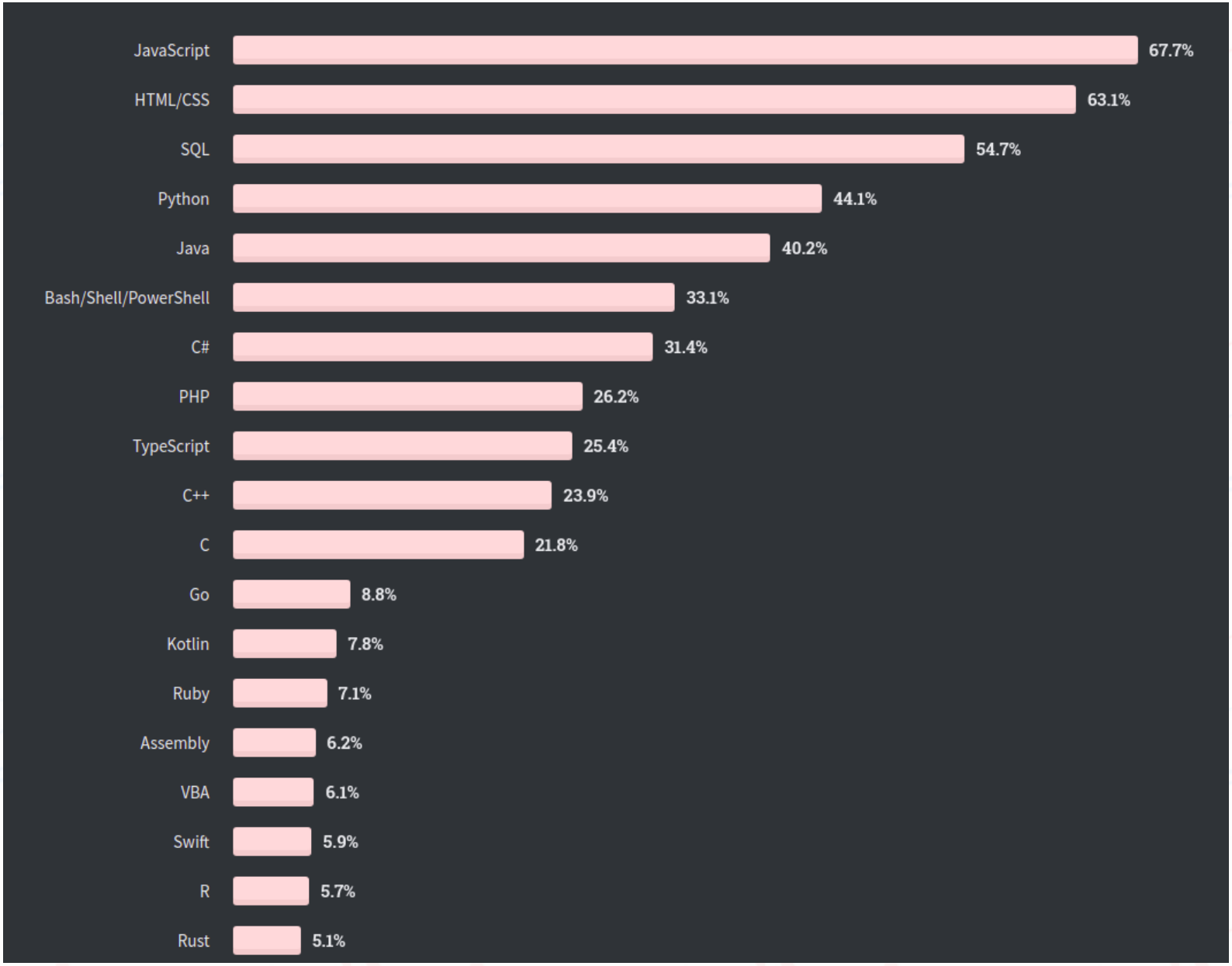
Cesta do pravěku

- 1957 programovací jazyk Fortran
- 1958 programovací jazyk Algol
- 1958 programovací jazyk Lisp
- 1959 programovací jazyk Cobol
- www.youtube.com/watch?v=2IRN4nZrDwk
- 1967 první objektově orientovaný jazyk Simula
- 1973 programovací jazyk C
- 1979 navazující jazyk C++
- 1980 Smalltalk-80
- 1995 Java (Sun Microsystems)

Technology trends across globe



https://www.simform.com/blog/top-programming-languages/



https://www.atatus.com/blog/top-seven-programming-languages-you-should-learn/

Java

- Dvě verze distribuce
  - JRE – Java Runtime Environment
    - spouštění vytvořené aplikace
  - JDK – Java Development Kit
    - základní vývojový balík nástrojů a jazyka
- různé implementace
  - Oracle Java - https://www.oracle.com/cz/java/
  - Open Java - https://openjdk.java.net/

Verze jazyka

- Podle rozsahu
  - (ME) – mobile edition
  - SE – standard edition
  - EE – enterprise edition
- Podle vývoje
  - https://en.wikipedia.org/wiki/Java_version_history
  - LTS verze
    - 8 – základ pro dnešní produkční systémy - FP
    - 11 – donedávna aktuální verze - moduly
    - 17, 21 – zcela nová LTS

Zpracování Java programu

- Program v jazyce Java je tvořen jedním nebo několika zdrojovými soubory. Tyto zdrojové soubory mají příponu *.java* např. *Program.java*
- Jazyk Java je implementován interpretačním způsobem – tomu odpovídá následující způsob zpracování.
  - Zdrojové soubory se přeloží překladačem (kompilátorem) *javac* do vnitřní formy (tzv. byte code) a uloží se do výstupního souboru s příponou *.class*: *Program.java* > *javac* > *Program.class*
  - Interpretaci vnitřní formy provede program *java* (v prostředí JVM – Java Virtual Machine) a tím se provede výpočet: *Program.class* > *java* > běh programu
- Program obvykle využívá řadu knihoven, které je třeba mít k dispozici jak při překladu, tak při interpretaci!!!
- Pro práci s Javou je klíčové vědět, kde hledat detailní informace - *https://docs.oracle.com/en/java/javase/21/docs/api/index.html*

Příklad JAVA programu

1. Vytvoříme soubor ***Program.java***:

```
public class Program {
  public static void main(String[] args)
    {
    System.out.println("První program");
    }
}
```

2. Spuštění překladače do byte-code:

   > ***javac Program.java*** -> vznikne soubor ***Program.class***

3. Spuštění interpretru:

   > ***java Program*** -> Výstup programu: *První program*

První program v jazyku Java

- Nejjednodušší zdrojový kód = jeden soubor
  - deklarace příslušnosti do konkrétního balíčku
    - `package program`
  - deklarace veřejné třídy
    - `public class Program`
  - hlavní procedura main - veřejná statická metoda bez návratové hodnoty a s daným seznamem parametrů
    - `public static void main(String[] args)`
- Soubor musí mít jméno shodné se jménem v něm uvedené veřejné třídy a příponu .java
- Tělo třídy nebo metody je uzavřeno v { }

Programátorská typografie

- soubor readme.txt
- jednotná volba syntaxe
- zápis programu
  - třída - *public class Obdelnik { ....}*
  - proměnné – *private int vyska;*
  - konstanty – *public final int SMERU = 8;*
  - konstruktory – *public Obdelnik() { ....}*
  - metody - *public int getX() { ....}*
- při víceslovném pojmenování se používá tzv. velbloudí notace: *mojePrvniMetoda*

- zápis programu

  yPos = y;

  nakresli();
- přístup k proměnným
  - obvykle nepřímo
    - metody *get* a *set*
    - zapouzdření

Komentáře

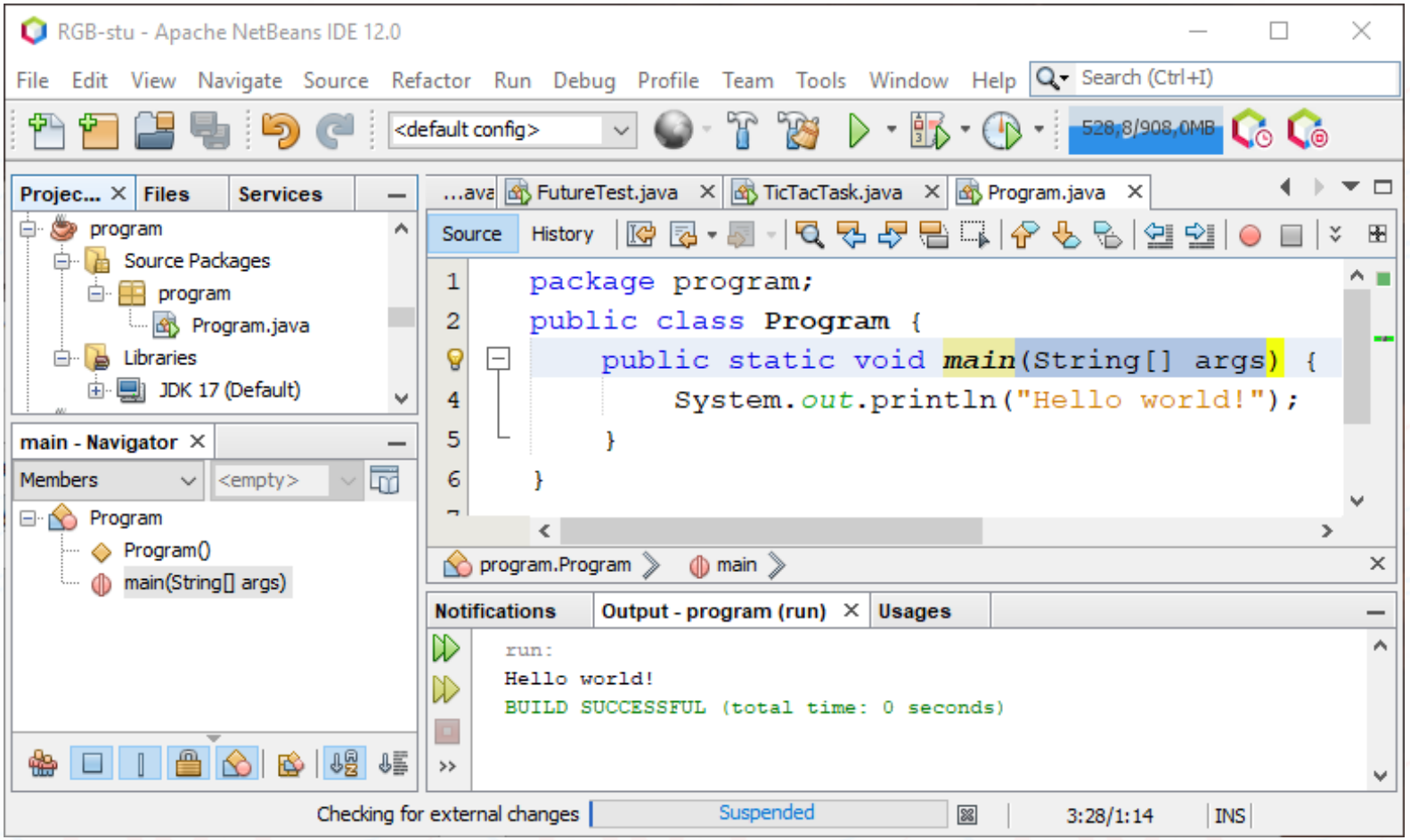
- Nutné pro čitelnost programu a týmovou spolupráci
- Dokumetovat každý krok (proměnná, metoda)
- Typy
  - řádkový - //
  - víceřádkový - /* …. */
  - dokumetační - /** ……*/ - javadoc

JavaDoc

- @author (třídy a rozhraní (interface))
- @version (třídy a rozhraní)
- @param (metody a konstruktory)
- @return (metody)
- @exception (výjimky, synonymum k @throws)
- @see (přidá „see also …“)
- @since (od určité verze)
- @serial (nebo @serialField nebo @serialData)
- @deprecated (v útlumu, nerozvíjeno)

Integrované vývojové prostředí (IDE)

Programy v jazyku Java budeme vytvářet pomocí IDE NetBeans 23, který přípravu programu, jeho překlad a provedení zjednodušuje:



Příprava vlastního prostředí s IDE NetBeans 12.x

- Nutně potřebujeme 2 součásti:

- Již zmiňovaný Java SE JDK (Java Development Kit), který stáhneme a nainstalujeme
  - https://www.oracle.com/java/technologies/downloads/
  - doporučována je i instalace dokumentace Javy (stejná stránka)

- Vlastní IDE NetBeans
  - https://netbeans.apache.org/

- Pro tvorbu GUI v prostředí Java FX nutno dále doplnit o tuto knihovnu a tzv. Scene Builder