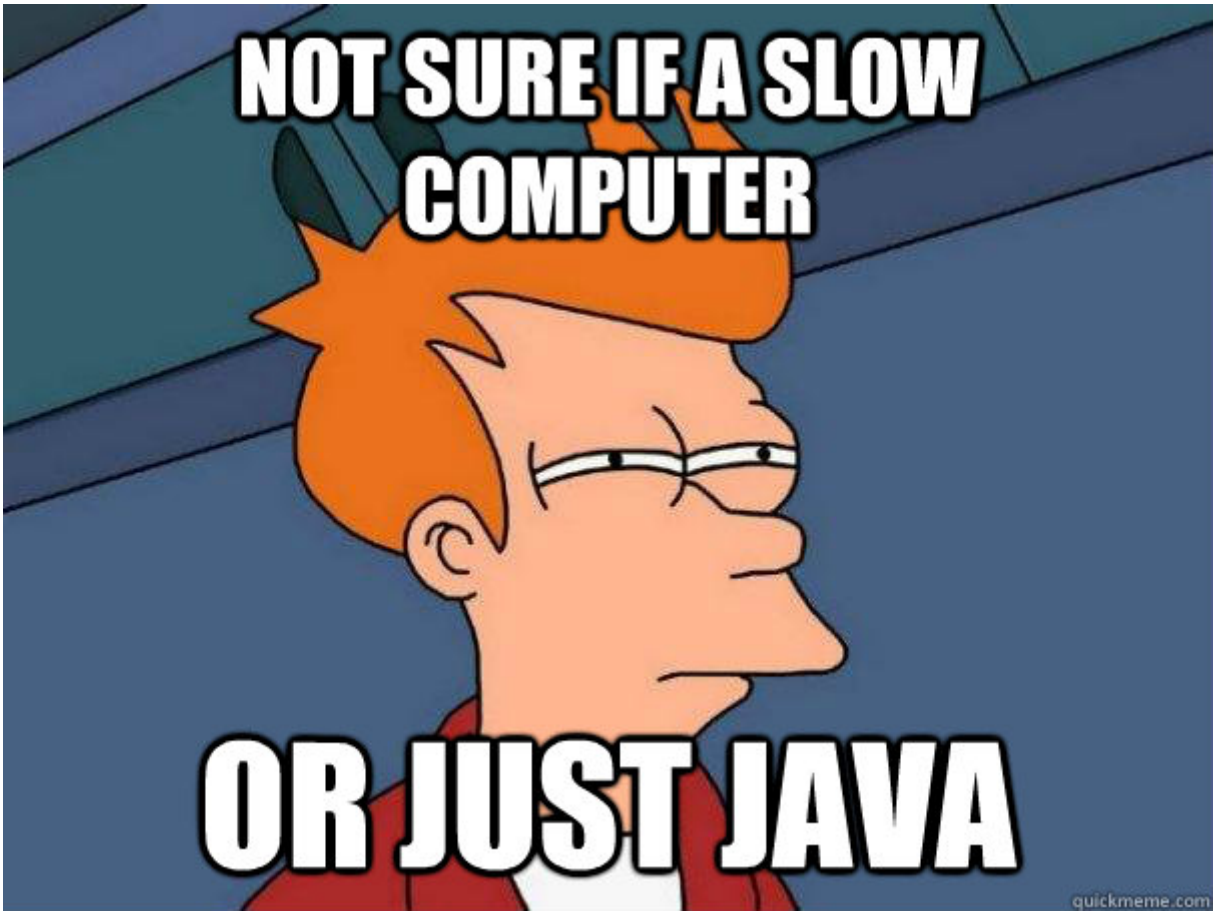
Umíme (trochu) Python nebo C, začínáme Javou

Table of Contents

Rychlost Javy . . . 1
Python je taky pomalý… . . . 2
Zpátky na zem… . . . 2
Java jako programovací jazyk . . . 3
Java v budoucnu . . . 3
Proč Java/1 . . . 3
Proč Java/2 . . . 3
Další přednosti . . . 4
Omezení . . . 4
Rozdíly . . . 4
Java pro programátora . . . 4
Platformy Java . . . 4
Aktuální verze . . . 5
Licence Javy . . . 5
Stažení Javy . . . 5
Budoucnost Javy . . . 6

Rychlost Javy

- takže proč vlastně ještě Java?



Python je taky pomalý...

- zase se v něm ale rychleji píše:

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

```python
print("Hello, World!")
```

Zpátky na zem...

- Delší kód ještě neznamená delší psaní.
- U rozsáhlejším programů se rozdíl stejně stírá.
- Python běží pomaleji a nic moc s tím nenaděláme.
- Rozhodující je kvalita a dostupnost existujících knihoven.
- Ve finále rozhodují požadavky a dovednosti týmu a zákazníka.

Java jako programovací jazyk

- Dle indexu TIOBE (na bázi výskytu "java programming" na webu) stále 3. nejpopulárnější jazyk na světě, ovšem s mírně klesajícím podílem, po jazycích Python a C++.
- Je *imperativním* jazykem vysoké úrovně nově s *funkcionálními* prvky.
- Je jazykem *univerzálním* — není určen výhradně pro specifickou aplikační oblast.
- Je jazykem *objektově-orientovaným* — program používá volání metod objektů (zasílání zpráv objektům).
- Je OO jazykem disponujícím *objektovými třídami*, což automaticky neplatí pro všechny jazyky s objekty.
- Ideovým předchůdcem Javy je *C++*.

Java v budoucnu

- Pro tradiční typy serverových podnikových aplikací a systémů zůstává Java (Enterprise Edition) klíčovou platformou spolu s .NET
- Perspektivním směrem vývoje je zachování Java platformy (JVM, stávající knihovny, aplikace, aplikační prostředí)
- Rychle se vyvíjejí *skriptovací jazyky* na této platformě: *Groovy*, *JRuby*, *Jython*, a zejména v poslední době *Kotlin*…)
- Mnoho novějších jazyků bylo inspirovaných Javou: *C#*, *Groovy*, *Ruby*, *Scala*, a zejména *Kotlin*.

Proč Java/1

- Java je bohatý, robustní jazyk pro vývoj a běh jednoduchých i rozsáhlých aplikací.
- Vývoj je efektivnější než na jejich předchůdcích (C++) a výsledné aplikace "běží všude" — tyto výhody nabízí i další moderní jazyky.
- Oproti novějším jazykům se Javě vytýká "upovídanost" — kód je delší a některé obraty se v jiných jazycích napíší kratšeji.
- Silnou typovaností, běhovou bezpečnostní kontrolou, efektivní automatickou alokací a dealokací paměti (garbage collection), stabilními knihovnami vč. open-source a rozsáhlým souborem dobrých praktik nabízí aplikacím velmi vysokou robustnost.

Proč Java/2

- Nezavádí zbytečnosti a vede ke správným a dále uplatnitelným návykům.
- Je velmi perspektivní platformou pro vývoj open-source i komerčního SW, mj. pro *extrémně velké* množství volně dostupných knihoven.
- Je spolu s následníkem Kotlin standardem *mobilního programování* na systému Android.
- Pro iOS (Apple) je základem jazyk Swift — ten ovšem vidíme fakticky jen tam.

Další přednosti

- Java *podporuje vytváření správných návyků* v objektovém programování.
- Program v Javě je *přenositelný* na jiné počítače a operační systémy na úrovni *zdrojového i přeloženého* kódu.
- Přeložený javový program běží v tzv. Java Virtual Machine (JVM).
- Zdrojový i přeložený kód je tedy přenositelný mezi všemi obvyklými platformami (UNIX/Linux, Windows, MacOS X).

Omezení

- Neuplatní se ve většině skutečně malých a low-power zařízení (kategorie Arduino, 8bitové kontrolery apod.)
- Lze ji použít pro zařízení třídy Raspberry Pi (32bitová)
- Ve variantě jazyka Java Card se používá pro vývoj bezpečných aplikací pro chytré karty.

Rozdíly

"Coding techniques used in a practical Java Card program differ significantly from those used in a Java program. Still, that Java Card uses a precise subset of the Java language speeds up the learning curve, and enables using a Java environment to develop and debug a Java Card program (caveat: even if debugging occurs with Java bytecode, make sure that the class file fits the limitation of Java Card language by converting it to Java Card bytecode; and test in a real Java Card smart card early on to get an idea of the performance); further, one can run and debug both the Java Card code for the application to be embedded in a smart card, and a Java application that will be in the host using the smart card, all working jointly in the same environment." [Wikipedia: heslo Java_Card]

Java pro programátora

Konkrétní možnosti:

- Výborná podpora vývojovými nástroji: dvojice *IntelliJ IDEA* a tradiční *Eclipse*, ale klidně i *Visual Studio Code*.
- Java má *automatické odklizení nepoužitelných objektů* (automatic garbage collection).
- Java je přímočařejší a jednodušší než C++ (méně syntaktických konstrukcí, méně nejednoznačností v návrhu), což zlepšuje čitelnost a redukuje riziko chyb. Obdobné platí i pro další populární jazyky, jako je Python.

Platformy Java

Java SE

(Standard Edition) pro běžná i serverová použití na běžných "plných" platformách (Linux, Mac

OS, Win) — Java SE at a Glance

Java EE

(Enterprise Edition) kromě samotného běhového prostředí (JVM) zahrnuje také množství dalších API. Existuje 20 implementací (aplikačních serverů) Java EE, jako je GlassFish Server, IBM WebSphere Application Server, Red Hat JBoss Enterprise Application Platform a další. Java EE at a Glance

Java ME

(Micro Edition) do přenosných, vestavěných a malých zařízení (mobily, televizory, IoT) — Java ME přehled

Java Card

platforma pro bezpečné chytré karty; je odlišné nejen API (knihovní třídy), ale i jazyk

Aktuální verze

Stav k lednu 2025:

- *Java Standard Edition 23* je aktuální "ostrou" verzí
- S kadencí pololetí vždy přijde verze nová (= jaro 2025 bude verze 24)
- Aktuální informace najdete vždy na webu Oracle Oracle Technetwork/Java.
- K předpokládanému vývoji a podpoře existuje Oracle Java SE Support Roadmap

Licence Javy

- Na stránkách https://www.java.com/download/ je java dostupná Oracle Java SE pro všechny platformy.
- Pro testovací, vývojové, demonstrační nekomerční využití je Java SE zdarma.
- Pro komerční provoz je Oracle Java placená.
- Licenční poplatky se liší dle způsobu nasazení (desktop až cloud) od cca $2.5 do $25 měsíčně na jeden procesor a zahrnuje podporu — podobně jako např. RHEL.
- Pro plné využití zdarma je možné sáhnout po OpenJDK (*Oracle's free, GPL-licensed, production-ready OpenJDK JDK 19 binaries for Linux, macOS, and Windows*) a JDK.Java.net

Stažení Javy

- samotné *vývojové prostředí* (JDK), např. OpenJDK 22 či Oracle Java SE 8 JDK
- jen *běhové prostředí* (JRE), např. Java SE 8 JRE: to nám tady *nestačí*, chceme vyvíjet
- Připravili jsme pro vás tutoriál, jak Javu nainstalovat.

Budoucnost Javy

- Na závěr optimistického úvodu si pročtěte zajímavý článek analytika od *Forrester*: Java Is A Dead-End For Enterprise App Development