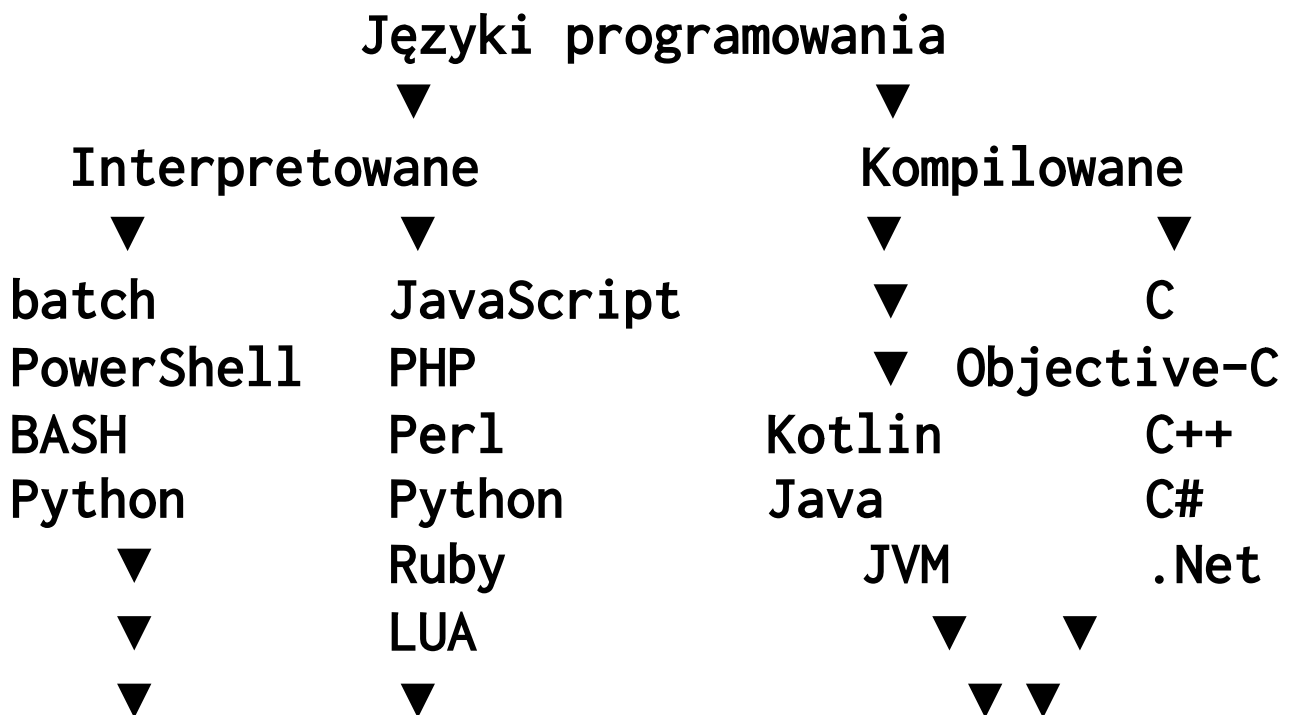


1. Wprowadzenie do języków programowania obiektowego



- w którym języku zaimplementować moją **SuperAplikację** (SA)? ...
- ... to zależy ... od zastosowania aplikacji ...

2. Zastosowanie grup języków programowania



- Python oraz PHP mogą pracować jako powłoki systemowe (DevOps)
- Python jest domyślnie zainstalowany w systemie Linux – można go traktować jako Powłokę systemową
- PowerShell 2.0 jest domyślnie zintegrowany z .NET od Windows 7 (zastępując pliki wsadowe Batch)
- PowerShell został otwarty i jest dostępny dla macOS oraz Linuksa
- BASH jest domyślną powłoką systemową w macOS, Linux oraz Unix

3. Interpreterzy oraz uruchamianie skryptów

- JavaScript – u klienta, przeglądarka internetowa zgodna ze standardami W3C
- JavaScript – na serwerze, node.js (brak obsługi JS 1:1 jak w przeglądarce)
- PHP – interpreter PHP<--> serwer HTTP (generuje dokument HTML)
- PHP – w terminalu PHP CLI (wyświetla: echo/print())
- PHP i Python – istnieją kompilatory do pliku EXE
- Python, BASH, Perl, PowerShell – skrypty uruchamiane przez terminal
- Python(+Tk/Qt) – aplikacje GUI

4. Kompilacja języków – platformy docelowe

- będąc autorem **SuperAplikacji** (SA) jak dostarczyć ją „wszędzie”?

ARM	x86	(32bit)
PowerPC	x86_64	(64bit)

- platforma docelowa: hardware (CPU) + software (SO, system operacyjny)
- dla Hardware problemem jest zestaw instrukcji procesora (różne instrukcje)
- dla Software problemem są biblioteki systemowe (i ich brak)
- emulator aplikacji sklepu Google Play (ARM) dla Windows (x86)
- uruchamianie aplikacji Win32 (x86) na urządzeniach mobilnych (ARM) zakończone „porażką” – uruchamianie w Chmurze Microsoft i streaming
- oprogramowanie WINE jako warstwa translacji aplikacji Windows dla systemów macOS, Linux oraz Unix (np. DirectX → OpenGL)
- Apple zaczął sprzedaż swoich komputerów z ARM (M1) – konwersja aplikacji
- Microsoft próbował swoich sił z ARM – WindowsRT – zakończone „porażką”

5. Kompilacja języków – bez problemów

- jak pokonać problemy Platformy docelowej? ...
- ... tak jak zrobiła to Java – która powstała dla układów scalonych różnych producentów (koniecznej kompilacji dla konkretnego CPU-mikroukładu) ...
- ... dzięki temu wylądowała jako platforma uniwersalna dla aplikacji desktopowych dla każdej platformy

App (c++)

???

SO (softw)

CPU (hardw)

Pomysł Javy (wieloletni sukces)

??? = JVM (Java Virtual Machine)

kompilewanie#1 do bajtkodu (kodu pośredniego)

kompilewanie#2 uruchomienie Aplikacji na Platformie docelowej

Pomysł Microsoftu (hmm, „porażka”?)*

??? = .NET (framework* dla Windows)

**miał być na początku nowym API dla systemów Windows, czyli częścią rdzenia systemu, ale się nie udało, więc stał się dodatkowym frameworkiem (XP, Vista, 7), stała część systemu (8, 8.1, 10)*