

Programování pro přírodovědce

v jazyce Java

Juraj Jašík

juraj.jasik@gmail.com
Univerzita Karlova v Praze
Přírodovědecká fakulta
Katedra organické a jaderné chemie

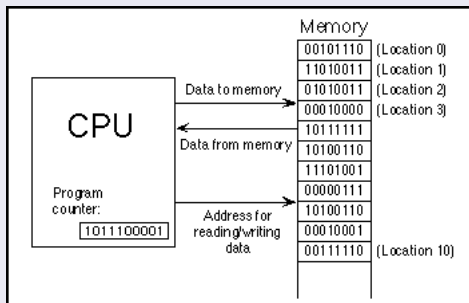
October 7, 2012

1

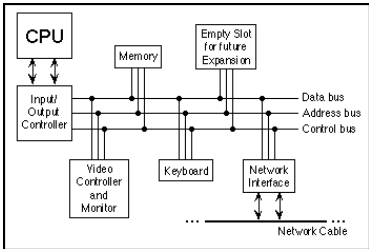
- 

- Fetch-and-execute cycle
 - *fetch* - načítanie inštrukcie z pamäte
 - *execute* - vykonanie inštrukcie
 - fetch, execute, fetch, execute, fetch, execute, fetch, execute, ...
- Program counter (PC)
 - jeden z *registrov* (pamäťové miesta v CPU)
 - kde sa práve nachádzame v programe
 - adresa inštrukcie, ktorú CPU načíta z RAM
 - počas *execute* sa PC aktualizuje a ukazuje na ďalšiu inštrukciu pre vykonanie

Prepojenie CPU s RAM



- jediná činnosť procesora je načítavanie inštrukcií z RAM a ich vykonávanie
- ako sú obsluhované periférne zariadenia? (myš, klávesnica, pevný disk, monitor, sieťová karta ...)
- *device driver*
 - program ktorý CPU vykonáva keď komunikuje s periférnym zariadením



- ako CPU zistí, že mu periféria posiela dáta?
 - *pooling*
 - neustále dotazovanie na dáta
 - neefektívne
 - *prerušenia* (interrupts)
 - signál poslaný perifériou
 - CPU odloží všetko, čo robí bokom, aby mohol spracovať prerušenie
 - po spracovaní sa vráti k tomu, čo robil pred prerušením
- Interrupt handler
 - program určený na spracovanie *prerušenia*
 - súčasťou *device driveru*

- Asynchrónna udalosť (asynchronous event)
 - nepredvídateľná (v čase) požiadavka periférie, alebo užívateľa
 - prerušenie delegované na (nad) úroveň operačného systému
 - kliknutie na ikonu, prepnutie okna, stlačenie klávesy, pripojenie USB zariadenia, ...
 - väčšinou generované operačným systémom
 - event handler
 - naturálne podporované Javou
- Multitasking
 - vykonávanie viacerých činností “naraz”
 - pomocou prerušení

- Thread (vlákno)

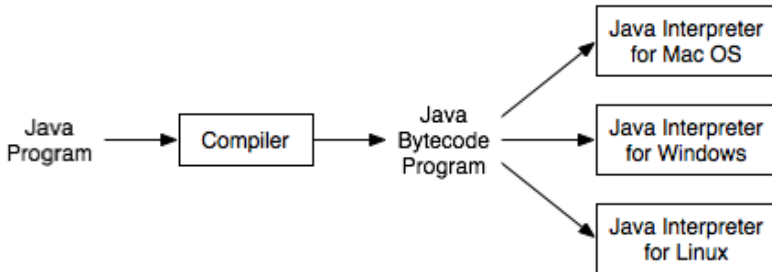
- individuálna úloha v multitaskingu
- počítač musí byť schopný prepínať medzi jednotlivými vláknami
- vlákno beží do vtedy, pokiaľ nenastane určitá situácia:
 - vlákno sa dobrovoľne vzdá kontroly a umožní CPU vykonávať iné vlákno
 - vlákno čaká na asynchrónnu udalosť - je *blokové*. Ostatné vlákna môžu bežať. Keď nastane asynchrónna udalosť, vlákno sa zobudí
 - činnosť CPU je prepnutá na iné vlákno, keď nastane prerušenie z časovača (*preemptive multitasking*). Vlákna tak bežia postupne preddefinované zlomky času
- naturálne podporované Javou

1 Úvod

- Strojový kód
- Asynchrónne udalosti, prerušenia
- **Java Virtual Machine**
- Základné bloky programu
- Objekty a objektovo orientované programovanie (OOP)
- Užívateľský interface
- Internet

- Strojový kód
 - veľmi jednoduché inštrukcie, priamo vykonávané CPU
- Jazyky vyššej úrovne abstrakcie (high-level programming languages)
 - Java, Pascal, Basic, C++, C#, ...
 - nutný preklad do strojového kódu
- Kompilátor (compiler)
 - nástroj na preloženie vyššieho jazyka do strojového kódu
 - preložený program - *executable*
 - rôzne typy počítačov - rôzne typy strojového kódu - rôzne kompilátory
- Interpreter
 - prekladá vyšší jazyk inštrukciu po inštrukcii
 - podobné CPU - *fetch-and-execute* cyklus
 - podstatne pomalšie ako skompilovaný program

- Java Virtual Machine (JVM)
 - Java je kombinácia kompilátora a interpretera
 - program v Jave je skompilovaný do Java bytecode
 - Java bytecode beží na virtuálnom stroji *Java Virtual Machine*
 - Java bytecode môže byť spustený na ľubovoľnom počítači, pokiaľ má JVM



- Výhody
 - Java bytecode interpreter je jednoduchý program - oveľa menšie náklady na vývoj ako kompilátor
 - internetová bezpečnosť - JVM pôsobí ako firewall
- Just-in-time kompilátor
 - prekonanie problému s pomalými interpretermi
 - prekladá väčšie celky Java bytekódu do strojového kódu počas vykonávania programu

1 Úvod

- Strojový kód
- Asynchrónne udalosti, prerušenia
- Java Virtual Machine
- Základné bloky programu
- Objekty a objektovo orientované programovanie (OOP)
- Užívateľský interface
- Internet

- 2 aspekty programovania:
 - dáta
 - *premenná* (variable) a jej *typ* (type)
 - inštrukcie
 - kontrolné štruktúry (control structures)
 - podprogramy (subroutines)
- *premenná*
 - pomenované miesto v pamäti
 - *typ premennej* určuje aké dáta *premenná* obsahuje

priradovací príkaz (assignment statement)

```
interest = principal * 0.07;
```

- kontrolné štruktúry (control structures) - inštrukcie umožňujúce zmenu v toku programu
 - sľučky, cykly (loops) - opakovanie sekvencie inštrukcií
 - vetvenie (branches) - zmena toku programu na základe nejakej podmienky

if statement

```
if (principal > 10000)
    interest = principal * 0.05;
else
    interest = principal * 0.04;
```

- podprogramy (subroutines)
 - umožňuje rozdeliť komplexný program na jednoduchšie časti
 - postupnosť inštrukcií vykonávajúca určitú konkrétnu úlohu
 - zoskupená pod určitým menom
 - procedúry, funkcie

```
drawHouse();
```

1 Úvod

- Strojový kód
- Asynchrónne udalosti, prerušenia
- Java Virtual Machine
- Základné bloky programu
- Objekty a objektovo orientované programovanie (OOP)
- Užívateľský interface
- Internet

- štruktúrové programovanie
 - riešenie komplexného problému sa rozdelí na riešenie viacerých jednoduchých problémov - *top-down programming*
 - vytvorenie *inštrukcií* potrebných na vyriešenie problému
 - nedokonalá podpora pre *dátové štruktúry*
 - ťažké opätovné využitie už naprogramovaných vecí (*modularita*)
- softwérový modul
 - komponent väčšieho systému
 - interaguje so zvyškom systému jednoduchým, presne definovaným a priamočiarym spôsobom
 - má byť ľahko pripojiteľný do systému
 - pokiaľ modul funguje korektne, detaily práce vnútra modulu nie sú pre celý systém podstatné - *ukrytie informácií*

- objekt
 - druh modulu
 - dáta + podprogramy na manipuláciu s dátami
 - samostatná entita, ktorá má *vnútorný stav* (dáta) a ktorý dokáže reagovať na *príkazy* (volania podprogramov)
 - samotné dáta by mali byť mimo objekt neprístupné, manipuluje sa s nimi pomocou príkazov

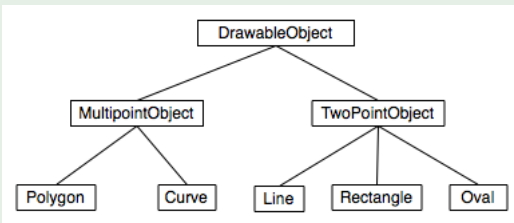
objekt MailingList

- dáta:
 - zoznam mien a adries
- príkazy:
 - pridaj meno a adresu do zoznamu
 - vymaž záznam v zozname
 - vypíš zoznam

- objektovo orientovaný prístup
 - identifikovať objekty v probléme
 - identifikovať príkazy, na ktoré by mali objekty vedieť reagovať
 - výsledný program je kolekciou objektov, ktoré majú svoje vlastné dáta a svoje vlastné príkazy
 - objekty interagujú navzájom pomocou príkazov
- polymorfizmus
 - rôzne objekty môžu rôzne reagovať na rovnaký príkaz
- trieda (class)
 - objekty, ktoré majú rovnakú štruktúru dát a rovnaké reakcie na príkazy spadajú do istej *triedy*

- podtrieda (subclass) a dedenie (inheritance)
 - *podtrieda* určitej triedy *dedí* všetky vlastnosti triedy
 - a pridáva ďalšie vlastnosti (dáta a príkazy)
 - môže zmeniť reakciu na niektoré príkazy (*override*)

DrawableObject

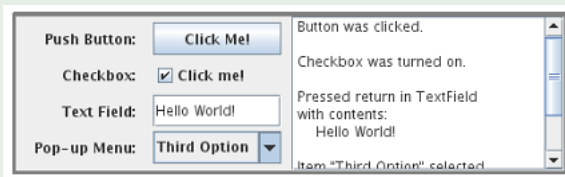


1 Úvod

- Strojový kód
- Asynchrónne udalosti, prerušenia
- Java Virtual Machine
- Základné bloky programu
- Objekty a objektovo orientované programovanie (OOP)
- **Užívateľský interface**
- Internet

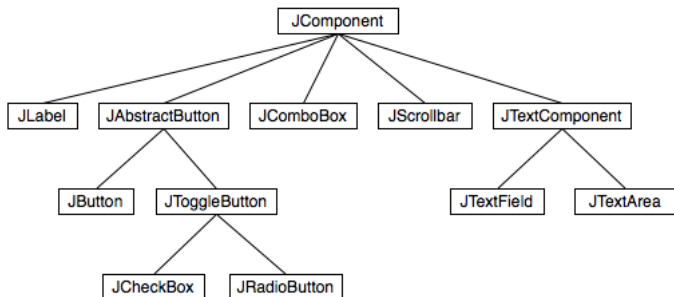
- command-line interface
 - terminál, konzola
 - užívateľ napíše príkaz
 - počítač zobrazí reakciu na príkaz
- Graphical User Interface (GUI)
 - počítač vykresľuje grafické komponenty
 - užívateľ interaguje s komponentami (myšou)

Štandardné grafické komponenty



- Java podporuje dve sady GUI komponentov:
 - Abstract Windowing Toolkit (AWT) - zastaralý
 - Swing - moderný
- Keď užívateľ interaguje s GUI komponentom, je vygenerovaná *udalosť* (event)
 - programátor vytvorí *event handler* - podprogram s inštrukciami ako má program reagovať na event

Java GUI komponenty sú implementované ako objekty



1 Úvod

- Strojový kód
- Asynchrónne udalosti, prerušenia
- Java Virtual Machine
- Základné bloky programu
- Objekty a objektovo orientované programovanie (OOP)
- Uživatelský interface
- Internet

- Java je vyvinutá pre internet
- *applet*
 - program v Jave prenášaný cez internet a zobrazený pomocou web prehliadača
 - môže poskytnúť komplexnú interakciu s používateľom
- vynikajúca podpora pre rôzne technológie
 - sieťové protokoly
 - kryptografia
 - kompresia dát
 - spracovanie zvuku
 - 3D zobrazovanie
 - ...
- podpora pre iné zariadenia než tradičné počítače
 - smartfóny, e-readers - Java ME (Mobile Edition)
 - multimediálne zariadenia
 - inteligentné domáce spotrebiče
 - ...

David J. Eck: Introduction to Programming Using Java, Sixth Edition (online: <http://math.hws.edu/javanotes/>)