

Otázka číslo 23

Jazyk Java

Java je objektově orientovaný jazyk, takže se program skládá ze vzájemně komunikujících objektů. Objekty spolu komunikují pouze přes své veřejné rozhraní - jeden objekt může využívat veřejného rozhraní objektu druhého. Programátor v jazyce Java definuje strukturu a chování těchto objektů.

Základní úlohou při programování v jazyce Java je tedy definice **struktury** a **chování** objektů. Struktura objektu je dána strukturou jeho **vnitřního stavu**, tedy množinou proměnných, jejichž životní cyklus je svázaný s životním cyklem **třídy** či její **instance**. Tyto proměnné se označují jako **atributy**. Chování objektu je dáno posloupnostmi příkazů, které jsou v něm obsaženy. Objekt v sobě obsahuje pojmenované posloupnosti příkazů (bloky) s parametry, které se označují jako **metody**. Každá metoda vykonává posloupnost příkazů a k dispozici má všechny **atributy** i **metody** daného objektu (včetně metod objektů vnořených v attributech) a své **parametry**, což jsou proměnné dodané externě.

Základní rysy jazyka

Protože byl jazyk Java od počátku zamýšlen jako multiplatformní, je to jazyk interpretovaný. Zdrojový kód se tedy nekompiluje do strojového kódu procesoru počítače, ale do instrukcí virtuálního zásobníkového procesoru JVM (Java virtual machine), tzv. **bytekódu**. Implementace JVM mohou být různé a existují pro všechny rozšířené platformy (Windows, Linux, Mac OS, FreeBSD, atd.). Stejný kód lze tedy spustit na všech JVM stejně, a to nezávisle na platformě. V některých speciálních případech samozřejmě lze narazit na drobné rozdíly, ale podpora Javy je vesměs velmi dobrá. V jazyce Java se programují i televize, set-top boky, ledničky, mobilní zařízení (Android) a tak dále.

Struktura programu

Program v jazyce Java se skládá ze **tříd**. Třída představuje schéma, podle kterého se vytváří její **instance**. Je to podobné, jako když se razítko otiskne na papír. Razítko je třída, zatímco jeho otisk na papíře je instance této třídy. Změnou razítka se změní i jeho další otisky, ale změnou otisků není nijak ovlivněno razítko. Z otisku také lze zjistit, jakým razítkem byl natištěn. Podobně lze i u instance zjistit, z jaké třídy pochází.

Spuštění celého programu začíná tak, že je na předem určeném objektu (tzv. main class) zavolána metoda s názvem main, které jsou předány parametry z příkazového řádku. Co objekt vykoná potom už je zcela v jeho režii (viz dále).

Instalace Netbeans IDE (netřeba se učit, je to návod pro vás)

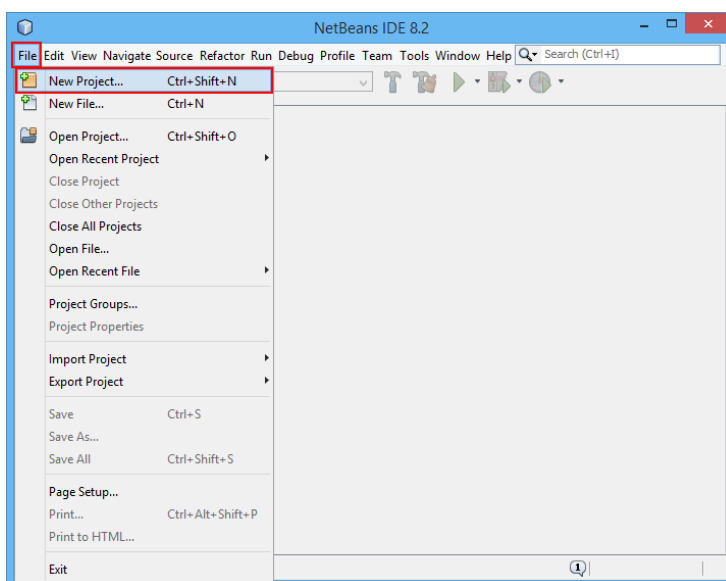
Nejprve si musíme stáhnout tzv. JDK (Java Development Kit), je to soubor základních nástrojů, které potřebujeme pro vývoj v Javě. Ten nalezneme na <http://www.oracle.com/...s/index.html>. Na stránce zaškrtneme, že souhlasíme s licencí, a vybereme JDK pro váš operační systém (nejčastěji Windows 32bit (označena jako x86) nebo 64bit, podle vaší verze operačního systému). Instalaci odklikáme (vynextíme), není třeba nic nastavovat. Je možné, že se s instalací spustí instalace dalších komponent, ty také odklikáme. Nyní stáhneme samotné IDE. IDE je zkratka Integrated Development Environment (integrováné vývojové prostředí) a jednoduše řečeno se jedná o aplikaci, ve které píšeme zdrojový kód a pomocí které potom naši aplikaci testujeme a ladíme. My budeme používat NetBeans, jelikož je asi nejrozšířenější a hlavně ho vyvíjí ta samá firma, která má na svědomí vývoj Javy samotné. Alternativou k NetBeans je ještě [Eclipse](http://netbeans.org/...s/index.html). Na adrese <http://netbeans.org/...s/index.html> stáhneme verzi pro vývoj v Java SE, která značí Java Standard Edition. Nainstalujeme (instalaci opět odklikáme) a spustíme.

Zálohování a verzování

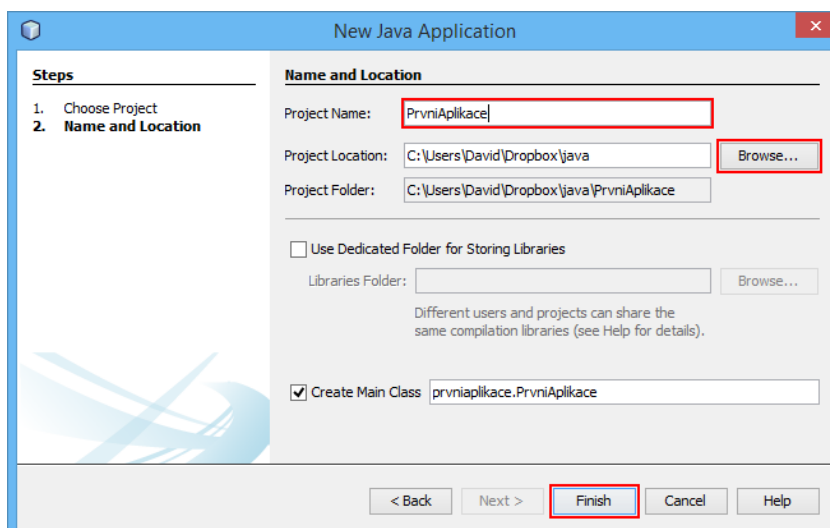
Kromě IDE programátor potřebuje nějaký nástroj, který bude zálohovat a verzovat jeho práci. Nemůžeme se spolehnout na to, že program prostě budeme ukládat, protože jsme lidé a ne stroje. Lidé dělají chyby, a když přijdete o několikadenní nebo dokonce několikatyždenní práci, může to zabolet. Je dobré naučit se na toto myslet hned od začátku. Velmi doporučuji program DropBox, který je extrémně jednoduchý a sám vaše soubory **verzuje** (tedy zachovává změny v čase a je možné se vrátit ke starším verzím projektu) a zároveň **synchronizuje** s webovým úložištěm, i kdybyste si projekt omylem smazali, přepsali, ukradli vám notebook nebo vám zkolaboval pevný disk, vaše data zůstanou v bezpečí. DropBox také umožňuje sdílet jeden projekt mezi více vývojáři.

Vytvoření projektu (toto se už učte)

Spustíme NetBeans a v aplikačním menu zvolíme položku File -> New Project.



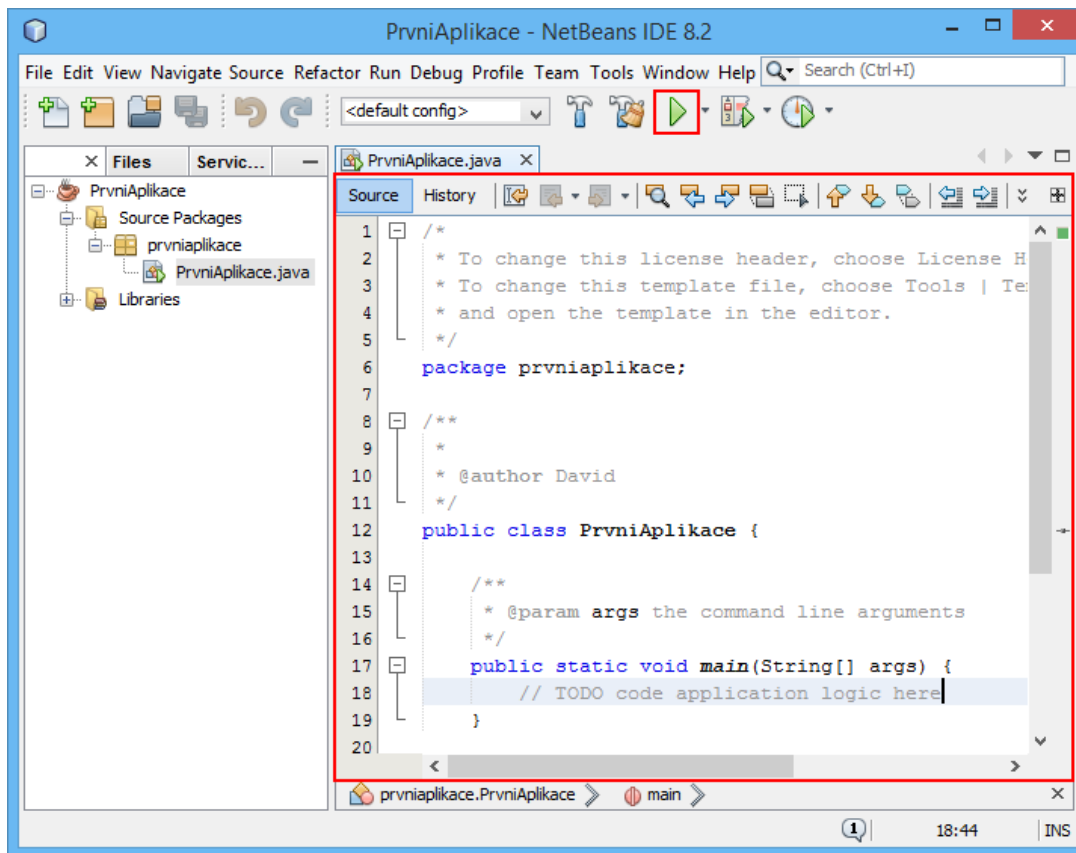
V okně New project vybereme z Java - Java Application. Jako jméno aplikace zvolíme PrvniAplikace. V Dropboxu si vytvoříte nějakou složku na vaše projekty, např. Java. U lokace pomocí tlačítka Browse vybereme složku C:\Users\vase_jmeno\Dropbox\Java. Nějakou dobu zůstaneme u konzolových aplikací (příkazová řádka), protože k jejich obsluze potřebujeme minimální znalosti z objektového světa a jsou tedy ideální k naučení základů jazyka. Okno by mělo vypadat asi takto:



Formulář potvrdíme.

Ovládání NetBeans

V NetBeans se nám založil nový projekt. Já jsem okno hodně zmenšil, aby se mi sem vešlo



Zajímat nás bude zejména prostřední okno, do kterého nám NetBeans vygeneroval kostru zdrojového kódu. Možná může být překvapením, že nezačínáme s prázdným oknem, ale rovnou s kusem kódu. Proč tomu tak je pochopíte, až si kód alespoň intuitivně vysvětlíme, vše bude vysvětleno během seriálu a některé části jsou na pochopení poměrně složité, proto nám zatím bude stačit vědět, že tam prostě jsou.

Package a class zatím nebudeme řešit, spokojíme se s tím, že je to určitý způsob, jak se aplikace v Javě strukturují. Klíčová pro nás bude metoda **main**, mezi ty složené závorky pod ní (tedy do jejího těla) budeme psát náš kód. Main je vyhrazené slovo a Java ví, že má po spuštění aplikace vykonat právě tuto metodu (může jich tam být totiž více, ale o tom opět později). Vlastně můžeme zatím ignorovat úplně všechno až na tělo metody main.

Druhým důležitým prvkem v okně pro nás bude zelené tlačítko Play v horní liště, které program zkompile a spustí. Můžete si to zkusit, protože náš program zatím nic nedělá, hned se zase vypne. Spuštění můžeme provést též klávesovou zkratkou F6.

Hello world

Je zarytým zvykem, že prvním programem v nějakém novém jazyce bývá tzv. Hello world. Jedná se o program, který jakýmkoli způsobem uživateli zobrazí hlášku "Hello world", případně nějaký podobný text. Opět zopakují, že příkazy budeme psát do těla metody main.

K výpisu textu slouží: `System.out.println("Text");`

System je tzv. **třída**. Pojmeme třída budeme zatím chápat soubor nějakých příkazů, příkazům se v Javě říká metody. **System** tedy obsahuje metody k obsluze vstupů a výstupů. Na výstupu (**out**) voláme metodu **println**, která vypíše text. Vidíme, že metodu na třídě voláme pomocí operátoru tečka. Každá metoda může obsahovat nějaké vstupní parametry, které se zadávají do závorky a jsou oddělené čárkou. V případě metody **println** je parametrem text k vypsání. Textu budeme říkat textový řetězec nebo jen řetězec (anglicky **string**) a budeme ho psát do uvozovek, aby tomu Java rozuměla a nezaměňovala ho s jinými příkazy. I kdyby metoda neměla žádné parametry, je závorka za ní povinná a byla by prázdná. Příkazy píšeme na samostatné řádky a za každý píšeme středník. Naše metoda **main** tedy bude nyní vypadat nějak takto:

[Spustit kód](#) Klikni pro editaci

```
• public static void main(String[] args)
• {
•     System.out.println("Hello ITnetwork!");
• }
•
•
```

Program spustíme pomocí klávesy F6.

PrvniAplikace Hello ITnetwork!