

Psaní programů v Javě

V této kapitole:

- Ve vývojovém prostředí NetBeans napíšete javový program
- Naučíte se uspořádat kód pomocí závorek
- Dozvíte se, jak se ukládají informace do proměnné
- Zobrazíte si informace, uložené v proměnné
- A nakonec svůj nový program uložíte, zkompilujete a spustíte

Pojmem *počítačový program* se obvykle označuje sada instrukcí říkajících počítači, co má dělat. Tyto instrukce jsou počítači předávány pomocí programovacího jazyka.

Během studia této kapitoly si vytvoříte jednoduchý program v jazyce Java, a to tak, že jeho kód zadáte do textového editoru. Poté program uložíte, zkompilujete a nakonec také otestujete. V další části kapitoly pak v programu úmyslně vytvoříte chybu, kterou vzápětí zase opravíte. To vše proto, abychom si ukázali potřebné postupy.

Co pro psaní programů potřebujete

Chcete-li vytvářet mody Minecraftu či jakékoliv jiné javové programy, musíte mít k dispozici nějaký programovací nástroj podporující sadu Java Development Kit (JDK). Typickým příkladem může být integrované vývojové prostředí NetBeans. Stručně řečeno, potřebujete nástroj obsahující nejen textový editor pro psaní programů, ale i funkce pro kompilování programů a jejich následné spouštění.

Lze říci, že v případě většiny programovacích jazyků jsou počítačové programy psány do nějakého textového editoru, jemuž se také někdy říká *editor zdrojového kódu*. Některé počítačové jazyky mají svoje vlastní editory. Součástí prostředí NetBeans je vlastní editor pro psaní javových programů.

Zdrojový kód javových programů je uložen do jednoduchých textových souborů, nemajících žádné speciální formátování, jako například text, zarovnaný na střed či tučné písmo. Editor zdrojového kódu prostředí NetBeans funguje jako jednoduchý textový editor, jehož součástí jsou však některá rozšíření, která jsou pro programátory velice užitečná. Například v průběhu psaní kódu jednotlivé části mění barvu, čímž editor zvýrazňuje různé elementy jazyka. Kro-

mě toho prostředí NetBeans využívá i správné odsazování řádek a uvnitř editoru vám nabízí užitečnou programátorskou dokumentaci.

Jelikož javové programy jsou textovými soubory, můžete je otevřít a upravit v libovolném textovém editoru. To znamená, že zdrojový kód můžete vytvořit v prostředí NetBeans, otevřít jej pomocí editoru Poznámkový blok či TextEdit a provést v něm potřebné změny, a nakonec jej bez jakýchkoliv potíží opět otevřít v prostředí NetBeans.

Vytvoření programu Splash

Jednou ze zvláštností Minecraftu je náhodná zpráva zobrazující se při spouštění klienta. V tuto chvíli máme na mysli onen žlutý text, zobrazující se v pravém rohu Minecraftu a překrývající jeho logo. Ukázku vidíte na obrázku 4.1.



Obrázek 4.1: Klient Minecraftu se zprávou zobrazovanou při spouštění

Zpráva na obrázku 4.1 zní „Finally with ladders!“. Mojang, což je společnost vyvíjející Minecraft, využívá okamžik spouštění programu k zobrazování různých zpráv, majících charakter rozličných vtipků, odkazů na klasické či nepříliš rozšířené videohry či sdělujících různé další neobvyklé věci. V některých případech tak šíří i informace týkající se vývoje Minecraftu, což je přesně tento případ. Pokud se podíváte na web, zjistíte, že hlášení „Finally with ladders!“ („Konečně se žebříky!“) je narážkou na dobu, kdy Notch, jeden z vývojářů Minecraftu, myšlenku žebříků ve hře neustále odmítal, nicméně zájem hráčů jej donutil ke změně názoru. Více informací najdete na adrese <http://minecraft.gamepedia.com/Ladders>.

První javový program, který si vytvoříte, bude zobrazovat vaši vlastní jednoduchou zprávu: „To nejsou obyčejné lžičky!“

Chcete-li začít pracovat na svém prvním programátorském projektu v prostředí NetBeans, pak pokud jste tak dosud neučinili, vytvořte si nový projekt, nazvaný Minecraft. Přitom postupujte podle následujících kroků:

1. Z nabídky File zvolte volbu New Project. Otevře se dialog New Project.
2. V levé části si vyberte kategorii projektů Java. Přejděte do pravé části dialogu a zde vyberte typ projektu Java Application. Poté klepněte na tlačítko Next.
3. Do pole Project Name zadejte Minecraft. (Pokud jste projekt s tímto názvem vytvořili již dříve, zobrazí se vám ve spodní části dialogu chybové hlášení Project folder already exists and is not empty.)
4. Zrušte zaškrtnutí volby Create Main Class.
5. Klepněte na tlačítko Finish.

Prostředí NetBeans vytvoří na pevném disku samostatnou složku, určenou pro projekt Minecraft. Tento projekt můžete používat i pro všechny další javové programy, které budete vytvářet během studia následujících kapitol – nejdéle však do kapitoly 17, nazvané „Čtení a zapisování souborů“. Poté již budete v prostředí NetBeans vytvářet pro každý nový mod Minecraftu samostatný projekt.

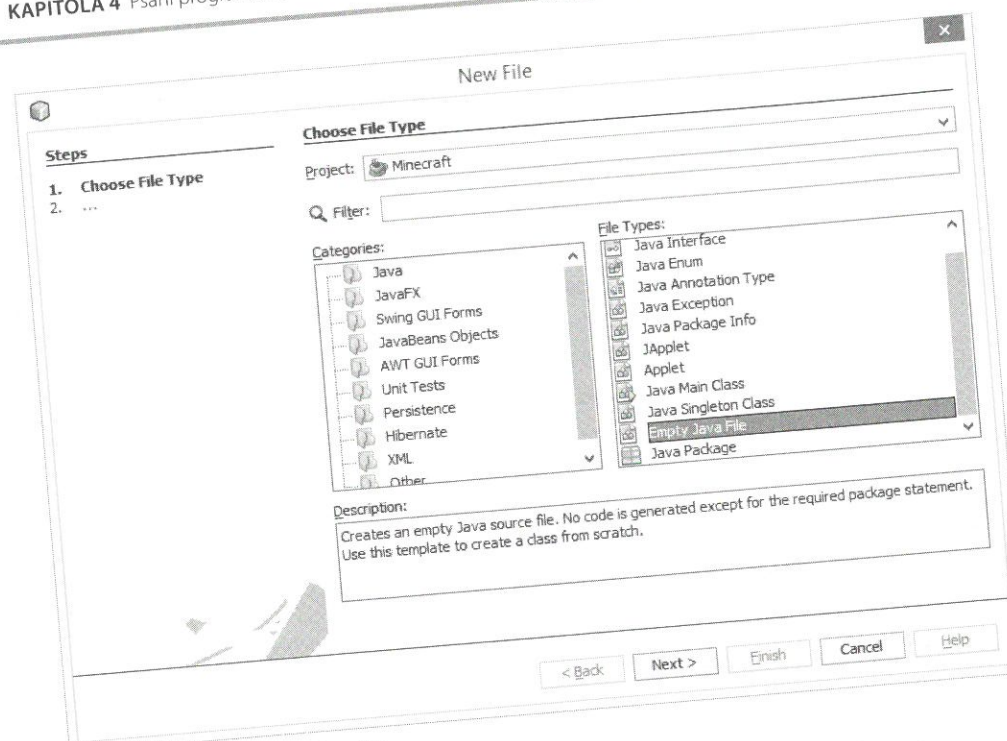
Zahájení práce na programu

Prostředí NetBeans seskupuje všechny vzájemně související programy do projektu. Nemáte-li dosud projekt Minecraft otevřený, zpřístupněte si jej takto:

1. Z nabídky File zvolte Open Project. Spustí se stejnojmenný dialog umožňující výběr souborů.
2. Je-li to nutné, vyhledejte složku, do níž jste nainstalovali server projektu Bukkit. Poté do této složky přejděte.
3. Tuto složku si otevřete.
4. Vyberte Minecraft a klepněte na tlačítko Open Project.

Projekt Minecraft se zobrazí v panelu Projects. Vlevo vedle něj bude zobrazena ikona šálku s kávou a znaménko plus, naznačující, že projekt lze rozbalit a zpřístupnit si tak jednotlivé soubory a složky tvořící součást projektu.

Chcete-li do právě otevřeného projektu přidat nový javový program, pak z nabídky File zvolte New File. Zobrazí se vám průvodce New File, jehož ukázku vidíte na obrázku 4.2.



Obrázek 4.2: Průvodce New File

V panelu Categories vidíte nejružnější druhy javových programů, které můžete vytvářet. V tomto panelu klepněte na složku Java. Díky tomu se vám v panelu File Types zobrazí všechny druhy souborů patřících do vybrané kategorie. V případě svého prvního javového projektu si zvolte Empty Java File a pokračujte klepnutím na tlačítko Next.

Přejdete tak do dialogu New Empty Java File. Chcete-li začít psát kód svého prvního programu, postupujte takto:

1. Do pole Class Name zadejte Splash.
2. Do pole Package zadejte com.javaminecraft.
3. Nakonec klepněte na tlačítko Finish.

Nyní již můžete začít pracovat na svém programu, neboť v okně editoru zdrojového kódu se vám otevřel prázdný soubor s názvem Splash.java. Svoji kariéru programátora v jazyce Java zahajte tím, že do tohoto okna zadáte veškerý kód uvedený ve výpise 4.1. Tyto příkazy jsou součástí toho, čemu se říká *zdrojový kód programu*.

Upozornění: Jak jsme vás upozornili již v předcházejících kapitolách, na začátky řádků nezadávejte jejich pořadová čísla následovaná dvojtečkami – ty jsou využívány v této knize pouze proto, abychom se na jednotlivé řádky mohli snadno odkazovat v následujícím textu.

Výpis 4.1: Zdrojový kód programu Splash

```
1: package com.javaminecraft;
2:
3: class Splash {
4:     public static void main(String[] arguments) {
5:         // Zde zadejte kod svého prvního programu
6:     }
7: }
```

Ujistěte se, že jste všechna malá a velká písmena zadali přesně podle našeho vzoru. Pro vložení mezer na začátky řádků 4–6 použijte mezerník či tabulátor. Po dokončení zvolte z nabídky File volbu Save a veškeré provedené změny tak uložte.

V tuto chvíli tedy soubor Splash.java obsahuje holou kostru javového programu.

Během studia dalších kapitol této knihy budete vytvářet mnoho dalších programů, jejichž základní kód bude – až na slovo Splash na třetím řádku – vypadat přesně takto. Toto slovo totiž představuje název vašeho programu, a proto se u každého programu mění. Obsah řádku 5 by vám měl být zřejmý: jedná se o komentář, v němž je česky napsáno, co máte dělat dále. Pomineme-li ukázkový mod Minecraftu, který jste napsali a otestovali v kapitole 3 „Vytvoření modu pro Minecraft“, je zbytek kódu pro vás pravděpodobně něčím novým.

Příkaz class

První řádek programu vypadá takto:

```
package com.javaminecraft;
```

Balíček (angl. package) představuje způsob seskupování javových programů. Tento řádek říká kompilátoru, že daný program má přiřadit k balíčku s názvem com.javaminecraft.

Za prázdným řádkem následuje třetí řádek, jehož obsah najdete níže:

```
class Splash {
```

Přeloženo do češtiny, tento řádek znamená něco: jako „Počítači, mému javovému programu přiřaď název Splash.“

Z kapitoly 3 si možná ještě pamatujete, že každá instrukce, kterou počítači předáte, se nazývá *příkazem*. Klíčové slovo class pak představuje způsob, jímž můžete svým javovým programům přiřazovat nějaký název. Jak však uvidíte později, toto klíčové slovo se využívá i k určení dalších věcí týkajících se daného programu. Právě díky významu pojmu class se javovým programům mnohdy říká *třídy*.

V našem příkladu platí, že název programu Splash odpovídá názvu souboru, do něž je zdrojový kód uložen, tj. souboru Splash.java. Obecné pravidlo totiž zní, že název javového programu musí odpovídat první části názvu toho souboru, v němž je zdrojový kód uložen, přičemž při zadávání názvu musí být použita i shodná velká či malá písmena.

Neshoduje-li se název programu s první částí názvu souboru, zobrazí se vám při kompilaci některých javových programů chybové hlášení, závislé na tom, jakým způsobem je příkaz `class` využit ke konfiguraci programu.

Co dělá příkaz `main`

Dostáváme se k dalšímu řádku zdrojového kódu, majícího tuto podobu:

```
public static void main(String[] arguments) {
```

Tento řádek říká počítači zhruba toto: „Hlavní část programu začíná právě zde.“ Javové programy bývají uspořádány do různých částí, a proto musí existovat nějaký způsob identifikace té části programu, která má být po spuštění programu provedena nejdříve.

Příkaz `main` je vstupním bodem většiny javových programů. Mody Minecraftu však představují výjimku z tohoto pravidla, neboť jsou prováděny serverem projektu Bukkit, a nelze je tedy spouštět samostatně. Zadá-li hráč nějaký příkaz podporovaný daným modem, server spustí jeho kód.

K dalším výjimkám z tohoto pravidla patří například aplety, což jsou programy, prováděné webovým prohlížečem na webových stránkách; servlety, spouštěné webovým serverem; či aplikace určené pro spuštění na mobilních zařízeních.

Všechny javové programy, které napíšete v rámci studia následujících 13 kapitol, budou využívat příkaz `main` jako svůj výchozí bod. To proto, že je budete na svém počítači spouštět přímo. Mody, aplety, servlety či aplikace pro mobilní zařízení jsou spouštěny nepřímo, a to buď jiným programem či zařízením.

A aby bylo možné snadno odlišit programy obsahující příkaz `main` od zmíněných ostatních typů programů, nazývají se tyto programy *aplikacemi*.

Ty zakroucené závorky

Na řádcích 3, 4, 6 a 7 programu `Splash` vidíte zakroucené závorky nějakého druhu – buď `{` anebo `}`. Tyto závorky umožňují seskupování řádek programu (podobným způsobem, jímž jsou kulaté závorky používány ve větách k seskupení slov). Vše, co se nachází mezi otevírací složenou závorkou `{` a uzavírací složenou závorkou `}`, je součástí jedné skupiny.

Tyto skupiny se nazývají *bloky*. Ve výpise 4.1 otevírací složená závorka na řádku 3 souvisí s uzavírací složenou závorkou na řádku 7, díky čemuž se z celého vašeho programu stává jeden blok. Tímto způsobem se složené závorky využívají k označení začátku a konce programu.

Bloky mohou být umístěny uvnitř dalších bloků (podobně jako závorky jsou použity v této větě (příčímž zde začíná druhá sada závorek)). Na řádcích 4 a 6 programu `Splash` vidíte složené závorky vytvářející další blok. Tento blok začíná příkazem `main`. Řádky nacházející se uvnitř bloku příkazu `main` budou provedeny ihned po spuštění programu.



Tip: Prostředí NetBeans vám může pomoci se zjištěním, kde blok vlastně začíná a kde končí. Ve zdrojovém kódu programu `Splash` klepněte na jednu ze složených závorek. Prostředí NetBeans zvýrazní žlutou barvou nejen tuto vybranou složenou závorku, ale i jí odpovídající párovou závorku. Všechny příkazy jazyka Java uzavřené mezi dvě žlutě zvýrazněné složené závorky představují jeden blok kódu. U krátkých programů, k nimž patří i námi připravovaný program `Splash`, není tento tip nijak zvlášť užitečný, nicméně jakmile začnete psát delší programy, tuto funkci prostředí NetBeans zcela jistě oceníte.

Následující příkaz je jediným umístěným uvnitř námi popisovaného bloku:

```
// Zde zadejte kod svého prvního programu
```

Tento řádek je pouze zástupným řádkem. Znaky `//` na začátku řádku totiž říkají počítači, aby celý řádek ignoroval, neboť byl do kódu umístěn pouze pro další osoby čtoucí zdrojový kód. Řádky sloužící tomuto účelu se nazývají *komentáři*.

V tuto chvíli máte dokončený celý javový program. To znamená, že program lze zkompileovat, nicméně po jeho spuštění se vůbec nic nestane. To proto, že doposud jste počítači neřekli, co má vlastně dělat. Blok příkazu `main` totiž obsahuje pouze jeden komentář, který bude kompilátorem ignorován. Pokud tedy chcete, aby program něco udělal, musíte mezi otevírací a uzavírací složenou závorku bloku `main` přidat nějaké další příkazy.



Poznámka: Na konec každého příkazu zapsaného v jazyce Java je nutné přidat středník(`;`), nicméně v programu `Splash` řádek `// Zde zadejte kod svého prvního programu` středníkem ukončen není.

Důvodem je to, že komentáře jsou kompilátorem zcela ignorovány. Vložíte-li tedy znaky `//` do kteréhokoliv řádku svého programu, řeknete vlastně kompilátoru Javy, aby ignoroval vše, co se na daném řádku nachází napravo od znaků `//`. Následující příklad ukazuje komentář nacházející se na tomtéž řádku jako příkaz:

```
System.out.println(pozdrav); // Zobrazení pozdravu
```

Ukládání informací do proměnných

Při psaní svých dalších programů zcela jistě narazíte na to, že po určitou dobu budete potřebovat nějaké místo, do něž budete moci ukládat informace. K tomuto účelu můžete využívat proměnné, což jsou vlastně úložiště, do nichž můžete ukládat celá čísla, čísla s plovoucí desetinnou čárkou, logické hodnoty typu pravda – nepravda, znaky či celé řádky textu. Informace uložené do proměnné lze měnit, a proto se tato úložiště dat nazývají *proměnnými*.

V souboru `Splash.java` nahraďte řádek 5 následujícím příkazem:

```
String pozdrav = "To nejsou obyčejné lžičky!";
```

Tento příkaz říká počítači, že má uložit text `To nejsou obyčejné lžičky!` do proměnné s názvem `pozdrav`.

V javovém programu musíte kompilátoru říci, jaký typ informací bude do proměnné ukládán. V našem programu je proměnná pozdrav definována jako *řetězec* (angl. *string*) – tj. řádek textu, jehož součástí mohou být číslice, písmena, interpunkční znaménka a další znaky. Tím, že na začátek příkazu vložíte `String`, nastavíte proměnnou tak, aby do ní bylo možné ukládat řetězcové hodnoty.

Vložíte-li tento příkaz do programu, musíte na jeho konec přidat středník. Jak jsme si řekli již před chvílí, na konci každého příkazu javového programu se musí nacházet středník. Z tohoto pohledu lze říci, že se středníky podobají tečkám na koncích vět. Kompilátor využívá středníky k tomu, aby zjistil, kde jeden příkaz končí a druhý začíná.

Budete-li na každý řádek kódu zadávat vždy pouze jeden příkaz, budou vaše zdrojové kódy čitelnější (přinejmenším tedy pro nás, lidské bytosti).

Zobrazení obsahu proměnné

Pokud v tuto chvíli upravený program spustíte, bude se vám i nadále zdát, že se vlastně nic neděje. Příkaz pro uložení textu do proměnné pozdrav totiž proběhne v pozadí. Chcete-li počítač donutit k tomu, aby vám ukázal, že skutečně něco dělá, můžete si nechat zobrazit obsah této nové proměnné.

Do programu `Splash` vložte za příkaz `String pozdrav = "To nejsou obyčejné lžičky!"` další prázdný řádek. Tento volný prostor pak využijte k zadání následujícího příkazu:

```
System.out.println(pozdrav);
```

Tento příkaz říká počítači, aby zobrazil hodnotu uloženou v proměnné `pozdrav`. K tomuto účelu využíváme příkaz `System.out.println`, vedoucí k zobrazení informací na výstupním zařízení systému – tedy na vašem monitoru.

Nyní jsme již někam pokročili.

Uložení dokončeného programu

Kód vašeho programu by nyní měl odpovídat kódu uvedenému ve výpise 4.2, byť je možné, že na řádcích 5–6 jste použili poněkud jiné odsazení. Dokončete všechny nezbytné úpravy a výsledný kód uložte výběrem volby `Save` z nabídky `File`.

Výpis 4.2: Dokončená a hotová verze programu `Splash`

```
1: package com.javaminecraft;
2:
3: class Splash {
4:     public static void main(String[] arguments) {
5:         String pozdrav = "To nejsou obyčejné lžičky!";
6:         System.out.println(pozdrav);
7:     }
8: }
```

Po spuštění tohoto programu počítač provede každý z řádků bloku `main`, tj. v tomto případě řádky 5 a 6. Ve výpise 4.3 vidíte, jak by kód programu vypadal, kdyby byl zapsán v běžné češtině, a nikoliv v Javě.

Výpis 4.3: Analýza programu `Splash` řádek po řádku

```
1: Přidej tento program do balíčku com.javaminecraft
2:
3: Zde začíná program Splash:
4:     Hlavní část programu začíná zde:
5:         Ulož text "To nejsou obyčejné lžičky"
           do proměnné pozdrav typu String
6:         Zobraz obsah proměnné pozdrav
7:     Hlavní část programu končí zde.
8: Program Splash končí zde.
```

Kompilace programu do souboru třídy

Než spustíte kterýkoliv javový program, musíte jej zkompileovat. Zjednodušeně řečeno, během kompilace programu jsou instrukce, které v programu předáváte počítači, převáděny do podoby, již je počítač schopen snáze porozumět.

Prostředí `NetBeans` kompiluje programy automaticky během jejich ukládání. Pokud jste zdrojový kód zapsali přesně tak, jak je uveden ve výpise 4.2, pak kompilace zcela jistě proběhla úspěšně.

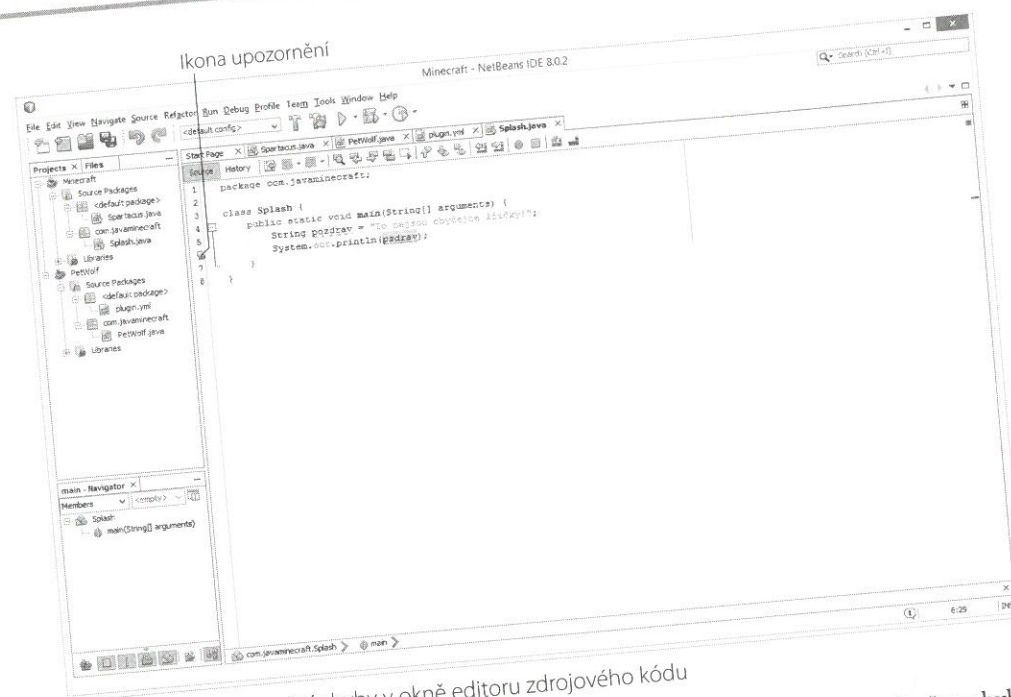
Výsledkem kompilace je zkompileovaná verze programu, uložená do souboru s názvem `Splash.class`. Platí, že všechny javové programy se kompilují do souborů tříd, majících příponu `.class`. Přitom jeden javový program může být tvořen několika vzájemně spolupracujícími třídami, nicméně v tak jednoduchých programech, jakým program `Splash` bezesporu je, je potřebná pouze jedna třída.

Z hlediska technického kompilátor mění zdrojový kód v jazyce Java na tzv. *bajtkód*, který lze spustit v prostředí virtuálního stroje Javy (*Java Virtual Machine – JVM*).

Poznámka: Kompilátor Javy se „ozve“ pouze tehdy, narazí-li na nějakou chybu, na kterou si chce postěžovat. Podaří-li se vám zkompileovat program úspěšně, bez jakýchkoliv chyb, nic se nestane. To není příliš uspokojivé. Když jsem začínal svoji kariéru programátora pracujícího v jazyce Java, doufal jsem, že každá úspěšná kompilace bude odměněna fanfárami či něčím podobným.

Opravy chyb

Z obrázku 4.3 je zřejmé, že pokud při psaní zdrojového kódu v editoru prostředí `NetBeans` uděláte nějakou chybu, zobrazí se v levém krajním sloupci příslušného řádku červená ikona upozornění.



Obrázek 4.3: Zvýraznění chyby v okně editoru zdrojového kódu

Jak jsme si již řekli, tato ikona se objeví v levém krajním sloupci toho řádku, v němž se chyba vyskytuje. Podržíte-li nad touto ikonou ukazatel myši, zobrazí se chybové hlášení obsahující tyto podrobnosti:

- Název javového programu
- Typ chyby
- Řádek, v němž byla chyba nalezena

Níže vidíte ukázkou chybového hlášení, které by se vám mohlo zobrazit, pokud byste v programu `Splash` udělali stejnou chybu jako my:

```
cannot find symbol
symbol: variable pzdrav
location: class Splash
```

Typ chyby je určen hned prvním řádkem uvedeného hlášení: `cannot find symbol`. Pro začínající programátory však tato hlášení mohou být často matoucí. Není-li pro vás zobrazené chybové hlášení dostatečně srozumitelné, neztrácejte příliš mnoho času tím, že se budete snažit zjistit jeho přesný význam. Namísto toho se zaměřte na ten řádek kódu, vedle nějž se chyba zobrazila, a zkuste v něm najít a případně opravit některé často opakované chyby.

Dokážete například na první pohled říci, co je špatného na následujícím příkazu?

```
System.out.println(pzdrav);
```

Chyba je způsobena překlepem v názvu proměnné, neboť ten by správně měl znít `pozdrav`, a nikoliv `pzdrav`. (Zkuste si tuto chybu úmyslně zopakovat i na svém počítači a uvidíte, co se bude dít.)

Zobrazí-li se vám během vytváření programu `Splash` jakékoliv chybové hlášení, ujistěte se, že vámi zadávaný kód skutečně odpovídá vzoru uvedenému ve výpise 4.2. Pokud najdete nějaké rozdíly, opravte je. Přesvědčte se také, že všechna malá a velká písmena jsou zadána v souladu s naším vzorem a že jste nezapomněli na žádné z interpunkčních znamének, jako například `{, }` či `;`.

Velmi často k odhalení chyby (či chyb) stačí pozorné prohlédnutí toho řádku kódu, u nějž je chybovým hlášením indikován výskyt chyby.

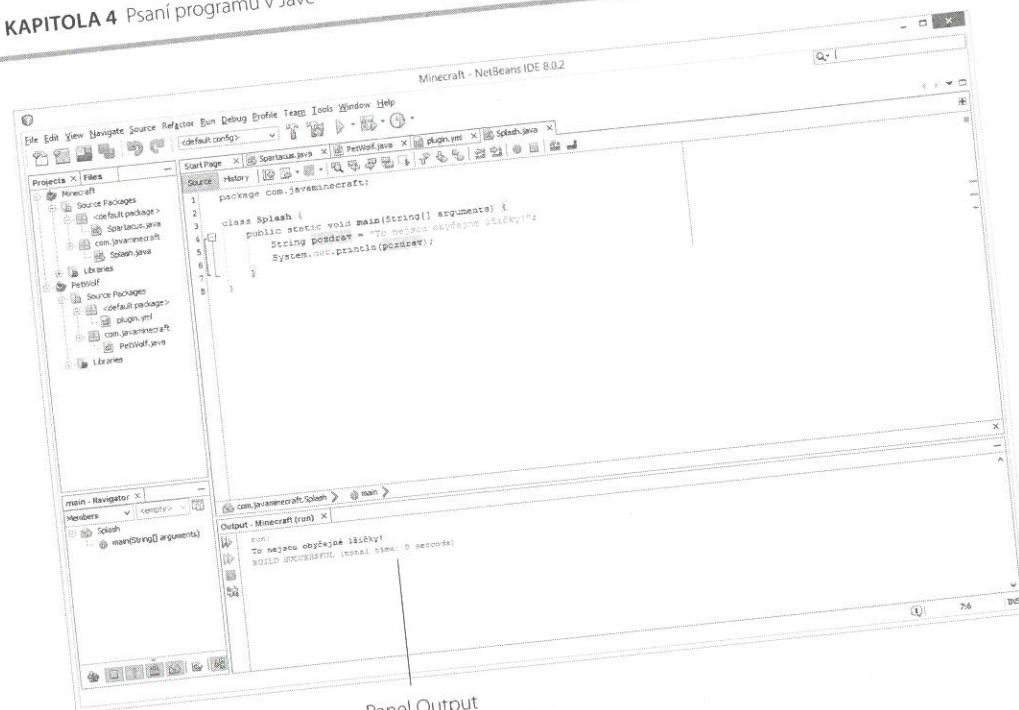
Nezapomeňte ale, že ten řádek, v němž je chybové hlášení zobrazeno, nemusí vždy odpovídat tomu řádku, který vyžaduje opravu. Proto si také důkladně prohlédněte i všechny příkazy předcházející řádku s chybovým hlášením a ujistěte se, zda se chyba náhodou nenachází v nich. Obvykle však platí, že příčina chyby se nachází v tom samém bloku kódu.



Tip: Na oficiálním webu této knihy, majícím adresu www.javaminecraft.com, najdete zdrojový kód všech programů, které budete v této knize vytvářet, v původní, tj. anglické, verzi. Pokud v programu `Splash` nemůžete najít žádné překlepy či jiné příčiny chyby, a přesto je v programu nadále nějaká chyba indikována, navštivte web této knihy a ze stránky kapitoly 4 si stáhněte původní verzi programu `Splash.java`. Poté zkuste spustit tuto verzi namísto té své.

Spuštění javového programu

Chcete-li se ujistit, zda program `Splash` dělá opravdu to, co jste chtěli, spusťte vytvořený soubor třídy pomocí virtuálního stroje JVM, což je interpreter, v němž je možné spouštět veškerý javový kód. V případě prostředí NetBeans přejděte do nabídky `Run` a poté zvolte `Run File`. Pod oknem editoru zdrojového kódu se otevře panel `Output`. Z obrázku 4.4 je patrné, že nejsou-li v kódu žádné chyby, pak se v tomto panelu zobrazí požadovaný výstup.



Obrázek 4.4: Spuštění javového programu Splash

Uvidíte-li v tomto panelu text *To nejsou obyčejné šišky!*, pak jste svůj první javový program nejen dobře napsali, ale i zkompilovali a spustili.

Poznámka: Námi použité hlášení pochází z českého filmu „Pelišky“. Pokud se však týká hry Minecraft, při jejím spuštění jsou zobrazována hlášení, která jsou náhodně vybrána z textového souboru `splashes.txt`, tvořícího součást balíčku **Minecraft.jar**. Zajímá-li vás význam jednotlivých hlášení, podívejte se na web <http://minecraft.gamepedia.com/Splash>.

Prázdné znaky a bílé znaky v javových programech

Při psaní programu `Splash`, jehož kód vidíte ve výpise 4.2, jste si dávali pozor na to, abyste před každým příkazem zadali správný počet mezer, a zajistili tak správné odsazení a zarovnání jednotlivých řádků. Možná jste se přitom ptali, zdali je to skutečně tak důležité.

Pokud se týká samotného kompilátoru (a počítače), prázdné znaky a bílé znaky jsou naprosto nedůležité. Odsazení je důležité pouze pro další lidi prohlížející si zdrojový kód vašeho programu – kompilátor Javy odsazení prostě ignoruje. Program `Splash` byste mohli klidně napsat i bez použití jakýchkoliv mezer či znaků tabulátoru k odsazení jednotlivých řádků, a přesto byste dokázali program úspěšně zkompilovat.

Byť počet mezer na začátku jednotlivých řádků není důležitý, měli byste ve svých javových programech používat konzistentní počty mezer či konzistentní odsazování řádků. Proč? Protože

díky odsazení jste schopni rychleji a snáze rozpoznat nejen celkové uspořádání programu, ale i to, ke kterému bloku programu daný příkaz patří.

Programy, které píšete, by měly být srozumitelné i ostatním programátorům, včetně vás samotných, a to zejména ve chvíli, kdy se ke zdrojovému kódu nějakého programu vrátíte o týden či měsíce později a chcete v něm opravit nějakou chybu či přidat rozšíření. Konzistentnost v používání mezer a v odsazování řádků je součástí toho, čemu se říká *styl programování*. Dobří programátoři si vytvoří svůj vlastní styl a používají jej ve veškeré své práci.

Absolutní základ

V průběhu studia této kapitoly jste se seznámili se všemi prvky typického javového programu. Dozvěděli jste se, že chcete-li vytvořit nějaký program, musíte provést tyto tři základní kroky:

1. V textovém editoru či pomocí nějakého nástroje, jako například NetBeans, napsat zdrojový kód programu.
2. Zkompilovat zdrojový kód do souboru třídy.
3. Spustit třídu pomocí virtuálního stroje JVM.

Současně jste se seznámili i s některými základními koncepty, souvisejícími s programováním, jako například kompilátory, interprety, bloky kódu, příkazy a proměnnými. Význam těchto konceptů se pro vás stane zřejmější během čtení následujících kapitol. Pokud se vám v závěru této kapitoly podařilo spustit program `Splash`, jste připraveni pokračovat.