

11 Vytvoření instalačního programu pomocí NSIS

V této kapitole si ukážeme, jak ke své aplikaci vytvořit instalační balíček pro Windows; k tomu využijeme volně dostupný nástroj NSIS.

V prvních oddílech této kapitoly si zjednodušíme situaci a budeme předpokládat, že na cílovém počítači je instalováno běhové prostředí Javy 8 a cesta k odpovídajícímu adresáři je uložena v systémové proměnné `PATH`; tyto oddíly slouží především k seznámení s prostředím a jazykem NSIS. Polední dva oddíly jsou věnovány vytvoření plnohodnotného instalačního programu bez instalace Javy a s ní.

Proč instalační balíček?

Ač se aplikace Fraktálník, kterou jsme v tomto skriptu vyvíjeli, se již značně rozrostla (obsahuje cca 40 tříd, tedy souborů `.class`, a vedle toho soubory ikon, nápovědy atd.), stále ji umíme „zabalit“ do dvou archivů JAR (ve skutečnosti do jednoho, neboť soubor JAR může obsahovat jiný soubor JAR). Proč si tedy komplikovat život instalačním balíčkem?

Kdybychom uživatelům dodali prostě jen dva archivy JAR na nějakém paměťovém médiu a řekli jim „zkopírujte si je někam“, nebo jim dali k dispozici webovou adresu, aby si je stáhli, určitě by třetina z nich něco pokazila. Co je ale horšího: Použitelnost naší aplikace závisí na tom, zda je na cílovém počítači instalováno běhové prostředí Javy 8, a to uživatel, který není programátor, nemusí vědět. Instalátor by tedy měl zjistit, zda je na počítači odpovídající verze Javy, a v případě potřeby ji instalovat.

Vedle toho jsou uživatelé Windows zvyklí, že se instalátor zeptá, kam má program instalovat, že pro něj vytvoří položku v nabídce *Start*, ikonu na ploše, že uživatele seznámí s licenčními podmínkami, že poskytne nástroj pro „automatizované“ odstranění – odinstalaci – programu atd. I když se často tváří, že je tyto věci obtěžují, jestliže je náš produkt neposkytne, budou mít pocit, že jsme jako dodavatelé neodvedli pořádnou práci.

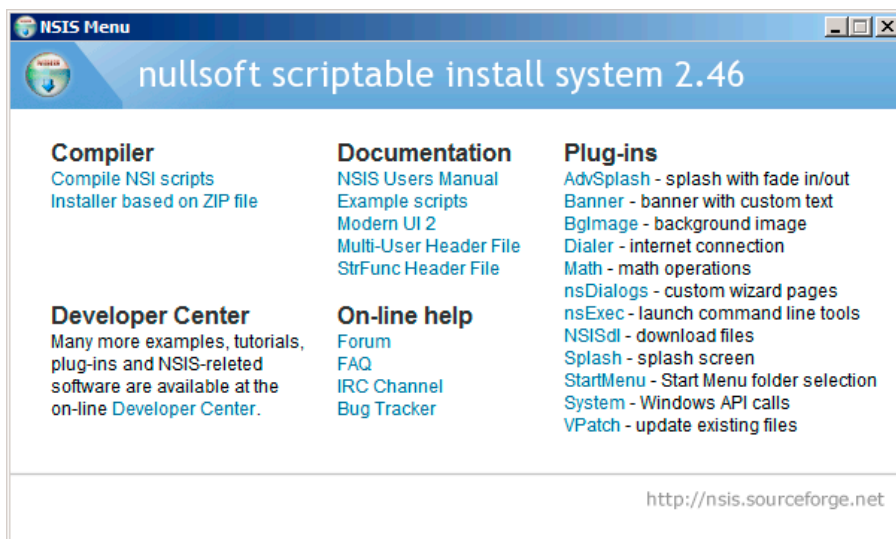
11.1 NSIS

K tomu, abychom mohli v Javě napsat program, který se postará o instalaci jiného programu, nám bohužel chybí řada speciálních znalostí. Při vytváření průvodce instalací totiž musíme umět mimo jiné vytvářet a odstraňovat zástupce souborů, zapisovat do registru Windows atd. Pod Windows Vista a Windows 7 nebo 8 také musíme umět získat oprávnění k zásahům do adresářů spravovaných operačním systémem. Jistě by bylo užitečné se naučit, jak se to vše dělá v Javě, ale z hlediska našeho cíle, vytvoření instalačního balíčku, je to cesta zbytečně složitá, neboť na trhu je k dispozici několik nástrojů, které nám mohou práci usnadnit. Jedním z nich je

např. Microsoft Installer, který lze získat na stránkách firmy Microsoft. Jiným velmi oblíbeným nástrojem je NSIS, se kterým se zde seznámíme. Nejprve si povíme nezbytné minimum informací o programu NSIS a o skriptování v něm a pak vytvoříme jednoduchý instalační program k naší aplikaci Fraktálník. Zatím se nebudeme snažit vytvořit průvodce, který by uživateli umožňovat seznámit se s licenčním ujednáním, určit cílový adresář, volit instalované součásti apod.; k tomu se vrátíme v oddílu 11.3.

11.1.1 Co je NSIS

Název NSIS je zkratkou slov *Nullsoft Scriptable Install System*, tedy *Skriptovatelný instalační systém firmy Nullsoft*. Je to systém s otevřeným zdrojovým kódem (open source), který umožňuje vytvořit instalační program („balíček“) pro Windows. Instalační program vytvořený pomocí NSIS je vše, co musíme dodat uživateli, který si chce instalovat naši aplikaci; všechny potřebné soubory jsou v něm uloženy jako data, která zapíše na cílový počítač. Program NSIS si můžeme bezplatně stáhnout ze stránek firmy Nullsoft [28]. Jeho instalace je velice jednoduchá. Po odsouhlasení licenčního ujednání nám původce umožní vybrat si adresář, do kterého chceme NSIS instalovat, a postará se o vše potřebné. Když program NSIS spustíme, objeví se hlavní okno aplikace, které ukazuje obr. 11.1; v něm najdeme základní nabídku této aplikace v podobě webových odkazů.



Obrázek 11.1: Hlavní okno a základní nabídka programu NSIS

V levé části okna najdeme příkazy pro překlad skriptů; v prostřední části nahoře jsou příkazy, kterými můžeme otevřít dokumentaci nebo vzorové skripty. Pod nimi jsou odkazy na webové fórum k tomuto programu, odpovědi na často kladené otázky (FAQ) a další. V pravé části okna jsou odkazy na doplňky (plug-in), které lze k tomuto programu doinstalovat.

Dokumentace

Dokumentaci k NSIS otevřeme pomocí odkazu v hlavním okně tohoto programu. První tři kapitoly popisují samotný program NSIS, čtvrtá popisuje skriptovací jazyk, který se používá

k vytváření instalačních programů. Další kapitoly popisují různé pokročilejší možnosti. Poznamenejme, že dokumentace k takzvanému modernímu uživatelskému rozhraní verze 2 má v úvodním okně NSIS samostatný odkaz *Modern UI 2*.

Ukázkové skripty

Jako začínající uživatelé oceníme sadu ukázkových skriptů, které jsou součástí instalace. Místo abychom psali své skripty od začátku, vyjdeme od některého z nich a jednoduchými úpravami ho přizpůsobíme svým potřebám. Často tak dosáhneme velmi pěkných výsledků, aniž bychom se museli prokousávat podrobnostmi skriptovacího jazyka NSIS.

Skriptování

Pomocí NSIS lze vytvořit instalační programy pro Windows, které budou umět nejen instalovat a odinstalovat aplikaci, kterou vytvoříme, ale také definovat potřebná nastavení v registru Windows a řadu dalších věcí. K tomu NSIS používá vlastní skriptovací jazyk. To je programovací jazyk zaměřený v tomto případě na vytváření instalačních balíčků. Díky tomu, že je jednoúčelový, je i velmi jednoduchý.¹ Na program v tomto skriptovacím jazyce se můžeme dívat jako na posloupnost volání funkcí, které se postarají o potřebné operace. Překladem skriptu vznikne instalační program.

Základní struktura skriptu

Skript pro NSIS je obyčejný textový soubor s příponou `.nsi`. Každý skript se skládá ze dvou základních částí — z tzv. *atributů*, které určují chování a vzhled instalačního programu, a ze *sekcí*, které popisují, co a jak je třeba instalovat nebo odinstalovat a další věci, které má instalační program provést. Nejprve zapisujeme atributy, pak ve skriptu následují sekce.

Sekce obvykle odpovídají za instalaci jednotlivých skupin souborů, tedy jednotlivých částí aplikace. Můžeme mít například jednu sekci pro instalaci souborů obsahujících program a základní dynamické knihovny a druhou sekci pro instalaci nápovědy. Vedle toho lze ve skriptech definovat i tzv. funkce; to jsou podprogramy, které bychom v Javě označili jako metody typu `void`. Podobně jako metody v Javě slouží k tomu, abychom nemuseli opakovaně programovat stejné operace. S funkcemi v NSIS se setkáme v posledním oddílu této kapitoly.

Sekce

Sekce se provádějí v pořadí, ve kterém je zapíšeme v instalačním skriptu. Každý skript musí obsahovat alespoň jednu sekci. Sekce začíná příkazem `Section` a končí příkazem `EndSection`. Za příkazem `Section` může následovat řetězec obsahující jméno sekce. Pak může následovat ještě identifikátor představující tzv. index sekce; s indexem se znovu setkáme v oddílu 11.3.2 při určování řetězců popisujících sekci v okně průvodce. Povězme si několik základních informací o jménech sekce:

- Jestliže se sekce jmenuje `'Uninstal'` nebo jestliže jméno sekce začíná `'un.'`, jde o sekci popisující program pro odinstalování aplikace.

¹Ve skutečnosti jde o beztypový nízkourovňový zásobníkový programovací jazyk (podle manuálu „někde mezi assemblerem a PHP“) vybavený nástroji pro speciální manipulace, a my budeme využívat především těchto speciálních nástrojů. Proto se nám bude ve většině této kapitoly *jevit* jako jednoduchý.

- Před jménem sekce může být volba /o, která říká, že tato sekce se implicitně nebude provádět, pokud to uživatel v odpovídajícím okně průvodce nezmění.
- Jestliže jméno sekce začíná znakem !, zobrazí se tato sekce v odpovídajícím okně průvodce tučně. Jestliže jméno sekce vynecháme nebo jestliže začíná znakem minus, jde o skrytou sekci, která se v uživatelském rozhraní vytvořeného průvodce nezobrazí.

Zápis skriptu

Na rozdíl od Javy neoddělujeme jednotlivé příkazy středníky, ale přechodem na novou řádku. To znamená, že každý příkaz musíme zapsat na samostatnou řádku; jestliže se nám tam nevejde, ukončíme řádku znakem \, za kterým přejdeme na novou řádku. Překladač skriptů si spojí takovouto řádku s řádkou následující. Příklady uvidíme dále.

Komentář

Máme několik možností, jak do skriptu zapsat komentář:

- Komentář může začínat znakem ; (středník) a končit na konci řádky.
- Komentář může začínat znakem # (mříž) a končit na konci řádky.
- Komentář může začínat znaky /* a končit znaky */. Takovýto komentář může zabírat i několik řádek.

Implicitní cílový adresář instalace

První věcí, kterou musíme v instalačním skriptu udělat, je určit cílový adresář, do kterého budou uloženy instalované soubory (domovský adresář aplikace). K tomu slouží atribut `InstallDir`, který musíme uvést před první sekci. Za slovem `InstallDir` uvedeme řetězec obsahující implicitní cestu, do které se bude aplikace instalovat. Poznamenejme, že znakové řetězce uzavíráme do uvozovek nebo do apostrofů. Jestliže například chceme použít adresář `C:\TEMP\Fraktál`, napíšeme

```
InstallDir "C:\TEMP\Fraktál"
```

nebo

```
InstallDir 'C:\TEMP\Fraktál'
```

V atributu `InstallDir` se můžeme odvolat i na některé systémové adresáře; k tomu slouží předdefinované proměnné. Jestliže bychom např. chtěli svou aplikaci instalovat do adresáře `Program Files\Fraktál`, napsali bychom

```
InstallDir "$PROGRAMFILES\Fraktál"
```

`$PROGRAMFILES` je předdefinovaná proměnná obsahující cestu k uvedenému adresáři. Tato cesta se může v různých instalacích Windows lišit, proto si ji instalační program zjistí ze systémového registru cílového počítače.

Poznámka

Další předdefinované proměnné jsou např. `$WINDIR`, jež určuje adresář s instalací Windows, `$SYSDIR`, určující adresář `Windows\System` nebo `Windows\System32`, `$TEMP` určující systémový

adresář pro dočasné soubory (obvykle `C:\Windows\Temp`) a další — podrobnosti lze najít v dokumentaci. Skutečné cesty zjišťuje instalační program za běhu na cílovém počítači. Poznamenejme, že v dokumentaci jsou tyto proměnné označovány jako konstanty, protože bychom neměli měnit jejich obsah.

Aktuální cílový adresář

Příkaz `SetOutPath` určuje adresář, do kterého se mají instalované soubory nakopírovat. Za slovem `SetOutPath` následuje specifikace adresáře; v ní se můžeme odvolat na proměnnou `$INSTDIR`, jejíž hodnotu jsme nastavili pomocí atributu `InstallDir`. Například příkazem

```
SetOutPath "$INSTDIR\data"
```

přikazujeme kopírovat soubory do podadresáře `data` instalačního adresáře.

Poznámka

Obsah proměnné `$INSTDIR` se může od hodnoty uvedené v atributu `InstallDir` lišit např. z toho důvodu, že uživatel změnil instalační adresář v jednom z dialogových oken průvodce.

Instalace souborů

Instalované soubory určujeme příkazem `File`, za kterým následuje jméno souboru. Můžeme zde uvést i žolík — tedy mj. zápis `*.*` určující, že chceme všechny soubory v daném adresáři. Například příkazem

```
File fraktál.jar
```

instalujeme soubor `fraktál.jar`. Uvedeme-li před jménem adresáře parametr `/r`, požadujeme rekurzivní prohledání podadresářů.

Jméno instalačního a odinstalačního programu

Jméno instalačního programu určíme atributem `OutFile`. Například zápisem

```
OutFile "Instaluj.exe"
```

přikážeme vytvořit instalační program pojmenovaný `Instaluj.exe`. Program pro odstranění aplikace vytvoříme příkazem `WriteUninstaller`, za který napíšeme jméno programu, a to včetně cesty. Například příkazem

```
WriteUninstaller "$INSTDIR\Odinstaluj.exe"
```

vytvoříme odstraňovací (odinstalační) program v adresáři aplikace, který se bude jmenovat `Odinstaluj.exe`.

Odstranění souborů

V sekci `Uninstal` popisujeme činnost odinstalačního programu. Obvykle zde potřebujeme smazat soubory tvořící aplikaci. K tomu použijeme příkaz `Delete`, za kterým uvedeme jméno souboru. I zde můžeme používat žolíky. Například příkazem

```
Delete $INSTDIR\*.*
```

smažeme všechny soubory v adresáři aplikace.

Vytvoření a smazání adresáře

Chceme-li vytvořit nový adresář, použijeme příkaz `CreateDirectory`, za kterým uvedeme jméno vytvářeného adresáře. Například příkazem

```
CreateDirectory "$INSTDIR\data"
```

vytvoříme podadresář `data` instalačního adresáře. Pro odstranění adresáře použijeme příkaz `RmDir`, za kterým opět uvedeme jméno adresáře. Uvedeme-li před jménem adresáře přepínač `/r`, smaže příkaz `RmDir` uvedený adresář, i když není prázdný.

Překlad instalačního skriptu

Překladem instalačního skriptu vznikne instalační balíček, který bude uložen ve stejném adresáři jako instalační skript. Překlad lze spustit z příkazové řádky nebo z okna, které otevřeme odkazem *Compile NSI scripts* z hlavního okna programu NSI. Podrobněji o tom budeme hovořit v závěru oddílu 11.2.1.

11.2 Jednoduchý instalační program

Nyní už víme o skriptování pro NSIS dost, abychom mohli k našemu programu pro práci s fraktály vytvořit jednoduchý instalační nástroj. Při jeho dalším vývoji si postupně ukážeme možnosti, o nichž jsme dosud nehovořili, ale které nám mohou být velmi užitečné.

11.2.1 První pokus

Abychom se seznámili s prací se soubory a adresáři, začneme u instalace, jež nevyužívá archiv JAR; můžeme si to vyzkoušet např. na Fraktálníku verze 1.0.

Vytvoříme pomocný adresář, který nazveme např. `Instalace`, a umístíme do něj vzorovou instalaci našeho programu (viz též 3.7.4) včetně spouštěcího skriptu `Fraktálník.bat`. Připomeňme si, že tato instalace obsahuje podadresáře `fraktal`, `nástroje` a `html`. V kořenovém adresáři instalace jsou navíc soubory `Julka.png` a `strom.png` a spouštěcí skript `Fraktálník.bat`. Je-li adresář s instalací uveden v systémové proměnné `PATH`, spustí uživatel náš program příkazem

```
Fraktálník
```

Pro spouštěcí skript můžeme také vytvořit zástupce na ploše.

Nejjednodušší instalační skript

Naše vědomosti nám už umožňují napsat nejjednodušší fungující instalační skript, který ovšem zatím nebude prakticky použitelný. Takovýto skript může mít následující tvar (příkazy v oddílu atributů jsou dostatečně vysvětleny v komentářích).

```
#Soubor Fraktál.nsi
#Obecné
!include "MUI2.nsh"           ;Nové uživatelské rozhraní
OutFile "Instalace.exe"      ;Jméno instalačního souboru
InstallDir "C:\Temp\Fr"      ;Instalační adresář
!insertmacro MUI_LANGUAGE "Czech" ;Jazyk: čeština
```

```

RequestExecutionLevel user ;Obvyklá oprávnění

#Sekce
Section "Instalace"
    SetOutPath "$INSTDIR"
#Instalované soubory
    File Fraktal.bat
    File Julka.png
    File strom.png
    SetOutPath "$INSTDIR\html"
    File html\*. *
    SetOutPath "$INSTDIR\fraktal"
    File fraktal\*. *
    SetOutPath "$INSTDIR\nástroje"
    File nástroje\*. *
#Vytvoření odinstalačního programu
    WriteUninstaller "$INSTDIR\Odstranit.exe"
SectionEnd

#sekce odinstalátoru - popisuje, co má odstranit
Section "Uninstall"
    RMDir /r "$INSTDIR"
SectionEnd

```

Poznámky

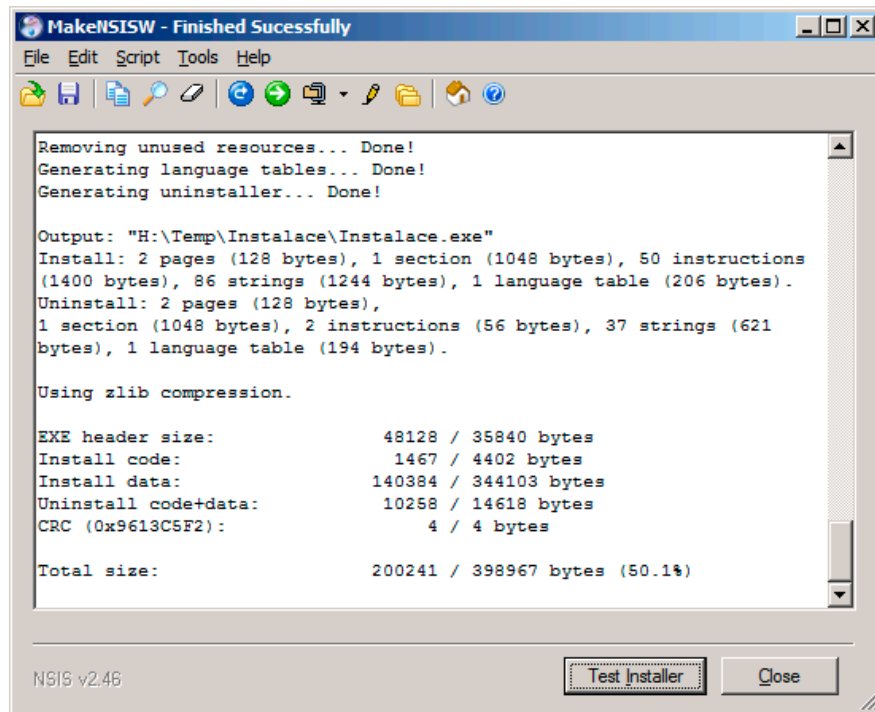
- Při určování instalovaných souborů v domovském adresáři aplikace nelze použít žolík (zápis *.*), neboť v něm je umístěn i instalační skript, který nechceme na cílový počítač kopírovat.
- Chceme-li instalovat adresář včetně podadresářů, použijeme v příkazu File volbu /r.

Překlad vytvořeného skriptu

Soubor s instalačním skriptem pojmenujeme `Fraktál1.nsi`, uložíme ho do adresáře s instalací a přeložíme ho. To znamená, že spustíme program NSIS a v jeho okně klepneme na odkaz *Compile NSI scripts*. Objeví se okno, které vidíte na obr. 11.2, bude však téměř prázdné. Příkazem *File | Load Skript...* nebo stisknutím tlačítka *Load Script...* (prvního v panelu nástrojů) otevřeme standardní dialogové okno pro otevření souboru, kterým vyhledáme překládaný skript. Vybereme ho a stiskneme tlačítko *Otevřít*. NSIS načte skript a pokusí se ho přeložit. V okně, které vidíte na obr. 11.2, bude zobrazovat zprávy o průběhu a výsledku překladač.

Je-li vše v pořádku, bude výpis končit souhrnnou zprávou o vytvořeném instalačním programu a v poslední řádce bude uvedena velikost vytvořeného souboru (Total size), souhrnná velikost instalovaných souborů a kompresní poměr. Ten závisí na povaze instalovaných souborů; v našem případě zabírá instalační program 50 % velikosti instalovaných souborů.

Obsahuje-li skript nějakou chybu, překlad skončí předčasně, instalační balíček se nevytvoří a v tomto okně najdeme zprávu, která nám řekne alespoň základní informace o zjištěném problému. Výpis bude v tom případě končit např. zprávou



Obrázek 11.2: Informace o průběhu překladač skriptu

Error in script "H:\Instalace\Fraktál.nsi" on line 22 -
- aborting creation process

Může se také stát, že překlad skončí úspěšně, ale překladač ohlásí jedno nebo několik varování. Je rozumné si tato varování přečíst a pokusit se odstranit jejich příčiny, neboť často jde o přehlédnutí programátora, která vlastně znamenají chybu.

Test instalace

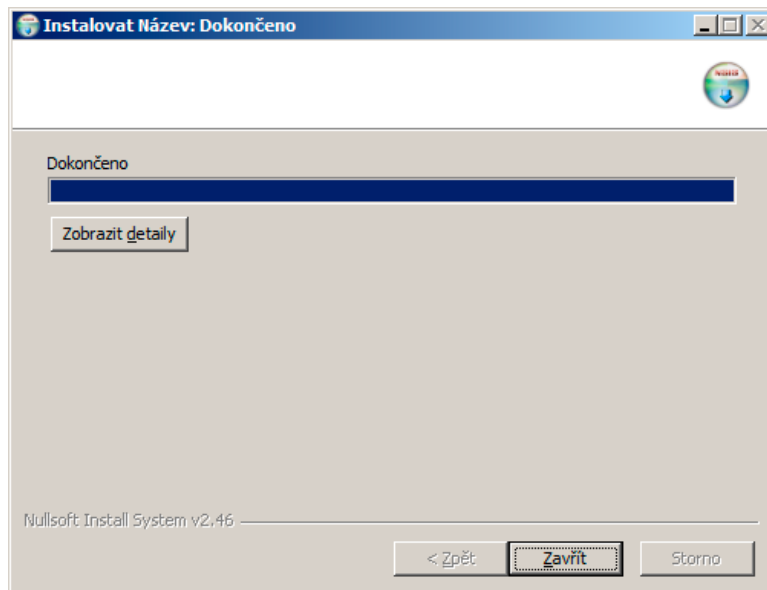
Po úspěšném skončení překladač najdeme v adresáři *Instalace* soubor *Instalace.exe* obsahující instalační program. Tlačítko *Test Installer* v okně, z něhož jsme spouštěli překlad, nám umožňuje ihned ho vyzkoušet. O průběhu a úspěšném dokončení instalace nás instalační program informuje dialogem, který vidíte na obr. 11.3. Podobný dialog uvidíme, když spustíme odinstalační program *Odstranit.exe*, který nám instalační program vytvořil v domovském adresáři aplikace.

Poznamenejme, že bude-li adresář aplikace otevřen v jakémkoli programu, odinstalační program sice odstraní soubory *Fraktálníku*, ale nesmaže adresář.

Nad prvním instalačním balíčkem

Náš první instalační program sice funguje, ale má dva zjevné nedostatky (a řadu méně zjevných). Podívejme se na ně:

- Sdělení *Instalovat Název: Dokončeno* v titulkové liště dialogu na obr. 11.3 je nesmyslné. Je důsledkem skutečnosti, že instalační program nezná jméno našeho instalovaného programu, takže si dosadil implicitní označení *Název*.



Obrázek 11.3: Instalace úspěšně proběhla

- Cílový adresář je v něm napevno určen. Takový instalační program není prakticky použitelný.

Vedle toho by bylo rozumné, aby instalační program seznámil uživatele s licenčním ujednáním, aby odinstalační program uživatele informoval, že se chystá program odstranit, a umožnil mu tuto akci zrušit, jestliže ji spustil omylem, aby vytvořil zástupce programu na ploše atd. Instalační program by také měl uživateli umožnit volbu součástí, které chceme instalovat (ne každý bude např. chtít instalovat nápovědu nebo – budeme-li instalovat některou z pozdějších verzí – knihovnu pro vývoj vlastních fraktálů).

První nedostatek napravíme ihned; s nástroji, které nám umožní vyřešit zbývající problémy, se seznámíme v následujícím oddílu.

Jméno instalovaného programu

Mezi atributy v instalačním skriptu vložíme příkazy

```
Name "Fraktálník"
Caption "Program pro výpočet fraktálů"
```

První z nich určuje jméno instalačního programu, druhý určuje nápis v titulku dialogů, které instalační skript zobrazí. Po této úpravě bude v titulkové liště dialogu oznamujícího dokončení instalace nápis *Program pro výpočet fraktálů: Dokončeno*. V titulku okna oznamujícího odstranění Fraktálníku bude nápis *Odinstalovat Fraktálník: Dokončeno*.

Upravený instalační skript najdete na webu v adresáři skripty v souboru `Fraktál1a.nsi`.

11.3 Průvodce instalací

Než se pustíme do psaní první verze skriptu pro vytvoření průvodce instalací, povíme si něco o nástrojích, které použijeme.

Tabulka 11.1: Možné úrovně oprávnění v příkazu RequestExecutionLevel

Hodnota	Význam
none	Rozhodnutí ponecháno na operačním systému
user	Běžná uživatelská úroveň
highest	Nejvyšší úroveň dostupná danému uživateli
admin	Administrátorská úroveň

11.3.1 Úroveň oprávnění

Kdybychom se pokusili instalovat náš program pod Windows 7 nebo novějším do adresáře Program Files nebo některého jiného systémového adresáře, skončil by neúspěchem, protože nemá odpovídající oprávnění. Ke získání tohoto oprávnění slouží, jak už víme, příkaz RequestExecutionLevel, za kterým uvedeme požadovanou úroveň. Možné hodnoty ukazuje tabulka 11.1.

Uvedeme-li none, bude výsledek stejný, jako když příkaz RequestExecutionLevel neuvedeme vůbec. V takovém případě bude program požadovat administrátorskou úroveň. Požaduje-li instalační program vyšší než běžnou uživatelskou úroveň, dotáže se Windows uživatele, zda souhlasí se zvýšením úrovně oprávnění, a dostane-li zápornou odpověď, ukončí instalační program.

Stejnou úroveň oprávnění jako instalační program bude mít i odinstalační program.

11.3.2 Stránky průvodce

Chceme-li vytvořit složitější instalační program, použijeme tzv. stránky (page), které představují okna průvodce instalací. V hlavičkovém souboru MUI2.nsh najdeme předdefinované stránky pro nejčastější okna průvodce. Skriptovací jazyk NSIS umožňuje vytvářet i vlastní stránky; s tím se setkáme v oddílu 11.5.

Připravené stránky průvodce instalací

Tabulka 11.2 obsahuje přehled připravených stránek průvodce instalací.

Tabulka 11.2: Předdefinované stránky průvodce instalací

Stránka	Význam
MUI_PAGE_WELCOME	Úvodní stránka průvodce
MUI_PAGE_LICENSE soubor	Licenční ujednání; soubor je jméno souboru s textem licence
MUI_PAGE_COMPONENTS	Přehledem součástí aplikace
MUI_PAGE_DIRECTORY	Volba cílového adresáře instalace
MUI_PAGE_STARTMENU stránka proměnná	Vytvoření položky v nabídce <i>Start</i>
MUI_PAGE_INSTFILES	Zobrazuje průběh instalace
MUI_PAGE_FINISH	Závěrečná stránka průvodce

Jednotlivé předdefinované stránky vložíme do skriptu příkazem !insertmacro. Například úvodní stránku průvodce vložíme příkazem

```
!insertmacro MUI_PAGE_WELCOME
```

Příkazy pro vkládání stránek průvodce je třeba uvést mezi atributy, před sekcemi. Za příkazem pro vložení stránky s licencí následuje textový řetězec obsahující jméno souboru s textem licenčního ujednání. To může být buď textový soubor, nebo – chceme-li formátovaný text – dokument ve formátu RTF, který lze vytvořit například v editoru MS Word.

Za příkazem pro vložení stránky pro vytvoření položky v nabídce *Start* následuje identifikátor stránky (ten lze použít v pozdějších odkazech na tuto stránku) a jméno proměnné, v níž bude jméno skupiny v nabídce zadané uživatelem. K oběma těmto stránkám se vrátíme dále.

Připravené stránky průvodce odinstalací

Také pro vytvoření průvodce odinstalací aplikace poskytuje NSIS sadu připravených stránek; jejich přehled ukazuje tabulka 11.3.

Tabulka 11.3: Předdefinované stránky průvodce odinstalací

Stránka	Význam
MUI_UNPAGE_WELCOME	Úvodní stránka průvodce
MUI_UNPAGE_CONFIRM	Žádost o potvrzení, že uživatel chce aplikaci odstranit
MUI_UNPAGE_LICENSE	Licenční ujednání; <i>soubor</i> je jméno souboru s textem licence
MUI_UNPAGE_COMPONENTS	Stránka s přehledem součástí aplikace
MUI_PAGE_DIRECTORY	Stránka pro volbu adresáře
MUI_PAGE_INSTFILES	Zobrazuje průběh odstraňování
MUI_PAGE_FINISH	Závěrečná stránka průvodce

Použití těchto stránek předepíšeme – stejně jako použití stránek průvodce instalací – příkazem `!insertmacro` uvedeným mezi atributy. Poznamenejme, že opravdu důležitá je pouze stránka se žádostí o potvrzení, protože dává uživateli možnost zastavit akci, kterou spustil omylem.

Popis sekcí v průvodci

V okně pro výběr součástí aplikace, které chce uživatel instalovat, se zobrazí zaškrťovací políčka pro jednotlivé sekce, které jsme pojmenovali a neoznačili jako skryté. Sekce, které u sebe nemají volbu /o, budou zaškrtnuty. Jako text se u tohoto políčka zobrazí jméno sekce uvedené v její deklaraci. Chceme-li dát uživateli k dispozici podrobnější popis, vložíme do skriptu za poslední sekci příkazy tvaru

```
LangString DESC_index ${LANG_jazyk} "Popis sekce"
```

Zde *index* je index sekce, k níž se tento příkaz vztahuje (viz 11.1.1), a *jazyk* je jméno jazyka, v němž bude instalátor pracovat. "Popis sekce" je v uvozovkách uzavřený popis obsahu sekce, který se v okně zobrazí, když uživatel ukáže na danou sekci myší. Například příkaz

```
LangString DESC_Sekce1 ${LANG_Czech} "Základní složky aplikace"
```

způsobí, že když uživatel ukáže myši v okně instalátoru pro výběr složek aplikace na položku odpovídající sekci s indexem `Sekce1`, zobrazí se text *Základní složky aplikace*.

Řetězce ve zvoleném jazyce

Okna instalátoru — tedy stránky — obsahují řadu nápisů, které se liší jazyk od jazyka. Mohou se lišit v různých sekcích. Při jejich určování využijeme opět index sekce. Instrukce pro nastavení nápisů začínají příkazem

```
!insertmacro MUI_FUNCTION_DESCRIPTION_BEGIN
```

a končí příkazem

```
!insertmacro MUI_FUNCTION_DESCRIPTION_END
```

Mezi nimi uvedeme pro každou ze sekcí příkaz tvaru

```
!insertmacro MUI_DESCRIPTION_TEXT ${index} $(DESC_index)
```

kde *index* je index sekce. Tyto příkazy uvedeme za popisem sekcí.

Přerušení instalace uživatelem

Každé z oken průvodce instalací (kromě závěrečného) obsahuje tlačítko *Storno*, kterým můžeme instalaci zrušit. Je ale rozumné, aby se instalační program zeptal, zda si uživatel opravdu přeje skončit, a v případě, že toto tlačítko stiskl omylem, aby mohl v instalaci pokračovat. To zařídíme příkazem²

```
!define MUI_ABORTWARNING
```

který umístíme před specifikace stránek instalátoru. Nyní se po stisknutí tlačítka *Storno* objeví okénko s dotazem, zda si uživatel opravdu přeje ukončit instalaci, a jestliže uživatel odpoví *Ne*, bude instalace pokračovat.

11.3.3 Jednoduchý průvodce instalací

Informace, které jsme si řekli v předchozím oddílu, nám stačí k vytvoření jednoduchého průvodce instalací Fraktálníku. Projdeme postupně všechny části skriptu; celý skript najdete na webu v adresáři `skripty\Instalace2` v souboru `Fraktál2.nsi`.

Atributy

Začátek skriptu se prakticky neliší od předchozí verze: Určíme jméno instalačního programu, cílový adresář a titulek oken a vyžádáme si uživatelskou úroveň práv. Určíme také, že se má skript dotázat uživatele, zda si opravdu přeje skončit, když v některém z oken průvodce stiskne tlačítko *Storno*.

```
# Soubor 12\03\Fraktál.nsi
!include "MUI2.nsh"
OutFile "Instalace.exe"
InstallDir "C:\Temp\Fraktál"
```

²Příkaz `!define` definuje makro; jeho význam je velmi podobný významu direktivy `#define` v jazyce C.

```
Name "Fraktálník"
Caption "Program pro výpočet fraktálů"
!define MUI_ABORTWARNING
RequestExecutionLevel user
```

Poté určíme stránky, které bude náš průvodce mít, a nastavíme jazyk. Stránky musíme zapsat v části vyhrazené atributům v pořadí, ve kterém se mají zobrazit. Zároveň vytvoříme textový soubor `Licence.rtf`, zapíšeme do něj text licenčního ujednání a uložíme ho do adresáře se vzorovou instalací.

```
#Stránky průvodce instalací
!insertmacro MUI_PAGE_WELCOME
!insertmacro MUI_PAGE_LICENSE "Licence.rtf"
!insertmacro MUI_PAGE_COMPONENTS
!insertmacro MUI_PAGE_DIRECTORY
!insertmacro MUI_PAGE_INSTFILES
!insertmacro MUI_PAGE_FINISH

#Stránky průvodce odinstalací
!insertmacro MUI_UNPAGE_WELCOME
!insertmacro MUI_UNPAGE_CONFIRM
!insertmacro MUI_UNPAGE_INSTFILES
!insertmacro MUI_UNPAGE_FINISH

#Jazyk průvodce
!insertmacro MUI_LANGUAGE "Czech"
```

Instalační sekce

Tato sekce je téměř stejná jako v předchozí verzi, liší se pouze jméno sekce a index; pro pohodlí čtenáře zde zopakujeme i její tělo, vynecháme však komentáře.

```
Section "Program Fraktálník" Sekce1
  SetOutPath "$INSTDIR"
  File Fraktal.bat
  File Julka.png
  File strom.png
  SetOutPath "$INSTDIR\html"
  File html\*. *
  SetOutPath "$INSTDIR\fraktal"
  File fraktal\*. *
  SetOutPath "$INSTDIR\nástroje"
  File nástroje\*. *
  WriteUninstaller "$INSTDIR\Odinstaluj.exe"
SectionEnd
```

Soubor s textem licenčního ujednání nebude tímto průvodcem do cílového adresáře instalován, pouze se zobrazí v dialogu, který vidíte na obr. 11.4 vpravo. Kdybychom ho chtěli také instalovat, museli bychom v této sekci uvést odpovídající příkaz `File`.

Odstalalační sekce

Také tato sekce je stejná jako v předchozí verzi: obsahuje jediný příkaz, kterým smažeme všechny soubory a adresáře aplikace.

```
Section "Uninstall"
  RMDir /r "$INSTDIR"
SectionEnd
```

Popis instalační sekce

Za poslední sekcí deklarujeme popis jediné instalační sekce našeho skriptu.

```
LangString DESC_Sekce1 ${LANG_Czech} "Hlavní program, dynamické knihovny a \
  nápověda"
```

Přiřazení řetězců sekcím

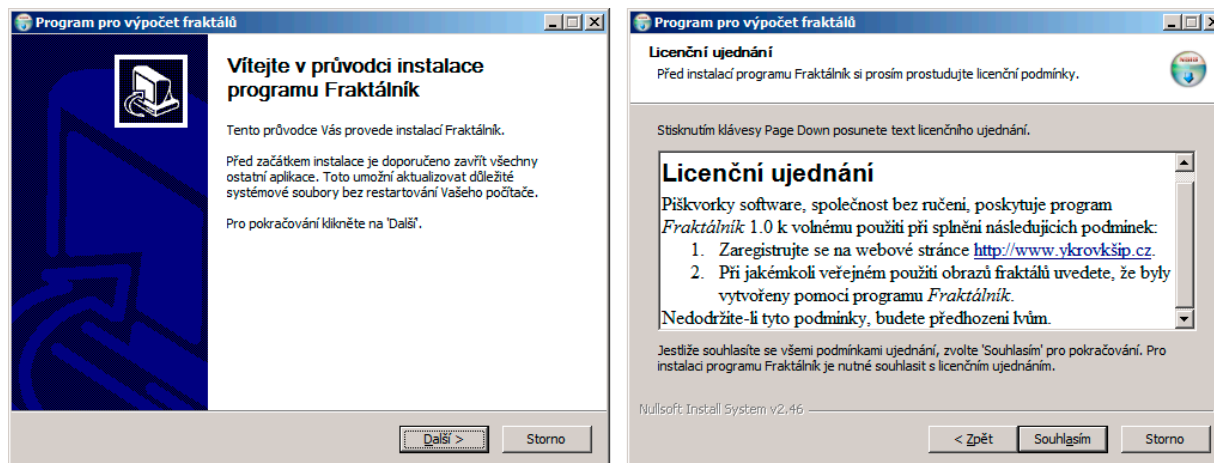
Poslední částí našeho skriptu bude přiřazení českých řetězců sekcím; jde o nápisy na tlačítkách, v oknech apod. O to se postarají příkazy

```
!insertmacro MUI_FUNCTION_DESCRIPTION_BEGIN
  !insertmacro MUI_DESCRIPTION_TEXT ${Sekce1} $(DESC_Sekce1)
!insertmacro MUI_FUNCTION_DESCRIPTION_END
```

Nyní můžeme skript přeložit.

Instalace pomocí průvodce

Jestliže vytvořený instalační program spustíme, objeví se nejprve úvodní stránka, kterou vidíte v levé části obr. 11.4. Po stisknutí tlačítka *Další* se zobrazí stránka s licenčním ujednáním, kterou vidíte v pravé části obr. 11.4.



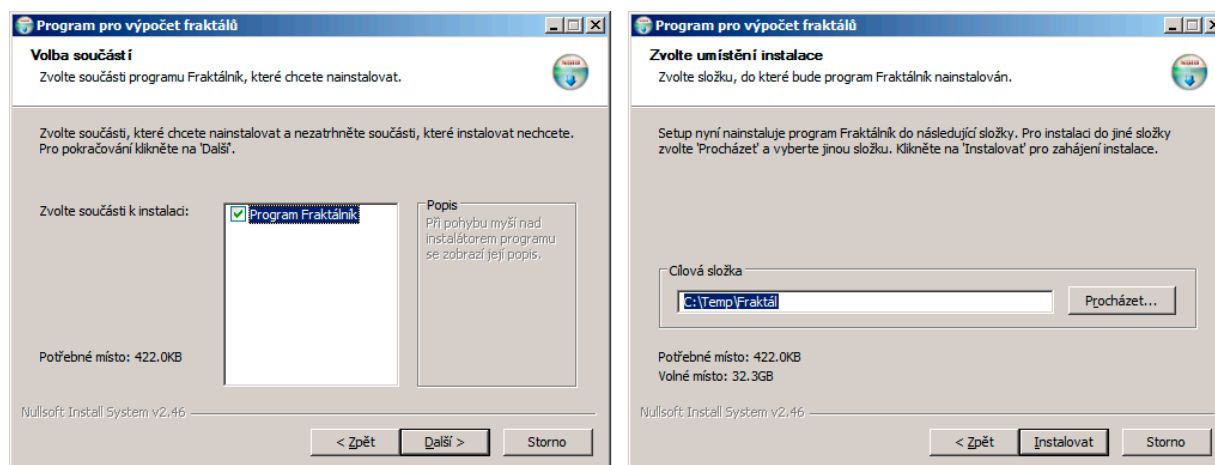
Obrázek 11.4: První dvě stránky průvodce instalací programu Fraktálník

Stiskne-li uživatel tlačítko *Storno*, řekne tím, že s licenčními podmínkami nesouhlasí, a instalace skončí. Stiskne-li uživatel tlačítko *Souhlasím*, bude instalace pokračovat a objeví se stránka,

na níž lze vybrat instalované součásti (obr. 11.5 vlevo). Už víme, že ty odpovídají jednotlivým sekcím v instalačním skriptu. Náš skript má zatím jedinou sekci, a proto se zde zobrazí pouze jediná možnost. Poznamenejme, že zruší-li uživatel zaškrtnutí této sekce, neinstaluje se nic.

Ukáže-li uživatel kurzorem myši na jméno sekce, zobrazí se v pravé části okna na panelu *Popis* text, který jsme uvedli v příkazu `LangString`; jinak je v tomto poli zobrazen (nebo spíše naznačen) jakýsi implicitní text.

Po stisknutí tlačítka *Další* se zobrazí další okno, v němž může uživatel zvolit cílový adresář (obr. 11.5 vpravo). Jako implicitní možnost se zde zobrazí adresář, který jsme uvedli v příkazu `InstallDir`. Nebude-li tento adresář uživateli vyhovovat, vyvolá stisknutím tlačítka *Procházet* standardní dialog pro volbu adresáře, s jehož pomocí určí nový cílový adresář. K adresáři, který uživatel zvolí, připojí průvodce automaticky jméno posledního podadresáře z výchozí hodnoty uvedené v příkazu `InstallDir` a tuto možnost zobrazí ve svém okně. To znamená, že uvedeme-li jako výchozí adresář `C:\Temp\Fraktál` a zvolí-li uživatel adresář `C:\Aplikace`, zobrazí se v okně průvodce jako cílový adresář `C:\Aplikace\Fraktál`. Uživatel může jméno cílového adresáře ještě ručně přepsat.



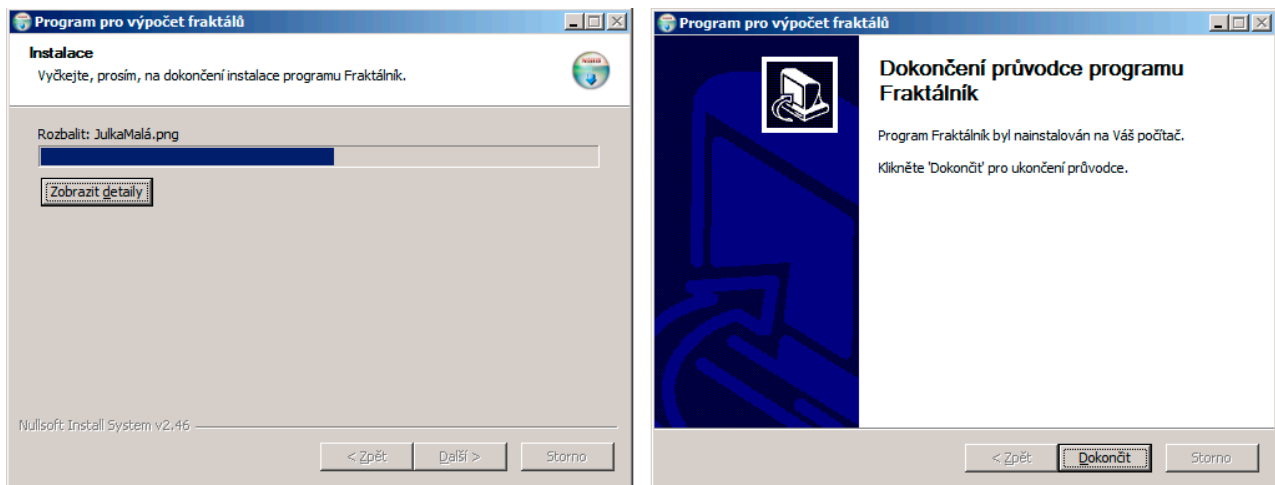
Obrázek 11.5: Okno pro volbu instalovaných součástí a cílového adresáře

Stisknutím tlačítka *Instalovat* spustí uživatel vlastní kopírování souborů. O průběhu ho bude informovat okno, které vidíte na obr. 11.6 vlevo (v našem případě jen blikne, neboť instalovaný program není příliš velký). Po skončení instalace se objeví závěrečné okno, které vidíte na obr. 11.6 vpravo.

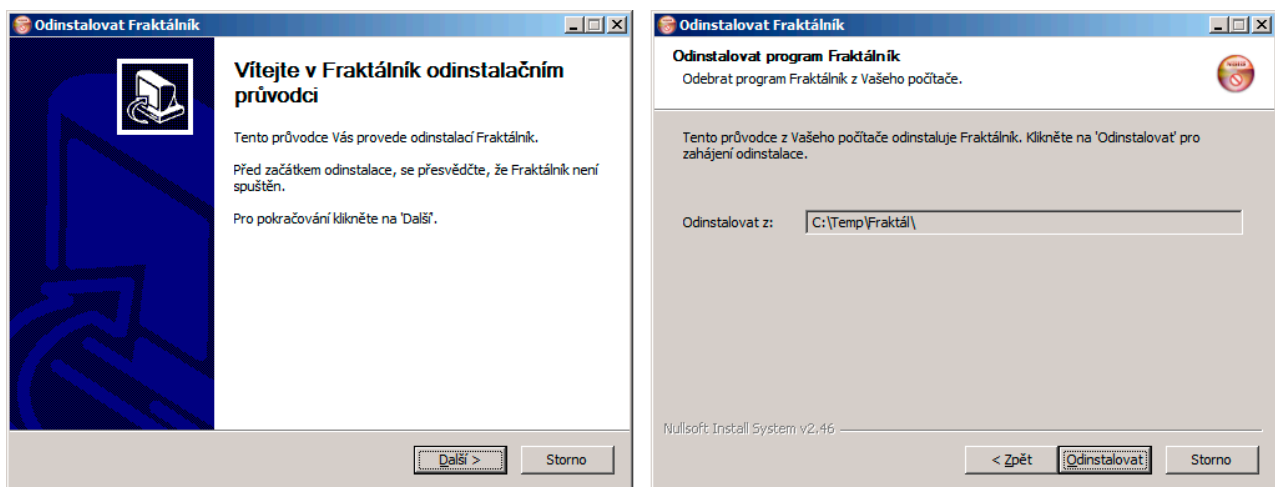
Odstranění programu

Spustíme-li program `Odinstaluj.exe`, zobrazí se úvodní okno, které vidíte na obr. 11.7 vlevo. Po stisknutí tlačítka *Další* se objeví okno s dotazem, zda uživatel opravdu chce odstranit program *Fraktálník* ze svého počítače (obr. 11.7 vpravo).

Jestliže uživatel stiskne tlačítko *Storno*, tento program skončí, aniž cokoli udělá. Stiskne-li tlačítko *Odinstalovat*, rozběhne se odinstalace; o jejím průběhu bude odinstalovací program uživatele informovat oknem podobným jako na obr. 11.6 vlevo a po skončení se objeví závěrečné okno, které ukazuje obr. 11.8.



Obrázek 11.6: Poslední dvě okna průvodce instalací Fraktálníku



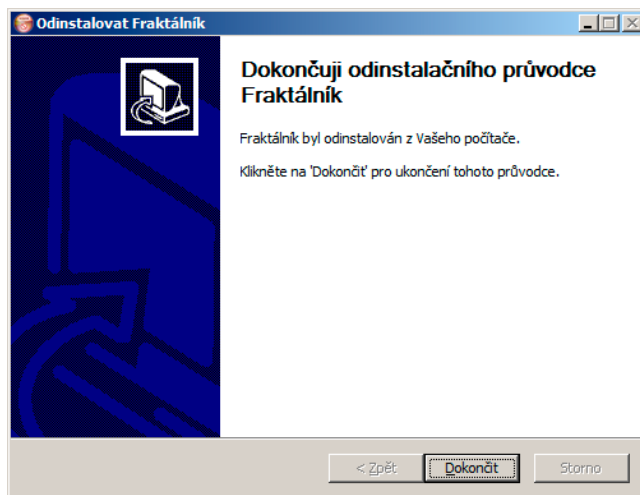
Obrázek 11.7: První dvě okna průvodce odinstalací Fraktálníku

Nad naším prvním průvodcem

S výsledkem bychom už mohli být spokojeni,³ nebýt jedné věci: Některé nápisy v oknech jsou slušně řečeno problematické jak z hlediska češtiny, tak z hlediska významu. Podívejme se na několik křiklavých příkladů:

- Sdělení v úvodním okně, že je třeba zavřít všechny ostatní aplikace, je „mimo“, neboť při instalaci Fraktálníku pouze zatím kopírujeme soubory do uživatelského adresáře, nic víc, takže není důvod uživatele k něčemu podobnému nutit.
- Formulace „Tento průvodce vás provede instalací Fraktálník“ je nesmyslná, neboť budí dojem, že se Fraktálník jmenuje instalace, nikoli instalovaný program.
- Neukazujeme-li v okně *Volba součástí* myši na žádnou ze součástí, je v poli pro popis těchto komponent zobrazen text „Při pohybu myši nad instalátorem programu se zobrazí

³Zatím pomíjíme skutečnost, že součástí instalace je soubor `Fraktál.bat`, který obsahuje napevno zapsanou cestu k adresáři s instalací, a jestliže uživatel tuto cestu v odpovídajícím okně průvodce změní, je třeba upravit tento soubor ručně. To vyřešíme v následujícím pododdílu.



Obrázek 11.8: Fraktálník byl úspěšně odstraněn

její popis“. Nejde ovšem o popis myši, jak tato věta tvrdí. Mělo by tam být „...nad jménem součásti programu se zobrazí její popis.“

- Také vyjádření „...nezatrhněte součásti...“ je více než nešikovné. Navíc v této větě chybí čárka.
- Formulace „Vítejte v Fraktálník odinstalačním průvodci“ je opravdu jazykovou perlou.

Tyto texty jsou nejspíše výsledkem nikým nekontrolovaného strojového překladu z angličtiny. Uvážíme-li, že průvodce instalací bude zpravidla prvním setkáním uživatele s naším produktem, je jasné, že bude významně spoluročovat jeho dojem z celého programu a že podle něj bude soudit i o péči, kterou jsme věnovali podstatně důležitějším součástem aplikace. Naštěstí umožňuje NSIS některé texty v těchto oknech změnit.

Vedle toho by neškodilo, kdyby místo univerzálního a nic neříkajícího obrázku na úvodní a závěrečné stránce průvodce bylo např. logo dodavatele. I to samozřejmě lze.

11.3.4 Úprava oken průvodce

Všechny příkazy, kterými měníme text nebo obrázek v některém z oken průvodce, začínají direktivou `!define`, pak následuje specifikace nahrazované položky a její nový text uzavřený v uvozovkách. Pro přechod na novou řádku ve vypisovaném textu použijeme zápis `$\n`.

Některé možnosti úprav

Úplný přehled příkazů pro úpravu oken průvodce najdete v dokumentaci. Zde si ukážeme jen některé z možností, které pro nás budou zajímavé. Stručný souhrn těchto příkazů obsahuje tabulka 11.4.

První dva příkazy lze použít pro kteroukoli stránku kromě úvodní a závěrečné stránky průvodců instalací a odinstalací. Tyto příkazy změní text nadpisu, resp. doprovodný text v bílém poli v záhlaví okna. Je třeba je zapsat vždy před příkazem vkládajícím odpovídající stránku a vztahují se pouze k této stránce.

Budeme-li např. chtít změnit záhlaví stránky pro volbu součástí z „Volba součástí“ na „Zvolte součásti“, aby byl text stylově podobný textu na následující stránce (a abychom se zbavili

Tabulka 11.4: Příkazy pro úpravu textu na stránkách průvodce

Stránka	Význam
MUI_PAGE_HEADER_TEXT	Titulek v záhlaví okna
MUI_PAGE_HEADER_SUBTEXT	Text pod titulkem v záhlaví
MUI_WELCOMEPAGE_TITLE	Nadpis v úvodní stránce průvodce
MUI_WELCOMEPAGE_TEXT	Text v úvodní stránce průvodce
MUI_COMPONENTSPAGE_TEXT_TOP	Text v horní části stránky pro volbu součástí instalovaného programu
MUI_COMPONENTSPAGE_TEXT_DESCRIPTION_INFO	Text, který se zobrazí na stránce pro volbu součástí, když na žádnou neukazujeme
MUI_DIRECTORYPAGE_TEXT_TOP	Text, který se zobrazí v horní části stránky pro volbu cílového adresáře
MUI_FINISHPAGE_TITLE	Nadpis na závěrečné stránce průvodce
MUI_UNCONFIRMPAGE_TEXT_TOP	Text v horní části okna se žádostí o potvrzení, že uživatel chce odinstalovat program

zbytečné mezery před závěrečným „í“, a zároveň budeme chtít upravit i doprovodný text, změníme skript následujícím způsobem:

```
!insertmacro MUI_PAGE_LICENSE "Licence.txt"
!define MUI_PAGE_HEADER_TEXT "Zvolte součásti"
!define MUI_PAGE_HEADER_SUBTEXT "Na této stránce zvolte \
součásti aplikace, které chcete instalovat."
!insertmacro MUI_PAGE_COMPONENTS
```

Úprava obrázku v úvodním a závěrečném okně

Chceme-li změnit obrázek v úvodním a závěrečném okně průvodce, použijeme příkaz

```
!define MUI_WELCOMEFINISHPAGE_BITMAP soubor
```

kde *soubor* je jméno souboru (v případě potřeby včetně cesty) s obrázkem ve formátu BMP. Doporučené rozměry jsou 164 × 314 obrazových bodů.

Úprava instalačního skriptu

Nyní se můžeme pustit do úprav našeho instalačního skriptu. Zde si ukážeme pouze úpravu úvodního okna průvodce. Před příkazy pro vložení stránek průvodce zapíšeme příkazy

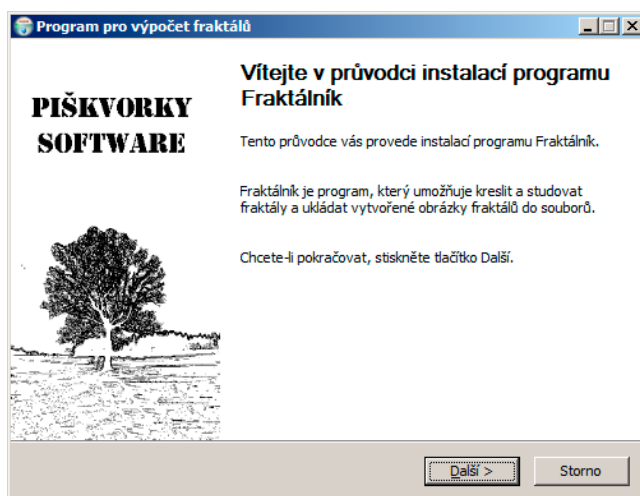
```
# Soubor 12\04\Fraktál.nsi
!define MUI_WELCOMEPAGE_TITLE "Vítejte v průvodci instalací \
programu Fraktálník"
!define MUI_WELCOMEPAGE_TEXT "Tento průvodce vás provede \
instalací programu Fraktálník.$\n$\nFraktálník je program, \
```

```

který umožňuje kreslit a studovat fraktály a ukládat vytvořené \
obrázky fraktálů do souborů.\
$\n$\n$\nChcete-li pokračovat, stiskněte tlačítko Další."
!define MUI_WELCOMEFINISHPAGE_BITMAP "Strom4.bmp"

```

Připomeňme si, že zápis `$(\n` ve vkládaném textu znamená přechod na novou řádku. Jako logo jsme použili soubor `Strom4.bmp`, který je k dispozici v adresáři `ikony` na webu. Podobu úvodního okna průvodce instalací Fraktálníku po této úpravě ukazuje obr. 11.9. Soubor `Fraktál3.nsi`, obsahující skript, v němž jsou upraveny i další texty, najdete na webu v adresáři skripty.



Obrázek 11.9: Úvodní stránka průvodce s upravenými texty a s logem dodavatele

Poznámka

V předchozích dvou výpisech jsme text, který chceme vložit do okna průvodce, rozdělili na několik řádek. Jak víme, jestliže řádka končí obráceným lomítkem, spojí ji překladač s následující řádkou. Přitom z následující řádky odstraní úvodní mezery. Proto jsme museli zapsat mezeru, která odděluje poslední slovo na jedné řádce od prvního slova na následující řádce, ještě před obráceným lomítkem.

11.3.5 Vytvoření textového souboru v instalačním balíčku

Součástí instalace je spouštěcí skript, který mimo jiné nastavuje domovský adresář aplikace jako aktuální adresář. To znamená, že je v něm třeba cestu k adresáři instalace upravit podle adresáře, který uživatel zvolí. Jednou z možností, jak to udělat, je vytvořit tento skript v průběhu instalace – a to uděláme i my.

Zápis do souboru v instalačním programu

Ve kterékoli ze sekcí můžeme vytvořit textový soubor. K tomu nám poslouží příkazy `FileOpen`, `FileWrite` a `FileClose`. Syntax prvního z nich je

```
FileOpen proměnná jméno režim
```

Do *proměnné* se uloží informace o otevřeném souboru, které použijeme v dalších operacích s ním; lze použít např. předdefinované proměnné \$0, . . . , \$9. Druhý parametr, *jméno*, je řetězec obsahující jméno souboru i s cestou. V tomto řetězci lze použít i proměnnou \$INSDIR apod. Třetí parametr, *režim*, vyjadřuje režim otevření; zde můžeme použít w (zápis), r (čtení) nebo a (připisování).

Příkaz pro zápis do souboru má tvar

```
FileWrite proměnná text
```

kde *proměnná* je proměnná určující soubor otevřený pro zápis a *text* je znakový řetězec, který chceme do souboru zapsat.

Po dokončení operace soubor uzavřeme příkazem

```
FileClose proměnná
```

kde *proměnná* určuje uzavíraný soubor.

Další příkazy pro práci se soubory najdete v dokumentaci; nám postačí uvedené tři.

Úprava instalačního skriptu

Ze souboru Fraktálník.nsi odstraníme příkaz pro kopírování souboru Fraktál.bat. Místo něho vložíme na závěr instalační sekce příkazy

```
FileOpen $9 "$INSDIR\Fraktálník.bat" w
FileWrite $9 "@echo off$\r$\n"
FileWrite $9 "CD $INSDIR$\r$\n"
FileWrite $9 "set classpath=.;fraktal.jar$\r$\n"
FileWrite $9 "java fraktal.Program$\r$\n"
FileClose $9
```

Zápisy \$\r\$\n ve vypisovaných řetězcích znamenají přechod na novou řádku.

Jestliže takto upravený soubor Fraktálník.nsi přeložíme a výsledný instalační program spustíme, přesvědčíme se, že vytvořený spouštěcí skript Fraktál.bat bude obsahovat jméno uživatelem vybraného adresáře.

Instalační skript po této úpravě najdete na webu v adresáři skripty v souboru Fraktálník4.nsi.

11.4 Pokročilejší instalační program

Za instalační program, kterýž jsme vytvořili v předchozím oddílu, se už nemusíme stydět. Do plnohodnotného instalačního programu pod Windows má však ještě daleko: Nevytvoří zástupce programu na ploše, nevytvoří odpovídající položku v nabídce *Start*, nevytvoří záznam o programu v registru Windows. Tyto možnosti nyní instalačnímu programu přidáme. Přitom přejdeme k verzi Fraktálníku označené 6.0.2 a budeme instalovat archiv JAR, takže půjde již o instalační program, který lze pro naše účely skutečně použít. Zároveň k programu sestavíme příručku uživatele (např. jako soubor ve formátu PDF), kterou budeme distribuovat spolu s programem. Volitelnou součástí instalace bude i knihovna obsahující třídu komplexních čísel a rozhraní VýpočetFraktálu a dokumentace k ní. Připomeňme, že tuto knihovnu jsme vytvořili v oddílu 5.1.2.

Adresář se vzorovou instalací bude tedy obsahovat na počátku soubor `fraktal6_0_2.jar` a soubor `Licence.rtf`; dále v něm vytvoříme podadresář `doc` s uživatelskou příručkou (souborem `Fraktál 6 - příručka uživatele.pdf`) a se souborem `Licence.txt`.⁴ Vedle toho zde vytvoříme podadresář `lib` obsahující soubor `Knihovna.jar` a v něm podadresář `javadoc` s dokumentací k ní.

11.4.1 Zástupce a ikony

Naším prvním úkolem je vytvořit zástupce programu na ploše a přidělit mu vhodnou ikonu. Při té příležitosti také přidělíme ikonu instalátoru a odinstalátoru.

Ikony

Jako ikonu hlavního okna programu jsme použili soubor ve formátu PNG, jak to vyžaduje Java. Pro ikonu zástupce na ploše, instalátoru a odinstalátoru však musíme použít soubor `.ico`, což je formát používaný pro ikony pod Windows. K vytvoření ikon je k dispozici řada volně dostupných programů;⁵ umějí to zpravidla také vývojová prostředí pro Windows. V omezené míře to umí i program Malování, který je součástí Windows 7.

Potřebné ikony vytvoříme z obrázku „netopýra“, který nakreslíme naším programem. Ikonou programu bude netopýr uzavřený do čtverce stejné barvy; ikona instalačního programu by měla být podobná ikoně instalovaného programu, nikoli však stejná; my použijeme nezarámovaného netopýra.

Pro ikony odinstalačního programu nabízí NSIS implicitně symbol odpadkového koše; ten není špatný, ale nenaznačuje souvislost s odstraňovaným programem. Lepší je použít přeškrtnutou ikonu programu nebo ikonu doplněnou symbolem zákazu.

Takto upravené ikony najdete na webu v adresáři `ikony` v souborech nazvaných `Netopýr.ico`, `Netopýr1.ico` a `Netopýr2.ico`.

Ikona instalačního a odinstalačního programu

K určení ikony instalačního a odinstalačního programu použijeme příkazy `Icon` a `Uninstall-Icon`, za nimiž uvedeme jména souborů s ikonami. Tyto příkazy je třeba uvést mezi atributy skriptu, ale až za příkazy určujícími stránky, jinak použije překladač implicitní ikonu. To znamená, že v adresáři vzorové instalace vytvoříme podadresář `ikona`, uložíme do něj obě ikony a na závěr úseku instalačního skriptu popisujícího atributy vložíme příkazy

```
Icon ikona\Netopýr.ico
UninstallIcon ikona\Netopýr1.ico
```

Poznamenejme, že určíme-li ikonu instalačního programu, použije ji NSIS i v záhlaví většiny dialogů průvodce instalací (obr. 11.10); podobně ikonu odinstalačního programu použije v oknech průvodce odinstalací.

Zástupce programu na ploše

K vytvoření zástupce slouží příkaz `CreateShortcut`, který má tvar

⁴Zde použijeme prostý text, neboť uživatel nemusí mít na svém počítači program, kterým by mohl otevřít soubor ve formátu RTF.

⁵Jenom např. server <http://www.slunecnice.cz> jich nabízí několik desítek.

```
CreateShortcut soubor_zástupce zastupovaný_soubor par_přík_ř ikona
index_ikony další_parametry
```

První dva parametry – řetězce obsahující jméno souboru zástupce a jméno zastupovaného souboru – musí být uvedeny, ostatní, počínaje parametry příkazové řádky zastupovaného programu, lze vynechat; vynecháme-li však jeden parametr, musíme lze vynechat i všechny následující. Poznamenejme, že soubor s ikonou musí být k dispozici v cílovém adresáři instalace; to znamená, že při instalaci musíme překopírovat i adresář ikona.

Aby se zástupce zobrazil na ploše Windows, musíme ho umístit do vyhrazeného adresáře; pod Windows 7 je to adresář `C:\Users\<uživatel>\Desktop`. V instalačním skriptu se na něj odvoláme pomocí předdefinované proměnné `$DESKTOP`.

To znamená, že v hlavní sekci uvedeme příkaz

```
CreateShortcut "$DESKTOP\Fraktál.lnk" '$INSTDIR\fraktal6_0_2.jar' "" \
"$INSTDIR\ikona\Netopýr.ico" 0
```

Program pro odinstalování aplikace by měl zástupce odstranit. K tomu použijeme příkaz `Delete`; do sekce odinstalačního programu přidáme příkaz

```
Delete "$DESKTOP\Fraktál.lnk"
```

11.4.2 Několik sekcí

Ne každý uživatel bude chtít ke svému programu instalovat dokumentaci (příručku uživatele a text licenčního ujednání) a samostatnou knihovnu pro vývoj tříd pro výpočet fraktálů. Rozdělíme tedy soubory tvořící instalaci programu Fraktálník do tří skupin. V první budou soubory tvořící vlastní aplikaci, ve druhé soubory tvořící dokumentaci a ve třetí knihovna spolu s dokumentací k ní.⁶ Tomu budou odpovídat tři sekce v instalačním programu. Dokumentaci a knihovnu implicitně nebudeme instalovat, takže před jména odpovídajících sekcí připojíme volbu `/o`. Naopak program by měl být instalován vždy; to zařídíme příkazem `SectionIn R0`, který způsobí, že vybrání této sekce nepůjde zrušit.

Následující výpis berte prosím jen jako ukázkou zacházení s více sekcemi; na smysluplné řešení instalace knihovny se podíváme v následujícím pododdílu.

```
#Soubor skripty\Fraktál5.nsi
Section "!Program Fraktálník" Sekce1
  SectionIn R0
  SetOutPath "$INSTDIR"
  File fraktal6_0_2.jar
  SetOutPath "$INSTDIR\ikona"
  File ikona\*. *
  SetOutpath "$INSTDIR\lib"
  File lib\Knihovna.jar
```

```
# ...Vytvoření zástupce na ploše zde vynecháme...
```

⁶To znamená, že knihovna pro obsahující komplexní čísla a rozhraní `VýpočetFraktálu` je zároveň ve dvou skupinách – jednak jako nezbytná součást aplikace, jež se instaluje vždy, jednak jako nástroj, jehož instalaci si uživatel volí. To má svůj smysl: Uživatel by neměl přistupovat k souborům, které jsou nezbytné pro běh aplikace, proto mu poskytneme samostatnou instalaci této knihovny. V adresáři se vzorovou instalací je ovšem jen jednou.

```

WriteUninstaller "$INSTDIR\Oinstaluj.exe"
SectionEnd

Section /o "Dokumentace" Sekce2
  SetOutPath "$INSTDIR\doc"
  File doc\*.*
SectionEnd

Section /o "Knihovna" Sekce3
# ... tohle vyřešíme v následujícím pododdílu lépe ...
  SetOutPath "$INSTDIR\knihovna"
  File /r knihovna\*.*
SectionEnd

```

Za poslední sekci budou následovat příkazy definující popis sekcí a jejich přiřazení sekcím:

```

# Popis sekcí
LangString DESC_Sekce1 ${LANG_Czech} "Program, fraktály a nápověda"
LangString DESC_Sekce2 ${LANG_Czech} \
  "Příručka uživatele a licenční ujednání"
LangString DESC_Sekce3 ${LANG_Czech} \
  "Knihovna pro vytváření vlastních tříd \
  fraktálů a dokumentace k ní"
# Přiřazení řetězců v daném jazyce sekcím
!insertmacro MUI_FUNCTION_DESCRIPTION_BEGIN
  !insertmacro MUI_DESCRIPTION_TEXT ${Sekce1} $(DESC_Sekce1)
  !insertmacro MUI_DESCRIPTION_TEXT ${Sekce2} $(DESC_Sekce2)
  !insertmacro MUI_DESCRIPTION_TEXT ${Sekce3} $(DESC_Sekce3)
!insertmacro MUI_FUNCTION_DESCRIPTION_END

```

Instalujeme knihovnu do jiného adresáře

Už jsme si řekli, že uživatel by neměl přistupovat k souborům, které jsou nezbytné pro fungování aplikace. To znamená, že musí mít svou kopii knihovny pro vytváření fraktálů uloženou pokud možno v jiném adresáři. V předchozím pododdílu jsme to neřešili, tam nám šlo pouze o první ukázkou skriptu s volitelnými a povinnými sekcemi. Nyní ale potřebujeme vytvořit skript, který – když si uživatel zvolí instalaci knihovny – od něj zjistí adresář, kam ji má umístit, potřebný adresář vytvoří a soubory do něj uloží.

K určení adresáře pro instalaci knihovny použijeme další stránku `MUI_PAGE_DIRECTORY`, kterou vložíme za stránku pro určení instalačního adresáře. Musíme samozřejmě upravit texty na ní, aby uživatel pochopil, že jde o adresář knihovny, nikoli vlastní aplikace, a přikázat uložení cesty získané touto stránkou do jiné proměnné než je `INSTDIR`. Vedle toho je třeba zařídit, aby se tato stránka objevila pouze v případě, že uživatel zbude chtít knihovnu instalovat.

Nejprve tedy deklarujeme globální proměnnou pro uložení cesty knihovny (jiné než globální proměnné nejsou v NSIS k dispozici):

```
Var Knihovna
```

Tuto deklaraci umístíme do úvodu skriptu, nejlépe před atributy. Pak mezi stránku pro zjištění adresáře aplikace a stránku informující o průběhu instalace zařadíme následující příkazy:

```

!define MUI_PAGE_HEADER_TEXT "Zvolte umístění knihovny"
!define MUI_PAGE_HEADER_SUBTEXT "Vyberte adresář, do kterého bude \
knihovna pro vývoj fraktálů instalována."
!define MUI_DIRECTORYPAGE_TEXT_TOP "Požadujete instalaci knihovny pro \
vlastní vývoj tříd pro výpočet fraktálů. Určete adresář, do něhož bude \
tato knihovna uložena."
# Nápis u textového pole obsahujícího zvolený adresář
!define MUI_DIRECTORYPAGE_TEXT_DESTINATION "Umístění knihovny pro vývoj \
fraktálů"
!define MUI_DIRECTORYPAGE_VARIABLE $Knihovna
# Tento příkaz zařídí případné přeskočení stránky
!define MUI_PAGE_CUSTOMFUNCTION_PRE zdaZobrazit
!insertmacro MUI_PAGE_DIRECTORY

```

Většinu příkazů, které jsme zde použili, už známe. Za zmínku stojí příkaz `MUI_PAGE_CUSTOMFUNCTION_PRE`, který příkazuje, aby se před zobrazením následující stránky zavolala z pozadí funkce `zdaZobrazit`. O funkcích a příkazech budeme podrobněji hovořit v následujícím oddílu, zde si pouze ukážeme její zdrojový text.

```

Function zdaZobrazit
  ${Unless} ${SectionIsSelected} ${Sekce3}
    Abort
  ${EndUnless}
  StrCpy $Knihovna "$INSTDIR\knihovna"
FunctionEnd

```

Tato funkce nejprve zjistí, zda je vybrána sekce instalace knihovny; jestliže není, provede příkaz `Abort`, který způsobí přeskočení této stránky. Je-li vybrána, uloží do proměnné `Knihovna` adresář, který stránka pro volbu adresáře knihovny nabídne uživateli jako výchozí.

Sekce pro instalaci knihovny pak bude mít tvar

```

Section /o "Knihovna" Sekce3
  SetOutPath "$Knihovna"
  File /r lib\*. *
SectionEnd

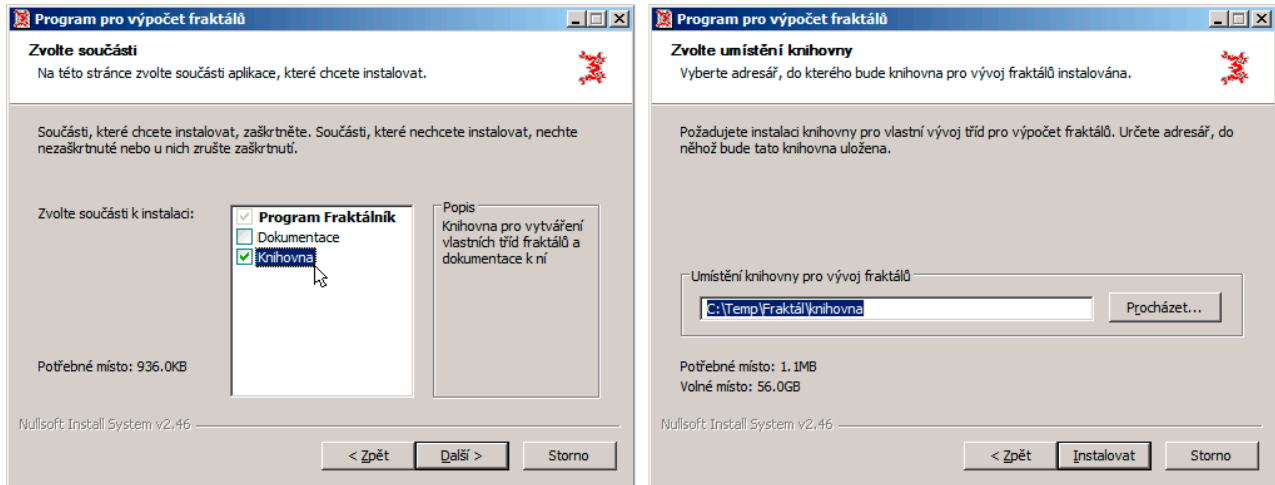
```

Zde kopírujeme z adresáře `lib` všechny soubory – tedy nejen archiv `Knihovna.jar`, ale i dokumentaci. Jestliže nyní takto upravený instalační skript přeložíme a spustíme, bude dialog pro volbu součástí vypadat podobně jako na obr. 11.10 vlevo. Když uživatel zaškrtně, že chce instalovat knihovnu, objeví se okno s dotazem na cílový adresář, které vidíte na obr. 11.10 vpravo. Tuto verzi instalačního skriptu najdete na webu v souboru `Fraktál6.nsi` v adresáři `skripty\Instalace3`.

11.4.3 Položka v nabídce Start

Naším dalším úkolem je vytvořit pro náš program položku v nabídce *Start* Windows. Tyto položky odpovídají zástupcům (souborům `.lnk`) uloženým ve vyhrazeném adresáři⁷ a skupiny

⁷Pod Windows 7 to je adresář `Programs`, který je uložen v adresáři `C:\Users\<uživatel>\AppData\Roaming\Microsoft\Windows\Start Menu`. V jiných verzích Windows může být tato cesta jiná.



Obrázek 11.10: Dokumentace a knihovna tvoří samostatné sekce, jež nejsou implicitně vybrány (vlevo); jestliže uživatel vybere instalaci knihovny, zobrazí průvodce po okně pro určení adresáře instalace programu okno s dotazem na adresář knihovny (vpravo)

programů v nabídce *Start* odpovídají podadresářům tohoto adresáře. Cestu k tomuto adresáři lze v instalačním skriptu získat pomocí předdefinované proměnné `$SMPROGRAMS`. Pro práci s ním je třeba administrátorské oprávnění.

To znamená, že mezi atributy zařadíme příkaz

```
RequestExecutionLevel admin
```

a do hlavní sekce, tedy do sekce, jež vypisuje vlastní program, přidáme příkazy

```
CreateDirectory "$SMPROGRAMS\Fraktálník"
CreateShortcut "$SMPROGRAMS\Fraktálník\Fraktálník.lnk" \
"$INSTDIR\fraktal6_0_2.jar" "" \
"$INSTDIR\ikona\Netopýr2.ico" 0
CreateShortcut "$SMPROGRAMS\Fraktálník\Odinstaluj Fraktálník.lnk" \
"$INSTDIR\Odinstaluj Fraktálník.exe" "" \
"$INSTDIR\ikona\Netopýr1.ico" 0
```

Do sekce odinstalátoru přidáme příkaz

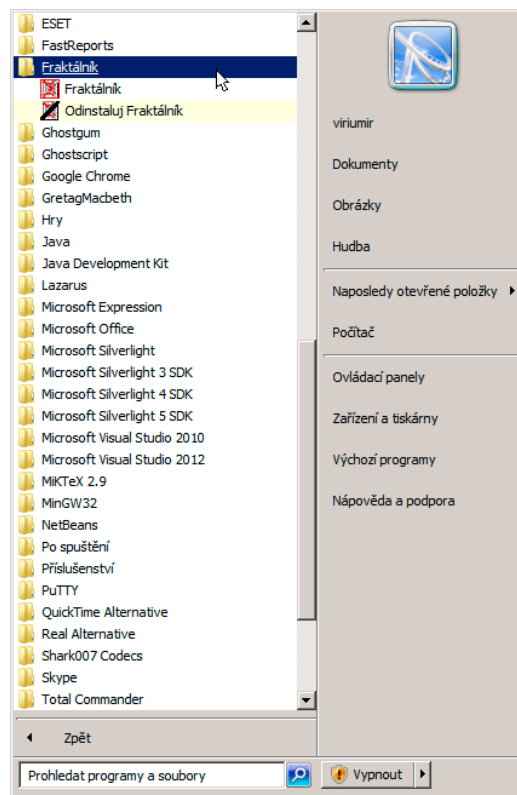
```
Rmdir /r "$SMPROGRAMS\Fraktálník"
```

kterým odstraníme adresář spolu s odkazy, jež jsou v něm uloženy.

Když takto upravený skript přeložíme a spustíme, zobrazí se v nabídce *Start* Windows skupina *Fraktálník* s položkami *Fraktálník* a *Odinstaluj Fraktálník* (viz obr. 11.11).

Poznámka

Windows by také měl automaticky zařadit první přidáný odkaz mezi často používané programy, tedy mezi programy, které se v nabídce *Start* zobrazí po jejím vyvolání ještě před volbou *Všechny programy*. Ve dvaatřicetibitové verzi Windows 7 mi to tak opravdu funguje; ve čtyřiašedesátibitové verzi mi však přidá mezi často používané programy odkaz na odinstalační program, a to bez ohledu na pořadí, ve kterém odkazy vytvořím.

Obrázek 11.11: Fraktálník a jeho odinstalační program v nabídce *Start*

11.4.4 Informace o programu v ovládacích panelech

Ovládací panel *Programy a funkce* ve Windows obsahuje přehled instalovaných programů; poklepáním na záznam o vybraném programu spustíme jeho odinstalaci. Je rozumné, aby tento přehled obsahoval i náš program.

Informace o instalovaných aplikacích jsou uloženy v registru Windows; to je, jak jistě víte, hierarchická databáze obsahující nejrůznější informace nezbytné pro správný chod operačního systému i jednotlivých aplikací. Informace zobrazované v panelu *Programy a funkce* jsou uloženy v části `HKEY_LOCAL_MACHINE` v uzlu se jménem aplikace, který je připojen k uzlu `Software\Microsoft\Windows\CurrentVersion\Uninstall`.

Aby tedy měl program Fraktálník svou položku v ovládacích panelech, musíme v registru vytvořit uzel *Fraktálník* a zapsat do něj alespoň dvě položky — jméno aplikace a cestu k odinstalačnímu programu. Tyto položky jsou určeny klíči *DisplayName* a *UninstallString*. Je ale rozumné zapsat tam ještě informaci o vydavateli (klíč *Publisher*) a o čísle verze (klíč *DisplayVersion*) a připojit ikonu programu, která se zobrazí v seznamu na panelu *Programy a funkce* (klíč *DisplayIcon*).

Vedle toho uložíme do uzlu `Software\Fraktálník` v části `HKEY_CURRENT_USER` informaci o instalačním adresáři naší aplikace.

Instalační program

K zápisu do registru Windows použijeme příkaz `WriteRegStr`, který má tvar

```
WriteRegStr HK uzel klíč hodnota
```

Zde *HK* je zkratka kořenového uzlu, např. HKLM pro uzel HKEY_LOCAL_MACHINE; *uzel* je znakový řetězec určující uzel, do kterého chceme zapisovat, *klíč* je jméno zapisované hodnoty a *hodnota* je řetězec, který chceme zapsat. Zápis do registru provedeme v hlavní sekci instalačního skriptu. Přidáme do ní příkazy

```
# Zápis adresáře s instalací do registru
WriteRegStr HKCU "Software\Fraktálník" "Install_Dir" $INSTDIR

# Zápis informací pro odinstalaci z ovládacích panelů do registru
WriteRegStr HKLM \
  "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník" \
  "DisplayName" "Fraktálník (lze pouze odstranit)"
WriteRegStr HKLM \
  "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník" \
  "DisplayVersion" "6.0.2.0"
WriteRegStr HKLM \
  "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník" \
  "Publisher" "Piškvorky software"
WriteRegStr HKLM \
  "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník" \
  "UninstallString" '$INSTDIR\Odinstaluj.exe'
WriteRegStr HKLM \
  "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník" \
  "DisplayIcon" '$INSTDIR\ikona\Netopýr2.ico'
```

Připomeňme si, že pro zápis do registru Windows musí mít instalační program administrátorské oprávnění. To jsme však nastavili už v souvislosti s vytvářením položky v nabídce *Start*.

Poznámky

- Všimněte si zápisu '\$INSTDIR\ikona\Netopýr2.ico' v poslední řádce výpisu. Jde o řetězec, který obsahuje uvozovky a je uzavřen do apostrofů. To znamená, že do registru se uloží řetězec představující cestu souboru ikony včetně uvozovek. To je nezbytné, neboť cesta vyjádřená tímto řetězcem může obsahovat mezeru — například když uživatel instaluje aplikaci do adresáře Program Files.
- Příkaz `WriteRegStr` zapíše do zadaného klíče registru znakový řetězec. Chceme-li zapsat celé číslo, použijeme příkaz `WriteRegDWORD`; pro zápis binárních dat poslouží příkaz `WriteRegBin`. Podrobnější informace o nich najdete v dokumentaci.

Odinstalační program

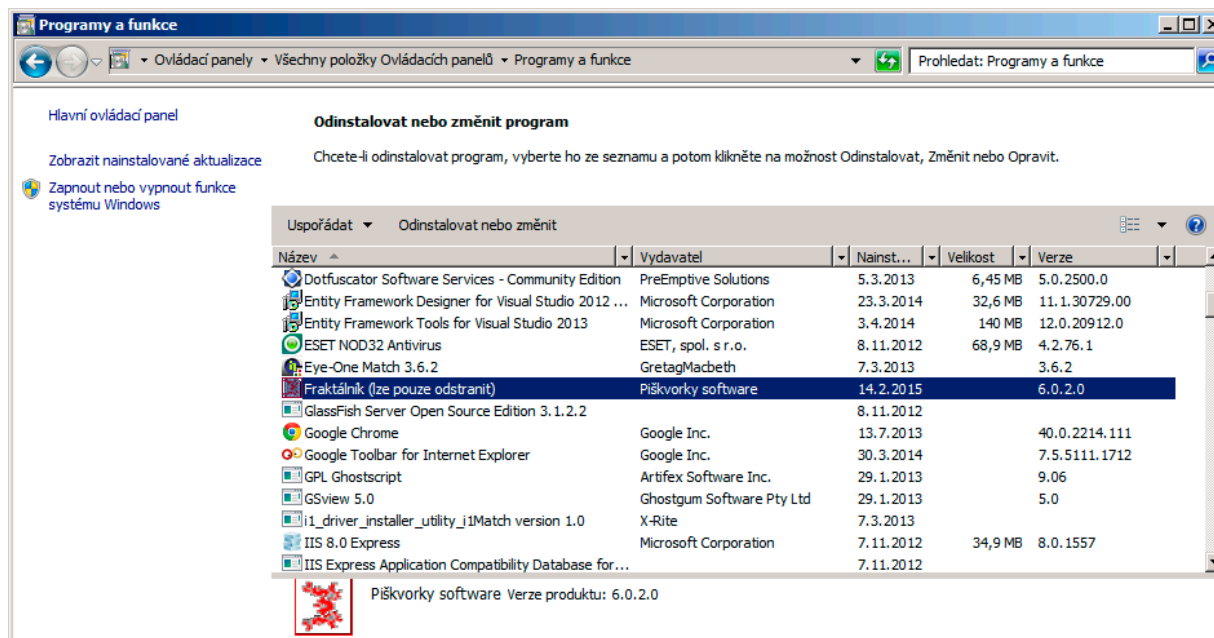
Odinstalační program musí tyto zápisy z registru odstranit. Toho dosáhneme odstraněním celého uzlu *Fraktálník* z registru příkazem `DeleteRegKey`, který má tvar

```
DeleteRegKey HK uzel
```

Zde *HK* je opět zkratka kořenového uzlu registru a *uzel* je jméno uzlu, který chceme odstranit. To znamená, že do sekce odinstalátoru našeho skriptu umístíme příkazy

```
DeleteRegKey HKCU "Software\Fraktálník"
DeleteRegKey HKLM \
    "Software\Microsoft\Windows\CurrentVersion\Uninstall\Fraktálník"
```

Jestliže náš instalační skript po těchto úpravách přeložíme a spustíme, bude program Fraktálník zobrazen na ovládacím panelu *Programy a funkce* (obr. 11.12) a poklepaním na tento záznam se spustí jeho odinstalace.



Obrázek 11.12: Záznam o programu Fraktálník v ovládacích panelech Windows

Nyní můžeme zrušit položku *Odinstalovat Fraktálník* v nabídce *Start Windows*. Odinstalace programu není obvyklá záležitost, takže tuto možnost není třeba v nabídce poskytovat – stačí, když je k dispozici v ovládacích panelech.

Instalační skript po posledních úpravách najdete na webu v adresáři `skripty\Instalace3` v souboru `Fraktálník6.nsi` spolu se vzorovou instalací.

Instalace do adresáře Program Files

Zatím jsme neuvažovali o instalaci do adresáře *Program Files*. K zápisu do tohoto adresáře je třeba administrátorské oprávnění; to nepředstavuje problém z hlediska instalačního programu, ale z hlediska běhu naší aplikace, neboť program *Fraktálník* zapisuje do instalačního adresáře svůj konfigurační soubor, instaluje do něj přidané fraktály atd.

Jedno z možných řešení je, že soubory, které instalované aplikace mění, ukládáme do zvláštního adresáře vyhrazeného pro data aplikací; pod Windows 7 to je typicky adresář `Users\<uživatel>\AppData\Roaming`; v jiných verzích Windows může být jiný. V Javě získáme řetězec obsahující cestu tohoto adresáře voláním funkce `System.getenv()`, které jako parametr předáme řetězec `"APPDATA"`.⁸ Ve skriptu NSIS je uvedený řetězec uložen v předdefinované proměnné `$APPDATA`.

⁸Připomeňme si, že statická metoda `System.getenv()` očekává řetězec obsahující jméno proměnné prostředí a vrátí řetězec obsahující její hodnotu. Pod Windows 7 je cesta k uvedenému adresáři v proměnné `APPDATA`; cesta k adresáři s uživatelským profilem, jako je `C:\Users\<uživatel>`, je v proměnné prostředí `USERPROFILE`.

11.5 Instalace společně s Javou

Aby byl instalační program opravdu použitelný, musí umět zabezpečit, že na cílovém počítači bude instalována správná verze běhového prostředí Javy. To bude naším cílem v posledním oddílu této kapitoly.

11.5.1 Předběžné úvahy

Než se pustíme do práce na nové verzi instalačního balíčku, musíme si ujasnit, co je naším cílem. Instalační program musí zjistit, zda je na cílovém počítači instalováno běhové prostředí Javy (JRE), a pokud ano, jaká verze. Jestliže tam JRE není nebo jestliže je k dispozici verze starší než 1.8, nabídne uživateli instalaci potřebné verze, a když uživatel nabídku přijme, instaluje ji. Když ji odmítne, instalační program skončí, neboť pak nemá smysl Fraktálník instalovat.

Musíme také získat instalační program odpovídající verze JRE (pro 64bitovou verzi Windows to je soubor `jre-8u31-windows-x64.exe`). Stáhneme si ho ze stránky [10].

Jak zjistíme přítomnost JRE

Protože pracujeme pod Windows, budeme hledat informace o Javě v registru. Zde máme dvě možnosti:

- Na cílovém počítači je pouze JRE. Záznam o něm je ve větvi `HKEY_LOCAL_MACHINE` v uzlu `SOFTWARE\JavaSoft\Java Runtime Environment` pod klíčem `CurrentVersion`.
- Na cílovém počítači je JDK. Záznam o něm je ve opět větvi `HKEY_LOCAL_MACHINE`, avšak v uzlu `SOFTWARE\JavaSoft\Java Development Kit` pod klíčem `CurrentVersion`.

Musíme vyzkoušet obě možnosti. Jestliže záznam v registru najdeme, zjistíme, zda existuje soubor `java.exe`, tedy zda záznam v registru nelže. (Uživatel mohl např. adresář s instalací Javy omylem smazat.) Cesta k tomuto souboru je ve stejném uzlu registru, ovšem pod klíčem `JavaHome`.

Porovnání verzí

K přečtení verze JRE použijeme funkci `ReadRegStr`, jež uloží do předané proměnné znakový řetězec obsahující číslo verze – např. "1.7.0_09" – nebo prázdný řetězec, jestliže požadovaný uzel nebo klíč neexistuje.

Jestliže existuje, získáme z něj nejprve hlavní číslo instalované verze (v našem příkladu to je 1) a porovnáme ho s hlavním číslem verze požadované instalace (to je také 1). Jestliže bude hlavní číslo verze u obou řetězců stejné (a to v bude), získáme z obou řetězců vedlejší číslo verze a rozhodneme podle něj.

11.5.2 Nástroje, které budeme potřebovat

Pro zjištění, zda je instalována potřebná verze Javy, budeme muset napsat poměrně komplikované funkce, takže si musíme říci několik slov o funkcích v NSIS a o příkazech, které použijeme.

NSIS rozlišuje dva druhy funkcí: Tzv. uživatelské funkce voláme v jiných funkcích nebo v sekcích příkazem `Call`. Funkce volané „z pozadí“ (callback) nevoláme, o jejich volání rozhoduje přeložený skript podle jistých pravidel sám.

Deklarace funkce

Deklarace funkce začíná klíčovým slovem `Function`, za nímž následuje identifikátor funkce. Končí klíčovým slovem `FunctionEnd`. V deklaraci neuvádíme seznam parametrů; veškeré parametry uložíme před voláním do zásobníku příkazy `Push` a funkce si je odtud vyzvedne pomocí příkazů `Pop`. Výsledek funkce zpravidla také uložíme do zásobníku.

K zavolání uživatelské funkce slouží příkaz `Call`, za kterým uvedeme jméno funkce. Funkce skončí provedením posledního příkazu v těle nebo provedením příkazu `return`.

Jména funkcí volaných z pozadí (callback) začínají tečkou. Mezi nimi má zvláštní postavení funkce `.onInit`, kterou instalační program zavolá na počátku běhu.

Práce se zásobníkem

K uložení hodnoty na vrchol zásobníku slouží příkaz

```
Push hodnota
```

kde *hodnota* představuje znakový řetězec buď zapsaný přímo nebo uložený v nějaké proměnné. K vyjmutí řetězce ze zásobníku a jeho uložení do *proměnné* slouží příkaz

```
Pop proměnná
```

```
Příkaz
```

```
Exch
```

může být bez parametrů nebo s jedním parametrem. Je-li bez parametrů, prohodí hodnotu na vrcholu zásobníku s hodnotou pod ní. Uvedeme-li jako parametr proměnnou, prohodí obsah této proměnné s hodnotou na vrcholu zásobníku.

Řízení toku výpočtu

K řízení toku výpočtu se v NSIS používají převážně podmíněné a nepodmíněné příkazy skoku a návěští. Lze také použít předdefinovaná makra, která fungují podobně jako některé podmíněné příkazy z vyšších programovacích jazyků.

```
Příkaz
```

```
Goto návěští
```

představuje nepodmíněný skok, který příkazuje přejít na příkaz označený *návěští*. *Návěští* je identifikátor a představuje pojmenování místa v programu. Od následujícího příkazu ho oddělujeme dvojtečkou. Místo návěští můžeme v příkazu `Goto` uvést i celé číslo, které říká, o kolik příkazů má program přejít dopředu nebo dozadu.

```
Příkaz
```

```
StrCmp řetězec1 řetězec2 když_rovno když_nerovno
```

porovná *řetězec1* a *řetězec2*. Jsou-li si rovny, přejde na místo určené parametrem *když_rovno*, jinak přejde na místo určené parametrem *když_nerovno*. Jak *když_rovno*, tak *když_nerovno* mohou být návěští nebo celá čísla. Význam celých čísel je stejný jako u příkazu `Goto`. Při porovnávání se nerozlišují malá a velká písmena. Parametr *když_nerovno* může chybět; pak v případě nerovnosti program pokračuje následujícím příkazem.

Pro porovnávání celých čísel slouží příkaz

```
IntCmp h1 h2 když_rovno když_h1_menší když_h1_větší
```

který porovná celočíselné hodnoty *h1* a *h2* (zadané jako řetězce). Jsou-li si rovny, přejde na místo určené parametrem *když_rovno*; je-li *h1* menší než *h2*, přejde na místo určené parametrem *když_h1_menší* a je-li *h1* větší než *h2*, přejde na místo určené parametrem *když_h1_větší*. Tyto parametry opět mohou být buď návěští, nebo celá čísla. Význam celých čísel je stejný jako u příkazu *Goto*. Třetí, nebo druhý a třetí parametr může chybět – pak program pokračuje následujícím příkazem.

Příkaz

```
IfFileExists cesta nalezen nenalezen
```

umožňuje zjistit, zda soubor nebo složka určená *cestou* existuje; podle toho bude program pokračovat označeným příkazem. I zde platí, že *nalezen* a *nenalezen* mohou být buď návěští, nebo celá čísla.

Příkaz *Abort* v uživatelské funkci ukončí instalační skript. Ve funkcích volaných z pozadí způsobí přeskočení stránky průvodce, která tuto funkci zavolala.

Chceme-li provést skupinu *příkazů*, je-li splněna daná *podmínka*, použijeme makra

```
#{If} podmínka
    příkazy
#{EndIf}
```

Chceme-li skupinu *příkazů* provést, jestliže daná *podmínka* není splněna, použijeme makra *#{IfNot}*, (resp. *#{Unless}*, jež znamená totéž); skupina podmíněně prováděných příkazů je v nich ukončena *{EndIfNot}* (resp. *{EndUnless}*). S příkladem jsme se setkali v závěru oddílu 11.4.2 ve funkci *zdaZobrazit*.

Uživatelské okno

Když náš instalační skript zjistí, že na cílovém počítači není instalována správná verze Javy, musí to uživateli oznámit a zeptat se ho, zda si přeje Javu instalovat nebo zda si přeje instalaci Fraktálníku ukončit. K tomu poslouží uživatelem vytvořené okno. To musíme nejprve popsat v souboru *.ini*. Popis souboru *jre.ini* pro náš instalační skript může mít tvar

```
[Settings]
NumFields=1

[Field 1]
Type="Label"
Left=0
Right=500
Top=0
Bottom=40
```

V první sekci, nadepsané *Settings*, určíme počet polí (oblastí) na uživatelském okně, do nichž chceme vložit nějaká data; zde uvádíme jedno. Pak následují popisy těchto polí. Zde říkáme, že půjde o textové pole (*Label*), a uvádíme souřadnice levého horního a pravého dolního rohu.

Tento soubor načteme např. ve funkci *.onInit*. Okno pak vytvoříme příkazem

```
Page custom funkce
```

který uvedeme na odpovídajícím místě mezi příkazy pro vložení stránek průvodce. *Funkce* se postará o funkčnost této stránky.

Definice makra

Makra bez parametrů definujeme příkazem

```
!define jméno hodnota
```

kde *jméno* je jméno, jímž se na toto makro budeme ve skriptu odvolávat, a *hodnota* je znakový řetězec, jímž bude *jméno* ve skriptu nahrazeno. Makro použijeme zápisem `#{jméno}`, který překladač nahradí odpovídající *hodnotou*.

Některé další příkazy

Pro spuštění programu instalujícího JRE použijeme příkaz

```
ExecWait program parametry proměnná
```

Tento příkaz spustí *program*, předá mu *parametry*, jsou-li uvedeny, a kód jeho ukončení uloží do *proměnné*. Skript, který takto spustil externí program, bude pokračovat až po jeho ukončení.

Při ladění skriptů, ale i při jiných příležitostech oceníme okna se zprávami. Nejjednodušší okno se zprávou vytvoříme příkazem

```
MessageBox MB_OK zpráva
```

11.5.3 Předběžné úpravy instalačního skriptu

Než se pustíme do vytváření konečné verze instalačního skriptu, zřehledníme jeho stávající zdrojový text a uděláme ještě další drobné úpravy.

Zpřehlednění zdrojového textu

Jméno programu, požadovaná verze JRE a další údaje se ve skriptu opakují na více místech; abychom je mohli později snadno změnit, definujeme je v úvodu jako makra:

```
!define Jméno "Fraktálník"
!define Verze "6.0.2"
!define KrátkéJméno #{Jméno}
!define VERZE_JRE "1.8.0"
!define Dodavatel "Piškvorcky software"
!define JreInstProgram "jre-8u31-windows-x64.exe"
```

Zde jsme zavedli dvě makra pro jméno naší aplikace; zatím budou obě stejná, *Fraktálník*, ale v budoucnu se můžeme rozhodnout používat v určitých situacích např. pojmenování *Úžasně fraktály* jako plné jméno a *Fraktálník* jako krátké jméno.

Mezi makry je i jméno instalačního programu 64bitové verze JRE 1.8 pro Windows.

Nyní tato makra použijeme v textech stávajících stránek průvodce. Například příkaz určující titulek úvodní stránky průvodce instalací bude mít tvar

```
!define MUI_WELCOME_PAGE_TITLE "Vítejte v průvodci instalací programu \
#{KrátkéJméno}"
```


Vedle toho deklaruje dvě nové globální proměnné

```
Var InstalovatJRE
Var CestaJRE
```

Jejich jména naznačují jejich účel.

Úprava sekcí

Ne každý uživatel bude chtít vytvořit zástupce programu na ploše a položku v nabídce *Start*. Proto umístíme odpovídající příkazy do samostatných sekcí, které budou implicitně vybrány.

Sekci instalující program Fraktálník ponecháme jako implicitně vybranou, nezrušitelnou a tučně vyznačenou.

11.5.4 Je tedy na cílovém počítači instalována Java?

Nyní se můžeme pustit do zjišťování, zda je na cílovém počítači instalováno běhové prostředí Javy potřebné verze. Tento úkol svěříme funkci *DetekceJRE*, jež předpokládá, že na vrcholu zásobníku je řetězec obsahující číslo požadované verze.

```
Function DetekceJRE
    Exch $0      ; Zjistí požadovanou verzi; teď je předchozí hodnota
                ; $0 v zásobníku a požadovaná verze JDK je v $0
    ; Zde uložíme hodnoty proměnných před vstupem do této funkce
    Push $1     ; $1 bude obsahovat řetězec se zjištěnou verzí Javy (např. 1.5.0)
    Push $2     ; $2 bude obsahovat adresář Javahome
    Push $3     ; $3 a $4 se použijí pro kontrolu hlavní/vedlejší verze Javy
    Push $4

    ReadRegStr $1 HKLM \
        "SOFTWARE\JavaSoft\Java Runtime Environment" "CurrentVersion"
    StrCmp $1 "" Detekce2
    ReadRegStr $2 HKLM
        "SOFTWARE\JavaSoft\Java Runtime Environment\$1" "JavaHome"
    StrCmp $2 "" Detekce2
    Goto PorovnejVerze

Detekce2:
    ReadRegStr $1 HKLM \
        "SOFTWARE\JavaSoft\Java Development Kit" "CurrentVersion"
    StrCmp $1 "" Nenalezeno
    ReadRegStr $2 HKLM \
        "SOFTWARE\JavaSoft\Java Development Kit\$1" "JavaHome"
    StrCmp $2 "" Nenalezeno

PorovnejVerze:
; $0 = požadovaná verze, $1 = nalezená verze, $2 = javaHome
IfFileExists "$2\bin\java.exe" 0 Nenalezeno
StrCpy $3 $0 1      ; Zjistí hlavní číslo verze.
```

```

;Např.: $1 = 1.5.0, nyní $3 = 1
StrCpy $4 $1 1 ; $3 = hlavní verze požadovaná,
; $4 = hlavní verze nalezená
IntCmp $4 $3 0 NalezenaStará NalezenaNová
StrCpy $3 $0 1 2
StrCpy $4 $1 1 2 ; Jako předtím. $3 je vedlejší požadovaná verze,
; $4 je vedlejší instalovaná verze
IntCmp $4 $3 NalezenaNová NalezenaStará NalezenaNová

Nenalezeno:
Push "0"
Goto KonecDetekceJRE

NalezenaStará:
Push "-1"
Goto KonecDetekceJRE

NalezenaNová:
Push "$2\bin\java.exe"
Goto KonecDetekceJRE

KonecDetekceJRE:
; Na vrcholu zásobníku je návratová hodnota (nh), pak $4, $3, $2, $1
Exch ; => $4, nh, $3, $2, $1, $0
Pop $4 ; => nh, $3, $2, $1r, $0
Exch ; => $3, nh, $2, $1, $0
Pop $3 ; => nh, $2, $1, $0
Exch ; => $2, nh, $1, $0
Pop $2 ; => nh, $1, $0
Exch ; => $1, nh, $0
Pop $1 ; => nh, $0
Exch ; => $0, nh
Pop $0 ; => nh
FunctionEnd

```

Nejprve umístíme požadovanou verzi do proměnné \$0 a připravíme si další 3 pomocné proměnné (uložíme jejich stávající hodnoty do zásobníku). Pak zkusíme přečíst z registru číslo aktuální verze JRE a uložit ho do \$1. Když funkce `ReadRegStr` neuspěje, vrátí prázdný řetězec, a to znamená, že JRE není k dispozici. Vrácenou hodnotu tedy porovnáme s "" a v případě rovnosti přejdeme k dalšímu pokusu o nalezení Javy (skočíme na návěští `Detekce2`).

Jestliže funkce `ReadRegStr` vrátí neprázdný řetězec, zkusíme přečíst adresář s instalací a uložit ho do \$2. Když zde dostaneme prázdný řetězec, znamená, to, že JRE není k dispozici a přejdeme opět na návěští `Detekce2`.

Jestliže jsou v obou proměnných, \$1 i \$2, uloženy neprázdné řetězce, znamená to, že běhové prostředí Javy je na cílovém počítači již instalováno a my potřebujeme zjistit, zda jeho verze vyhovuje našim požadavkům. Přejdeme tedy do části funkce `DetekceJRE` označené návěštím `PorovnejVerze`.

Do části uvedené návěstím `Detekce2` se dostaneme pouze v případě, že neuspějeme při pokusu najít v registru záznam o JRE. Zde se podobným způsobem pokusíme o detekci JDK. Jestliže ani zde neuspějeme, znamená to, že Java na cílovém počítači není a přejdeme do části uvedené návěstím `Nenalezeno`, jinak přejdeme do části uvedené návěstím `PorovnejVerze`.

V části uvedené návěstím `PorovnejVerze` nejprve zkontrolujeme, zda instalace Javy opravdu existuje, tedy zda existuje soubor `java.exe`, jehož cesta je uložena v `$2`. Jestliže neexistuje, přejdeme na návěští `Nenalezeno`, neboť přes ujištění registru Windows Java na cílovém počítači není k dispozici. Jinak pomocí funkce `StrCpy` získáme z řetězců uložených v `$0` a `$1` hlavní číslo požadované a nalezené verze, uložíme si je do `$3` a `$4` a porovnáme je. Dokážeme-li rozhodnout už podle hlavního čísla verze, přejdeme na návěští `NalezenaNová` nebo `NalezenaStará`, jinak získáme vedlejší číslo verze a rozhodneme podle něj.

Jestliže byla nalezena požadovaná nebo novější verze Javy, uložíme do zásobníku cestu souboru `java.exe`; byla-li nalezena starší verze, uložíme na vrchol zásobníku řetězec `"-1"` a nebyla-li Java nalezena vůbec, uložíme tam `"0"`. Ve všech případech přejdeme k závěrečným operacím, tedy na návěští `KonecDetekceJRE`. Tam obnovíme původní hodnoty proměnných (vyjmeme je ze zásobníku).

11.5.5 Kontrola instalace Javy a odpovídající stránka průvodce

Průvodce musí zkontrolovat, zda je k dispozici odpovídající verze Javy, a v případě, že není, nabídnout její instalaci. Nejdříve musí ovšem načíst soubor `jre.ini`, aby mohl vytvořit odpovídající stránku.

Funkce `.onInit`

O načtení souboru `jre.ini`, který popisuje základní uspořádání dialogu s informacemi o instalaci Javy, se postaráme ve funkci `.onInit`, kterou skript volá ještě před zobrazením první stránky průvodce. Její zdrojový text bude jednoduchý.

```
Function .onInit
    !insertmacro INSTALLOPTIONS_EXTRACT "jre.ini"
FunctionEnd
```

Nové stránky průvodce

Ke kontrole Javy poslouží uživatelem definovaná stránka, za níž bude následovat stránka informující o případném průběhu instalace. Kontrolu instalace svěříme funkci `KontrolaInstalaceJRE`, kterou napíšeme dále. Za vložení stránky s licenčním ujednáním tedy přidáme příkazy

```
Page custom KontrolaInstalaceJRE
!define MUI_PAGE_HEADER_TEXT "Instalace běhového prostředí Javy"
!define MUI_PAGE_HEADER_SUBTEXT "Prosím počkejte na dokončení \
    instalace Javy"
!insertmacro MUI_PAGE_INSTFILES
```

Funkce KontrolaInstalaceJRE

V této funkci nejprve vložíme na vrchol zásobníku číslo požadované verze Javy a zavoláme funkci DetekceJRE. Po návratu vyjmemme výsledek ze zásobníku, porovnáme ho s řetězcem "0" a "-1" a podle toho přejdeme na odpovídající návěští.

Je-li je nalezena zastaralá verze Javy, vytvoříme text stránky průvodce s odpovídajícími informacemi a se žádostí rozhodnutí, zda instalovat JRE. Podobně se zachováme v případě, že Java nebyla vůbec nalezena, pouze text zobrazené informace se bude lišit. V obou případech přejdeme do části označené návěštím NutnoInstalovatJRE.

Jestliže byla nalezena potřebná instalace Javy, přejdeme do části označené návěštím JREJižInstalována.

V části NutnoInstalovatJRE uložíme do proměnné InstalovatJRE řetězec "ano", v části JREJižInstalována řetězec "ne"; pak obnovíme hodnotu proměnné \$0 a ukončíme funkci. Podívejme se na její zdrojový kód.

```
Function KontrolaInstalaceJRE
  Push "${VERZE_JRE}"
  Call DetekceJRE
  Exch $0 ; Získej hodnotu ze zásobníku
  StrCmp $0 "0" Nenalezeno
  StrCmp $0 "-1" NalezenaStará
  Goto JREJižInstalována
```

NalezenaStará:

```
!insertmacro INSTALLOPTIONS_WRITE "jre.ini" "Field 1" "Text" \
  "${KrátkéJméno} vyžaduje novější verzi běhového prostředí Javy, \
  než jaká byla zjištěna\ na vašem počítači. Stisknete-li tlačítko \
  Instalovat, bude spuštěna instalace JRE ${VERZE_JRE}. \n\
  Stisknutím tlačítka Storno zrušíte instalaci JRE i aplikace \
  ${KrátkéJméno}."
!insertmacro MUI_HEADER_TEXT "$(TITULEK_JRE)" "$(PODTITULEK_JRE)"
!insertmacro INSTALLOPTIONS_DISPLAY_RETURN "jre.ini"
Goto NutnoInstalovatJRE
```

Nenalezeno:

```
!insertmacro INSTALLOPTIONS_WRITE "jre.ini" "Field 1" "Text" \
  "Na vašem počítači nebyla nalezena Java. Bude spuštěna instalace \
  JRE ${VERZE_JRE}. \nStisknutím tlačítka Storno zrušíte instalaci \
  JRE i aplikace ${KrátkéJméno}."
!insertmacro MUI_HEADER_TEXT "$(TITULEK_JRE)" "$(PODTITULEK_JRE)"
!insertmacro INSTALLOPTIONS_DISPLAY_RETURN "jre.ini"
Goto NutnoInstalovatJRE
```

NutnoInstalovatJRE:

```
Exch $0 ; $0 nyní má stránka installoptions návratovou hodnotu
Pop $0 ; Obnov $0
StrCpy $InstalovatJRE "ano"
Return
```

```

JREJižInstalována:
  StrCpy $InstalovatJRE "ne"
  StrCpy $CestaJRE $0
  !insertmacro INSTALLOPTIONS_WRITE "jre.ini" "UserDefinedSection" \
    "JREPath" $CestaJRE
  Pop $0 ; Obnov $0
  Return
FunctionEnd

```

V příkazech definujících záhlaví okna se odvoláváme na řetězce \$(TITULEK_JRE) a \$(POD-TITULEK_JRE). Jejich hodnotu definujeme příkazy

```

LangString TITULEK_JRE ${LANG_Czech} "Běhové prostředí Javy"
LangString PODTITULEK_JRE ${LANG_Czech} "Pro správnou funkci aplikace \
  ${KrátkéJméno} je potřebná instalace JRE ${VERZE_JRE}"

```

které uvedeme u popisu sekcí.

11.5.6 Sekce pro instalaci Javy

Sekci pro detekci a případnou instalaci Javy deklaruujeme jako první, neboť tyto operace musejí předcházet vlastní instalaci Fraktálníku. Před jejím spuštěním už proběhla funkce `Kontrola-InstalaceJRE`, jež uložila do proměnné `InstalovatJRE` řetězec s informací, zda je instalace potřebná, a pokud našla vyhovující verzi JRE, uložila také cestu k němu do proměnné `CestaJRE`. Tuto sekci deklaruujeme jako skrytou, tj. neobjeví se v mezi možnostmi v okně pro volbu součástí. Podívejme se na její zdrojový kód.

```

Section -InstalaceJre jre
  Push $0
  Push $1
  Strcmp $InstalovatJRE "ano" InstalujJRE Konec
  DetailPrint "Spouštění instalace JRE"

InstalujJRE:
  File /oname=$TEMP\${JreInstProgram} ${JreInstProgram}
  DetailPrint "Instalace JRE spuštěna"
  ExecWait '$TEMP\${JreInstProgram}' /s /v\"/qn REBOOT=Suppress \
    JAVAUPDATE=0 WEBSTARTICON=0\"' $0
  DetailPrint "Instalace JRE dokončena"
  Delete "$TEMP\${JreInstProgram}"
  StrCmp $0 "0" OvěřeníInstalace 0
  Push "Instalace JRE byla abnormálně ukončena."
  Goto ZrušInstalaciJRE

OvěřeníInstalace:
  DetailPrint "Kontrola výsledku instalace JRE"
  Push "${VERZE_JRE}"
  Call DetekceJRE

```

```
Pop $0 ; Zjištění hodnoty JRE
StrCmp $0 "0" ZrušInstalaciJRE 0
StrCmp $0 "-1" ZrušInstalaciJRE 0
Goto Konec
Push "Instalace JRE se nezdařila"
Goto ZrušInstalaciJRE
```

ZrušInstalaciJRE:

```
Pop $1
MessageBox MB_OK "Instalace bude ukončena z důvodu : $1"
Pop $1 ; Restore $1
Pop $0 ; Restore $0
Abort
```

Konec:

```
Pop $1 ; Restore $1
Pop $0 ; Restore $0
SectionEnd
```

V této sekci nejprve uložíme stávající hodnoty proměnných \$0 a \$1 do zásobníku. Pak zjistíme obsah globální proměnné `InstalovatJRE`, a jestliže neobsahuje řetězec "ano", není třeba JRE instalovat a přejdeme na návěští `Konec`; v opačném případě přejdeme na návěští `InstalujJRE`.

V části označené návěštím `InstalujJRE` uložíme příkazem `File` do pomocného adresáře určeného předdefinovanou konstantou `$TEMP` instalační soubor `JRE`, spustíme ho a počkáme na výsledek instalace. Kód ukončení programu bude v proměnné \$0. Pak instalační soubor vymažeme a podle kódu ukončení rozhodneme, co dále. Jestliže se instalace z jakéhokoli důvodu nezdařila, přejdeme na návěští `ZrušInstalaciJRE`, jinak přejdeme na návěští `OvěřeníInstalace`.

V části označené návěštím `OvěřeníInstalace` uložíme do zásobníku číslo verze, zavoláme funkci `DetekceJRE` a pokud tato funkce uspěje, přejdeme do části označené návěštím `Konec`. Jestliže v této části zjistíme, že Java není k dispozici, tedy že se instalace nezdařila, přejdeme na návěští `ZrušInstalaciJRE`.

V části `ZrušInstalaciJRE` obnovíme hodnoty proměnných \$0 a \$1, zobrazíme okno se zprávou o neúspěchu a instalaci příkazem `Abort` ukončíme.

V části nadepsané `Konec` obnovíme hodnoty proměnných.

Poznamenejme, že po úspěšném ukončení instalační sekce bude globální proměnná `CestaJRE` vždy obsahovat cestu souboru `java.exe`. V současné verzi ji sice nepotřebujeme, ale lze si představit rozšíření naší úlohy, kdy bude nutné mít k tomuto souboru přístup.

Úplný zdrojový text tohoto instalačního skriptu, doplněný o zobrazování řady oken se zprávami o postupu, najdete na www v souboru `Fraktál7.nsi` v adresáři `skripty`. Poznamenejme, že tento příklad vznikl přepracováním a přizpůsobením skriptu uveřejněného na webu [29].

Jestliže vytvořený instalační balíček spustíme na počítači, na němž je instalována starší verze Javy, objeví se po odsouhlasení licenčního ujednání okno, které vidíte na obr. 11.13 vlevo. Stiskneme-li tlačítko *Instalovat*, objeví se okno s ukazatelem postupu. Po stisknutí tlačítka *Zobrazit detaily* uvidíme zprávy o jednotlivých událostech vypisované příkazy `DetailPrint` (obr. 11.13 vpravo).

