



příklady
ke stažení na

WWW.GRADA.CZ

Cocoa

úvod do programování počítačů Apple

Ondřej Čada

- Základy programování v Cocoa
- Stručný průvodce jazykem C, objekty a Objective C, prostředí Xcode
- Visuální programování s Interface Builderem
- Standardní objektové knihovny



Upozornění pro čtenáře a uživatele této knihy

Všechna práva vyhrazena. Žádná část této tištěné či elektronické knihy nesmí být reprodukována a šířena v papírové, elektronické či jiné podobě bez předchozího písemného souhlasu nakladatele. Neoprávněné užití této knihy bude **trestně stíháno**.

Používání elektronické verze knihy je umožněno jen osobě, která ji legálně nabyla a jen pro její osobní a vnitřní potřeby v rozsahu stanoveném autorským zákonem. Elektronická kniha je datový soubor, který lze užívat pouze v takové formě, v jaké jej lze stáhnout s portálu. Jakékoliv neoprávněné užití elektronické knihy nebo její části, spočívající např. v kopírování, úpravách, prodeji, pronajímání, půjčování, sdělování veřejnosti nebo jakémkoliv druhu obchodování nebo neobchodního šíření je zakázáno! Zejména je zakázána jakákoliv konverze datového souboru nebo extrakce části nebo celého textu, umístování textu na servery, ze kterých je možno tento soubor dále stahovat, přitom není rozhodující, kdo takovéto sdílení umožnil. Je zakázáno sdělování údajů o uživatelském účtu jiným osobám, zasahování do technických prostředků, které chrání elektronickou knihu, případně omezují rozsah jejího užití. Uživatel také není oprávněn jakkoliv testovat, zkoušet či obcházet technické zabezpečení elektronické knihy.





Copyright © Grada Publishing, a.s.

Cocoa

úvod do programování počítačů Apple

Ondřej Čada

Vydala Grada Publishing, a.s.
U Průhonu 22, Praha 7
jako svou 3605. publikaci

Odpovědný redaktor Tomáš Vild
Sazba Tomáš Vild
Návrh a grafická úprava obálky Vojtěch Kočí
Počet stran 200
První vydání, Praha 2009

© Grada Publishing, a.s., 2009
Cover Photo © fotobanka allphoto

V knize použité názvy programových produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Vytiskly Tiskárny Havlíčkův Brod, a.s.
Husova ulice 1881, Havlíčkův Brod

ISBN 978-80-247-2778-3 (tištěná verze) © Grada Publishing, a.s. 2011
ISBN 978-80-247-6356-9 (elektronická verze ve formátu PDF) © Grada Publishing, a.s. 2011

Obsah

Úvod	13
Co očekáváme od čtenáře?	13
Stručný přehled obsahu	13
Něco málo o historii	14
1. Malá ochutnávka na začátek: jak se v Kakau programuje?	15
1.1 Instalace vývojového prostředí	16
1.2 Pro začátek si vystačíme s Xcode	17
1.3 První – a nejjednodušší – program... ..	17
1.4 Čtečka RSS v tuctu programových řádků	20
1.5 Webový prohlížeč zcela bez programování	22
2. Vývojové prostředí Xcode	27
2.1 Důležité předvolby	27
2.1.1 „Building“	28
2.1.2 „Indentation“	28
2.2 Hlavní okno projektu	28
2.2.1 „Groups & Files“	29
2.2.2 Seznam v pravé horní části okna	30
2.2.3 Editory	32
2.3 Inspektory	32
2.3.1 Informace o projektu	33
2.3.2 Informace o souborech	33
2.3.3 Informace o cílech	35
2.4 Parametry pro sestavování cílů	36

3. Pojdme si napsat aplikaci...	39
3.1 Základní objektový design	40
3.1.1 Model	40
3.1.2 Vzhled	41
3.1.3 Řízení	42
3.2 Příprava projektu	42
3.2.1 Identifikátor a ikona aplikace	43
3.2.2 Drobná úprava kódu	45
3.3 Datový model	46
3.3.1 Princip datového modelování	46
3.3.2 Sestavení modelu v Xcode	47
3.4 Interface Builder	49
3.4.1 Co přesně je „xib“?	49
3.4.2 Základy práce s Interface Builderem	50
3.4.3 Základní úpravy „xibu“	52
3.4.4 Příprava GUI a řídicího objektu pro účty	53
3.5 Několik drobných vylepšení	57
3.5.1 Prázdňá textová pole	57
3.5.2 Součty transakcí	58
3.5.3 Lokalizace do dalších jazyků	59
4. Stručný průvodce jazykem C pro začátečníky	63
4.1 Staré dobré „Céčko“	64
4.2 Program, překlad, spustitelný program, aplikace...	64
4.3 Komentáře	65
4.4 Typy a proměnné	65
4.5 Konstanty a výrazy	68
4.6 Příkazy a bloky	71

4.7	Funkce a struktura programu	73
4.8	Standardní funkce	74
4.9	Globální a statické proměnné	74
4.10	Více zdrojových souborů	75
4.11	Preprocesor	76
5.	Objekty a Objective C	79
5.1	Neobjektová rozšíření Objective C	79
5.2	Objekty, atributy a zprávy	80
5.2.1	Zprávy	81
5.2.2	Syntaxe zpráv v Objective C	82
5.3	Třídy, instance a dědičnost	83
5.3.1	Třídy a instance	83
5.3.2	Dědičnost	84
5.3.3	Typy a konstanty Objective C pro práci s objekty	84
5.3.4	Textové objekty	85
5.3.5	Speciální příjemci	86
5.4	Definice nových tříd	87
5.5	Rozhraní	87
5.5.1	Instanční proměnné	88
5.5.2	Deklarace atributů	89
5.5.3	Deklarace zpráv	90
5.6	Implementace	91
5.6.1	Metody, „self“ a „super“	92
5.6.2	Přístupové metody	95
5.6.3	Přístup k atributům s využitím tečkové konvence	98
5.7	Metody tříd	98
5.8	Pozdní vazba	100

5.9 Objekty a neobjektové typy	101
5.10 Selektory	102
5.11 Kategorie a rozšíření	103
5.11.1 Rozhraní a implementace	103
5.11.2 Doplnění služeb knihovných tříd	104
5.11.3 Rozšíření	105
5.12 Protokoly	106
5.12.1 Deklarace protokolu	106
5.12.2 Třídy odpovídající protokolům	107
5.13 Předběžná deklarace třídy	108
5.14 Příkaz for/in	109
5.15 Přístup k instančním proměnným	110
5.16 Je to vše?	110
6. Základy práce s objekty	111
6.1 Vznik a zánik objektů	111
6.1.1 Zprávy „alloc“ a „init“	112
6.1.2 Kombinované zprávy pro třídy	112
6.1.3 Kopie; měnitelné a neměnné objekty	112
6.1.4 Další způsoby vytváření objektů	113
6.1.5 Správa paměti	113
6.1.6 Základy explicitní správy paměti	114
6.2 Základní služby objektů	115
6.2.1 Obecné služby	115
6.2.2 Hierarchie tříd	116
6.2.3 Schopnost reagovat na určené zprávy	116
6.2.4 Nepřímé provedení zprávy	117
6.3 Implementace vlastních tříd	118
6.3.1 Inicializace	118

6.3.2	Inicializace s parametry	119
6.3.3	Hlavní inicializátor	120
6.3.4	Pomocné metody tříd	121
6.3.5	Archivace	122
6.3.6	Ověření správnosti atributů	123
6.3.7	Odložená inicializace	124
6.3.8	Uvolnění zdrojů	125
6.4	Kdy nové třídy vytvářet?	127
6.4.1	Podtřídy	127
6.4.2	Delegace	128
6.4.3	Mechanismus akce/cíl	130
6.4.4	Kategorie	131
6.5	Spolupráce s Interface Builderem	132
6.5.1	Vytváření instancí v „xibu“	132
6.5.2	Vazby mezi objekty	134
6.5.3	Implementace akcí	135
7.	Standardní objektové knihovny	137
7.1	Zpracování chyb	138
7.1.1	Chybové záznamy a třída NSError	138
7.1.2	Výjimky...	141
7.1.3	...a třída NSError	142
7.1.4	Generování výjimek a „NSAssert“	143
7.2	Základní datové typy a kontejnery	143
7.2.1	Čísla	144
7.2.2	Porovnání hodnot objektů	144
7.2.3	Obecná binární data	144
7.2.4	Data a časové okamžiky	145
7.2.5	Základy práce s texty	147
7.2.6	Lokalizované texty	150
7.2.7	Pole a množiny objektů, a metody s proměnným počtem argumentů ...	151

7.2.8	Množina s počítáním výskytů	155
7.2.9	Indexování jménem: NSDictionary	156
7.2.10	Třídění pomocí deskriptorů	157
7.2.11	Výběr pomocí predikátů	158
7.2.12	Deskriptory, predikáty a GUI	158
7.2.13	NSNumber	160
7.3	Základy práce se soubory	161
7.3.1	Jména souborů a složek	161
7.3.2	Zobrazení jmen v uživatelském rozhraní	162
7.3.3	Služby třídy NSWorkspace	162
7.3.4	Archivace objektových sítí	163
7.3.5	Lokalizované soubory	164
7.4	Nepřímé předávání zpráv pomocí notifikací	165
7.4.1	Základní princip	165
7.4.2	Konkrétní třídy a služby	166
7.4.3	Standardní notifikace	167
7.4.4	Notifikace mezi aplikacemi	167
7.5	Grafický systém	167
7.5.1	Rámce	168
7.5.2	Geometrie	169
7.5.3	Základy práce s barvami	170
7.5.4	Základy kreslení	170
7.5.5	Bézierovy křivky	171
7.5.6	Kreslení obrázků	172
7.5.7	Rámce a buňky	172
7.5.8	Kreslení textů	173
7.6	Řízení aplikace	174
7.6.1	Třídy „NSResponder“ a „NSEvent“	174
7.6.2	Základy práce s myší	175
7.6.3	Základy práce s klávesnicí	176
7.6.4	Asynchronní panely	178

7.7 Dokumentová architektura	179
7.7.1 Specifikace typů podporovaných dokumentů	179
7.7.2 Obslužné třídy	179
7.7.3 Konkrétní příklad	180
7.7.4 Podpora služby „Undo“	184
7.7.5 Příkazy a „First Responder“	186
7.8 Textový systém	187
7.9 Aplikační předvolby	189
7.9.1 Základní principy	189
7.9.2 Přístup k předvolbám z programu	190
7.9.3 Podpora jiných než základních datových typů	191
7.9.4 Přístup k předvolbám z GUI	191
7.9.5 Konkrétní příklad	192
Závěr	195
Rejstřík	197

Java's not worth building in.

Nobody uses Java anymore.

It's this big heavyweight ball and chain.

Steve Jobs

Úvod

V této knize se naučíme psát aplikace pro počítače Apple. Využijeme k tomu standardního objektového vývojového prostředí Cocoa; naučíme se pracovat s jeho asi nejzajímavějším prostředkem, který nemá jinde obdoby – s Interface Builderem –, a samozřejmě se seznámíme s programovacím jazykem Objective C. Kniha obsahuje i samostatný úvod do dnes již klasického jazyka C v míře, potřebné pro programování v Cocoa.

Postupy, jež si zde ukážeme, poslouží i jako dobrý základ pro programování aplikací pro operační systém OS X (tedy pro zařízení iPhone a iPod Touch) – psaní těchto aplikací je založeno na týchž principech, jako programování v prostředí Cocoa v Mac OS X. Konkrétní detaily systému OS X, zvláště specifika jeho grafického uživatelského rozhraní, však v knize bohužel nelze uvést, neboť v době jejího psaní bylo celé vývojové prostředí ještě chráněno licencí firmy Apple před jakýmkoli zveřejňováním detailů.

Co očekáváme od čtenáře?

Jen naprosté minimum:

- schopnost pracovat s počítačem – není možné, abychom si v takovéto knize vysvětlovali, jak používat myš a k čemu slouží dok;
- základní představy o algoritmizaci obecně. Naproti tomu, znalost konkrétního programovacího jazyka není zapotřebí, tomu se budeme věnovat od základů;
- přístup k počítači Apple s operačním systémem Mac OS X, na němž si čtenář bude moci vše vyzkoušet.

Stručný přehled obsahu

Nejprve si ukážeme, jak se v Cocoa programuje, a naučíme se pracovat se základními vývojovými prostředky – s aplikacemi Xcode a Interface Builder. Ukážeme si také,

že díky síle Interface Builderu je možné sestavit netriviální aplikaci zcela bez znalosti programovacího jazyka Objective C, jen s využitím prostředků tzv. vizuálního programování.

Čtvrtou a pátou kapitolu pak věnujeme programovacímu jazyku: nejprve si stručně popíšeme jazyk C v rozsahu potřebném pro programování v Cocoa, a pak podrobně jeho nadstavbu Objective C.

Poslední – a nejrozsáhlejší – dvojice kapitol bude věnována praktickým pravidlům práce s objekty a službám nejčastěji využívaných knihovných tříd. Naleznete tam také řadu praktických příkladů a ukázek.

Něco málo o historii

Je možná vhodné si uvědomit, že ačkoli technologicky je Cocoa díky důslednému využití objektového systému s velkou rezervou nejmodernější z dnes běžně užívaných vývojových prostředí, novinkou v pravém smyslu slova není: vzniklo totiž jako vývojové prostředí operačního systému NeXTStep v polovině osmdesátých let minulého století – tedy před více než dvaceti lety. To je velkou výhodou prostředí Cocoa: díky tomu totiž jsou jeho dětské chyby již dávno vychytány. Navíc existuje po celém světě široká základna programátorů, kteří s programováním v tomto prostředí mají rozsáhlé a dlouholeté zkušenosti.

Operační systém NeXTStep se sice nikdy masově nerozšířil – příčinou zřejmě byla nepřilíh dobrá marketingová politika firmy NeXT – dobyl si však pevné místo na trhu jako daleko nejlepší prostředí pro tvorbu klíčových aplikací. Mnoho příznivců mu získalo špičkové grafické uživatelské rozhraní a geniální vývojové prostředí, jež tehdy neměly konkurenci (a vlastně ji nemají dodnes, vizte příští odstavec). Za zmínku snad stojí to, že právě zde vznikl World Wide Web, nebo revoluční a dodnes nedoceněný tabulkový procesor Lotus Improv.

V roce 1997 firma Apple, jež tou dobou byla dlouhodobě ve ztrátě, a jež měla za sebou dlouhou řadu neúspěšných pokusů něco rozumného provést se svým beznadějně zastaralým operačním systémem Mac OS, koupila NeXT „lock, stock, barrel and Steve“. Uběhlo deset let, a firma Apple dnes pod Jobsovým vedením prosperuje jako nikdy dříve; současný operační systém Mac OS X není ničím jiným, než novou a vylepšenou verzí NeXTStepu. Zvláště vývojové prostředí Cocoa získalo řadu podstatných rozšíření (např. objektové vazby, díky nimž si ve třetí kapitole ukážeme sestavení dosti výkonné aplikace, aniž bychom napsali jeden jediný řádek v programovacím jazyce).

1.

Malá ochutnávka na začátek: jak se v Kakau programuje?

Abychom hned zpočátku získali alespoň rámcovou představu, kterak se programuje v prostředí Cocoa, ukážeme si nejprve velmi jednoduchý příklad či dva. Také se seznámíme se základní aplikací integrovaného vývojového prostředí Apple, s Xcode.

Ilustrace a konkrétní ukázky postupů, s nimiž se v této knize setkáte, odpovídají verzi Mac OS X 10.5 Leopard a vývojovému prostředí Xcode 3. Máte-li ovšem na svém počítači některou z verzí starších, nebo snad je-li již mezitím k dispozici verze novější, o nic nejde: základní principy objektového vývojového prostředí Cocoa se nemění již řadu let, a rozdíly v konkrétních detailech nejsou nijak zásadní.

Z našeho pohledu asi nejpodstatnější novinky přineslo Objective C 2.0; pokud pracujete ve starším systému, kde je k dispozici pouze verze 1.0, vizte zvláště třetí kapitolu – hlavně odstavce věnované tzv. atributům – a také správu paměti, popsanou v kapitole čtvrté.

1.1 Instalace vývojového prostředí

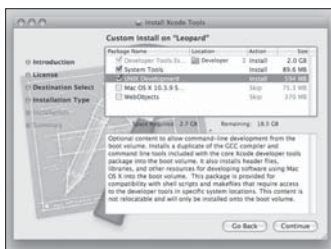
Vývojové prostředí Cocoa sice je standardně dodáváno s každým počítačem Apple, není však obvykle při dodávce nainstalováno na disk. Nejste-li si jisti, podívejte se začátek příštího bodu, který ukazuje, jak nalézt aplikaci Xcode; pokud ji máte, je vše v pořádku – ne-li, je zapotřebí nejprve celé vývojové prostředí nainstalovat.

To je velmi jednoduché: nejprve vložíme do optické mechaniky DVD s vývojovým prostředím, jež je standardní součástí dodávky počítače (spolu se záložními DVD, obsahujícími operační systém). Pokud je nemáme momentálně k dispozici, nic se neděje; připojíme se prostě na vývojářské webové stránky Apple „<http://developer.apple.com/technology/xcode.html>“. Zde – po případné registraci, jež je zdarma, a povinném přihlášení – nalezneme v sekci „Developer Tools“ odkaz „Xcode 3 / Free Download“. Získaný diskový obraz poklepáním připojíme k systému souborů, a pak již pokračujeme stejně, jako z instalačního DVD.



Obrázek 1.1: Instalační balík Xcode

Otevřeme poklepáním instalační balík „XcodeTools.mpkg“ (obr. 1.1), a dále stačí postupně přijímat standardní nabídky instalátoru. Snad jen je vhodné si zkontrolovat, abychom nezapomněli instalovat UNIXovou podporu – v této knize bychom se obešli i bez ní, ale pokročilejším programátorům se v praxi řada jejich služeb hodí. U nejnovější varianty vývojového prostředí se také můžeme rozhodnout, kam je instalovat; nemáme-li však nějaké specifické požadavky, standardní složka „Developer“ je optimální volbou.

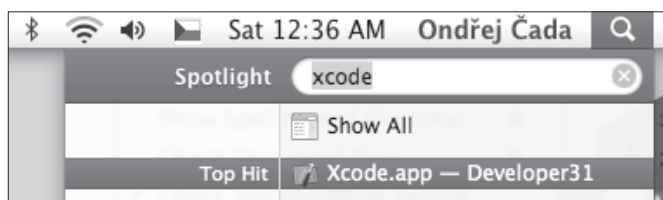


Obrázek 1.2: Instalace vývojového prostředí

Tak nainstalujeme na disk vlastní vývojové prostředí spolu s řadou velmi šikovných pomocných a obslužných programů; některé z nich si v této knize ukážeme blíže, ale většinu z nich prozatím ponecháme zkušenějším programátorům.

1.2 Pro začátek si vystačíme s Xcode

Programování aplikací v systému Mac OS X v podstatě vždy začíná tím, že spustíme integrované vývojářské prostředí *Xcode*, jež standardně najdeme ve složce „/Developer/Applications“. Nejnovější – v době psaní této knihy – verze vývojového prostředí však umožňuje instalaci víceméně kamkoli; asi nejjednodušší a nejspolehlivější proto je využít služby *Spotlight*, jež aplikaci zadaného jména nalezne kdekoli na disku. Klepneme tedy jen na ikonku lupy v pravém horním rohu obrazovky a zapíšeme požadované jméno; pak již stačí jen stisknout klávesu ENTER pro spuštění aplikace (obr. 1.3).



Obrázek 1.3: Spuštění Xcode prostřednictvím Spotlightu

Po prvním spuštění Xcode automaticky otevře okno, obsahující seznam novinek a zajímavých odkazů „Xcode News“ (ve starších verzích „Release Notes“); klidně je zavřeme – kdykoli bychom chtěli, máme je znovu k dispozici prostřednictvím nabídky *Help*. Po prvním spuštění také bývá vhodné určit některé z předvoleb; chcete-li, naleznete popis těch nejdůležitějších hned v příští kapitole, věnované právě prostředí Xcode. Prozatím však můžeme klidně vše ponechat ve výchozím standardním nastavení.

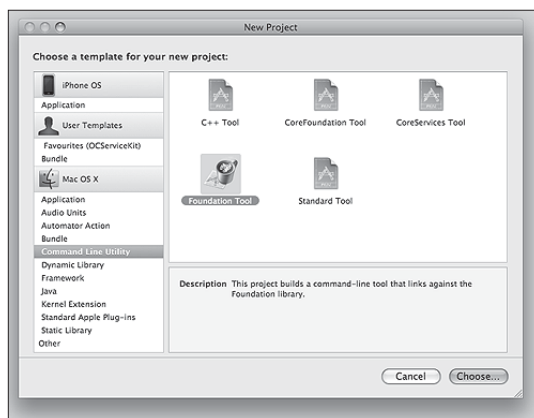
1.3 První – a nejjednodušší – program...

O autoru této knihy je obecně známo, že je tradicionalista a konzervatívec; nelze tedy dost dobře začít jinak, nežli tím, čím tradičně začínaly učebnice programování snad již v dobách, kdy se v nich popisovalo přepojování drátků v Eniacu: programem *Hello World*. Ten nedělá nic jiného, nežli to, že... vypíše text „Hello, World!“. Jak uvidíme, v Cocoa to doslova a do písmene nemůže být snazší – kromě vytvoření a spuštění projektu totiž není zapotřebí udělat zhola nic; ukážeme si ale na tomto triviálním příkladu ty nejzákladnější základy práce s aplikací Xcode.

Nejprve použijeme v aplikaci Xcode příkaz *File > New Project* pro vytvoření našeho prvního pokusného *projektu*: tak se v programátorském žargonu říká skupině všech souborů, jež dohromady umožní vytvořit požadovaný program. V Xcode se téměř vždy jedná o samostatnou složku, která obsahuje vše potřebné (přičemž naprostou většinu toho Xcode vytvoří automaticky, aniž bychom se o to museli explicitně starat).

Aplikace po volbě příkazu *New Project* zobrazí panel, v němž si můžeme vybrat požadovaný typ projektu; existuje předlouhá řada standardních vzorů, podle kterých Xcode dokáže různé typy projektů sestavovat, a zkušení programátoři mohou dokon-

ce vytvářet vzory vlastní. My prozatím zvolíme vzor „Foundation Tool“ ze skupiny „Command Line Utility“ v bloku „Mac OS X“, jak ukazuje obr. 1.4.



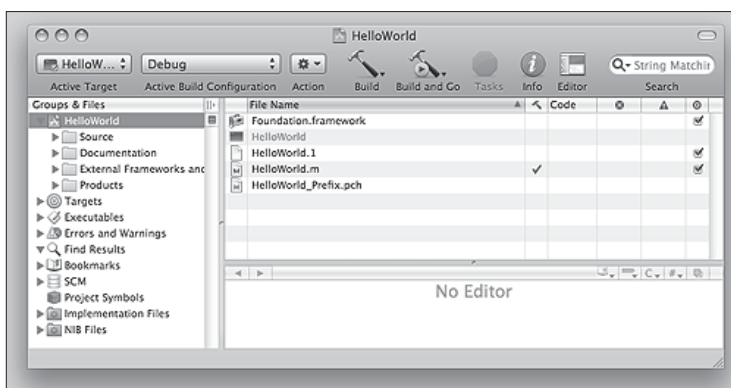
1.4. volba vzoru projektu



Ve starších verzích Xcode měl tento panel poněkud odlišný vzhled a byl méně přehledný; volba „Foundation Tool“ ze skupiny „Command Line Utility“ však platí i tam.

V následujícím kroku jen pomocí standardního (ve starších verzích Xcode mírně ne-standardního) panelu pro uložení souboru zvolíme jméno nového projektu a složku, do níž jej chceme umístit. Ani na jedno ani na druhé vývojové prostředí neklade zhora žádná omezení, zvolíme tedy cokoli, co nám nejlépe vyhovuje: třeba složku „~/Projekty“ a jméno „HelloWorld“.

Xcode pak vytvoří složku obsahující projekt, vygeneruje základní soubory, jež jsou zapotřebí, uloží je na disk a zobrazí okno, jehož prostřednictvím lze s projektem dále pracovat – vidíme je na obrázku 1.5.



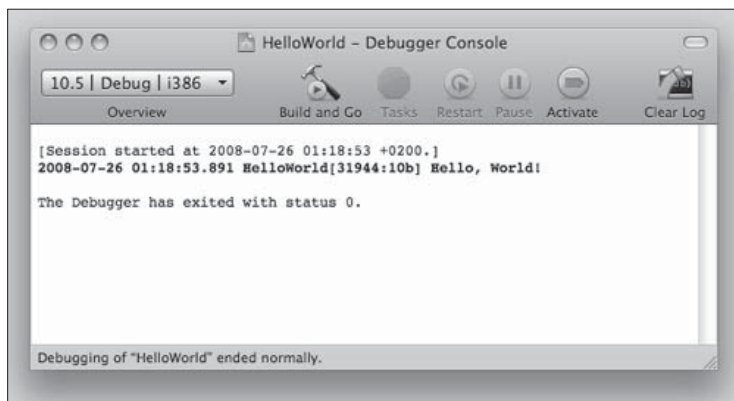
Obrázek 1.5: Okno projektu v Xcode

Co přesně v okně projektu vidíme a k čemu to slouží si řekneme později; nyní ale především sestavíme a spustíme náš první program.

1. Malá ochutnávka na začátek...

Nic snazšího: vůbec se nemusíme obtěžovat nějakým psaním zdrojových textů či úpravou vygenerovaných souborů. Jen klepneme myší na ovladač „Build and Go“ zhruba uprostřed lišty v horní části okna, a Xcode se pustí do práce: vytvoří potřebné pomocné soubory, přeloží program a spustí jej. Výsledek jeho běhu zobrazí v okně „Debugger Console“; neotevře-li se toto okno automaticky, zobrazíme si je pomocí příkazu *Run / Console* z hlavní nabídky aplikace Xcode.

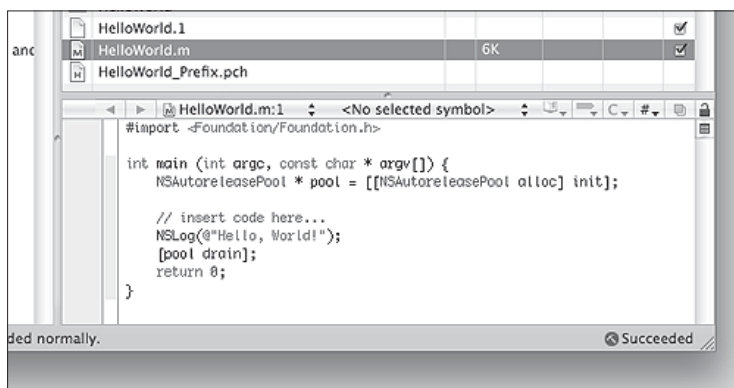
A voilà: okno obsahuje přesně požadovaný výpis – tedy „Hello, World!“. Pokud vše zkoušíte v Xcode, vidíte jej na obrazovce; jinak se můžete podívat na obr. 1.6.



Obrázek 1.6: Výsledek – výpis „Hello, World!“

To bylo až příliš jednoduché, že?

Inu, pravda je, že vytvoření tohoto programu byl tak trochu švindl – využili jsme totiž toho, že programátoři Apple jsou neméně konzervativní než autor této knihy; výpis textu „Hello, World!“ je proto součástí standardních vzorů, podle nichž se sestavují nově vytvářené projekty. Podívejme se na odpovídající kód – stačí v pravé horní části okna projektu klepnout myší na soubor „HelloWorld.m“, a jeho obsah se nám buď automaticky, nebo po klepnutí na ovladač „Editor“ v liště zobrazí v pravé dolní části, kde byl dosud jen implicitní text „No Editor“ (obr. 1.7).



Obrázek 1.7: Obsah standardně vygenerovaného souboru „HelloWorld.m“

Povšimněme si řádku „NSLog(@"Hello, World!")“ – právě ten zajistil požadovaný výpis.

Ovšemže nás takovýto podraz a hanebné využití práce cizích programátorů neuspokojí; chceme přece napsat nějaký vlastní program! Co kdybychom třeba změnili text „Hello, World!“ na „Nazdar, lidi“? Proč ne, to by fungovalo bez nejmenších problémů...



Pozor na nabodenička: kupříkladu varianta „Čau, světe!“ by *nefungovala*. Z historických důvodů totiž nelze ve zdrojových textech dost dobře používat znaky s diakritickými znaménky (resp. je třeba se držet znaků z dolní poloviny tabulky ASCII). Později si ukážeme, že to vůbec nijak nevádí, protože vývojové prostředí Cocoa nabízí daleko pohodlnější a praktičtější způsoby, kterak zobrazit nějaký text, než psát jej do zdrojového kódu; prozatím se prostě budeme držet angličtiny.

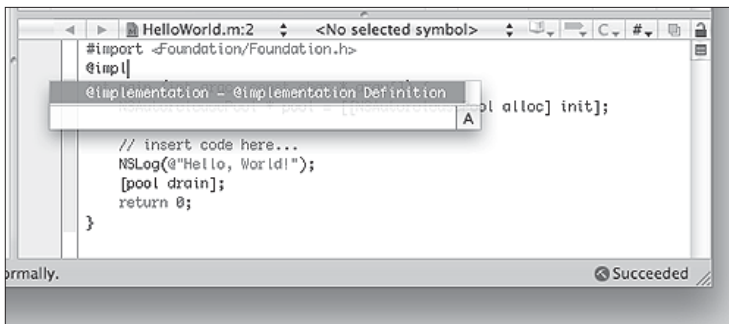
V příkladech vytištěných v této knize ovšem budeme pro lepší čitelnost používat české komentáře; ve zdrojových souborech v Xcode si to *v komentářích* můžeme dovolit bez obav, jen si musíme dát pozor, aby bylo použito kódování UTF-8. Přesto však v praxi je daleko lepší psát komentáře anglicky: to bude bez obtíží fungovat v libovolném vývojovém prostředí a v jakémkoli překladači, a navíc to umožní, aby na projektu mohli případně časem spolupracovat programátoři z libovolné země.

Ale ne, pouhá změna textu je pod naši úroveň: v prostředí Cocoa je možné i nemožné a obtížné věci jsou v něm snadné; měli bychom se tedy postarat o to, aby hned náš první program dělal něco rozumného. Jen tak se pozdravit, to nic není, to dokáže každý; co takhle hned ale také říci, co je ve světě nového?

1.4 Čtečka RSS v tučtu programových řádků

Pojďme náš program trochu vylepšit – jen docela málo, nebudeme muset napsat více, než asi tučet nových řádků.

Nejprve klepnutím umístíme kurzor hned na první volný řádek – tedy pod text „#import <Foundation/Foundation.h>“, napíšeme „@impl“, a stiskneme klávesu F5; editor Xcode rozezná, co jsme napsali, a nabídne automaticky doplnění, jak ukazuje obr. 1.8.



Obrázek 1.8: Automatické doplňování

Stisknutím mezerníku si vyžádáme vložení nabídnutého textu „@implementation ... @end“; připišeme za jeho prvou část ještě „HW:NSObject“, a stisknutím kombinace CTRL-/ zvýrazníme značku „methods“, již nám editor automaticky vložil mezi „@implementation“ a „@end“.

1. Malá ochutnávka na začátek...

Nyní opíšeme následující desetířádkový text:

```
static BOOL print=NO;
+(void)parser:p didStartElement:element namespaceURI:u
qualifiedName:q attributes:a {
    print=[element isEqual:@"description"];
}
+(void)parser:p didEndElement:e namespaceURI:u qualifiedName:q {
    if (print) printf("\n");
}
+(void)parser:p foundCharacters:string {
    if (print) printf("%s",[string UTF8String]);
}
```

Konečně pak níže nahradíme řádek „NSLog(@"Hello, World!")“ následující trojicí řádků:

```
NSXMLParser *xml=[[NSXMLParser alloc]
initWithContentsOfURL:[NSURL URLWithString:
    @"http://www.bbc.co.uk/syndication/feeds/news/ukfs_news/
world/rss091.xml"]];
[xml setDelegate:[HW class]];
[xml parse];
```

A to je celé; pro kontrolu se můžete podívat na obrázek 1.9, který zobrazuje kompletní zdrojový text po našich úpravách. Znovu klepneme myší na ovladač „Build and Go“ při horním okraji okna, a pokud jsme při opisování někde neudělali chybu, program se přeloží a spustí – a máme-li funkční připojení k Internetu, do konzole se vypíše seznam světových novinek, načtených z BBC. Výpis bude vypadat velmi přibližně takto – samozřejmě, konkrétní novinky budou odlišné:

[Session started at 2008-07-26 02:15:36 +0200.]

Visit BBC News for up-to-the-minute news, breaking news, video, audio and feature stories. BBC News provides trusted World and UK news as well as local and regional perspectives. Also Entertainment, business, science, technology and health news.

A deadline expires for Bosnian Serb ex-leader Radovan Karadzic to appeal against extradition to The Hague on genocide charges.

Global trade talks that earlier looked near collapse have made progress and a deal might now be possible.

Four Hamas militants and a five-year-old girl die in an explosion targeting a car on Gaza City's beachfront, medics say.

Chadian rebels free an American missionary after holding him hostage for more than nine months, his organisation says.

A Qantas plane makes an emergency landing in Manila as a hole causes it to lose cabin pressure, an airport official says.

President Bush signs an order expanding US sanctions on Robert Mugabe's „illegitimate“ government in Zimbabwe.

...