Plně kompatibilní s Android 5

Ľuboslav Lacko

Vývoj aplikací pro Android

Rychlý přechod na vývoj pro Android

Od základů po pokročilé techniky

Tipy pro Android Wear a Google Glass

computer press

Ľuboslav Lacko

Vývoj aplikací pro Android

Computer Press
Brno
2015

Vývoj aplikací pro Android

Ľuboslav Lacko

Překlad: Martin Herodek
Obálka: Martin Sodomka
Odpovědný redaktor: Martin Herodek
Technický redaktor: Jiří Matoušek

Objednávky knih:
http://knihy.cpress.cz
www.albatrosmedia.cz
eshop@albatrosmedia.cz
bezplatná linka 800 555 513
ISBN 978-80-251-4347-6

Vydalo nakladatelství Computer Press v Brně roku 2015 ve společnosti Albatros Media a. s.
se sídlem Na Pankráci 30, Praha 4. Číslo publikace 19014.

1. vydání

ALBATROS MEDIA a.s.

Obsah

KAPITOLA 10

KAPITOLA 11

Úvod

Vzhledem k aktuálním trendům v IT a životnímu stylu hlavně mladých lidí, kteří si bez mobilních zařízení – chytrých telefonů a tabletů – už nedokáží představit svou existenci, získávají na významu mobilní aplikace. Většina uživatelů se zařadila do hlavního proudu, tedy mezi uživatele aplikací. Někteří však mají vyšší ambice a chtějí například vyřešit svoje individuální požadavky, případně mají nápad a chtějí ho realizovat formou mobilní aplikace. Tato publikace vám formou praktických postupů odhalí know-how vývoje aplikací pro mobilní platformu Android, ať už se jedná o chytré telefony nebo tablety.

Poznámka: Podle údajů od agentur IDC a Gartner ovládá v současnosti Android přibližně 80 % trhu s mobilními přístroji a je provozován na více než miliardě telefonů a tabletů. To platí ve všeobecnosti, nikoliv však v podnikové sféře.

V publikaci se pokusíme zbořit mýtus o složitosti vývoje mobilních aplikací a nároků na vybavení pro tuto činnost. První pokusy s vývojem mobilních aplikací nevyžadují žádné investice, jelikož díky emulátorům dokážete vyvíjet i bez toho, abyste měli příslušné zařízení fyzicky k dispozici.

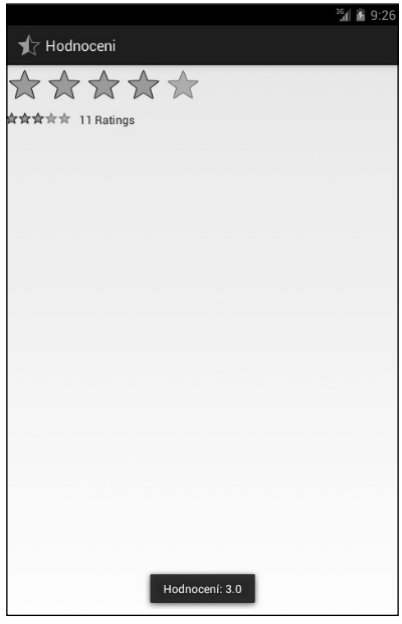
Publikace je určena i migrujícím vývojářům, kteří chtějí svou úspěšnou aplikaci, vytvořenou původně pro počítač nebo pro některou mobilní platformu, zpřístupnit i milionům uživatelů zařízení s Androidem.

Publikace předpokládá určité předběžné znalosti:

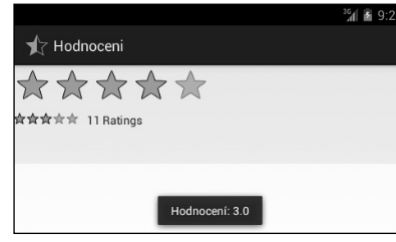
- Znalost programovacího jazyka Java. Postačí i znalost C#, jelikož tento jazyk je odvozený od Javy a je mu velmi blízký.
- Znalost základních principů objektově orientovaného programování.
- Zkušenosti s používáním moderních integrovaných vývojových prostředí.
- Znalost SQL (nejlépe SQLite, ale postačuje základní všeobecný přehled).
- Znalost základů XML, jelikož v tomto formátu se realizuje návrh uživatelského rozhraní.

Všechny příklady v této publikaci byly odladěny na telefonu Sony Xperia L s operačním systémem Android 4.2.2 a tabletu Lenovo Yoga Y10 s Androidem 4.4.

Poznámka: Některé screenshoty cvičných aplikací zobrazují relevantní obsah pouze v horní a dolní části. Většina plochy uprostřed snímku obrazovky je prázdná. Proto jsme pro úsporu místa v publikaci tyto obrázky upravili tak, že jsme prázdnou oblast odstranili.



Obrázek Ú.1: Původní obrázek



Obrázek Ú.2: Upravený obrázek

Zpětná vazba od čtenářů

Nakladatelství a vydavatelství Computer Press, které pro vás tuto knihu připravilo, stojí o zpětnou vazbu a bude na vaše podněty a dotazy reagovat. Můžete se obrátit na následující adresy:

Computer Press
Albatros Media a.s., pobočka Brno
IBC
Příkop 4
602 00 Brno

nebo

sefredaktor.pc@albatrosmedia.cz

Computer Press neposkytuje rady ani jakýkoli servis pro aplikace třetích stran. Pokud budete mít dotaz k programu, obraťte se prosím na jeho tvůrce.

Errata

Přestože jsme udělali maximum pro to, abychom zajistili přesnost a správnost obsahu, chybám se úplně vyhnout nelze. Pokud v některé z našich knih najdete chybu, ať už chybu v textu nebo

v kódu, budeme rádi, pokud nám ji oznámíte. Ostatní uživatele tak můžete ušetřit frustrace a pomoci nám zlepšit následující vydání této knihy.

Veškerá existující errata zobrazíte na adrese *http://knihy.cpress.cz/K2170* po klepnutí na odkaz Soubory ke stažení.

KAPITOLA 1

Nástroje pro vývoj

V této kapitole:
- Co budete potřebovat
- Vytvoření emulátoru
- Ověření konfigurace na cvičném projektu
- Android Studio
- Embarcadero RAD Studio XE6
- Xamarin MonoTouch a Mono for Android
- Game Maker Studio na vývoj her

Co budete potřebovat

K vývoji aplikací budete potřebovat čtyři základní nástroje a komponenty:

- Java Development Kit (JDK)
- Vývojové prostředí Eclipse
- Android Development Tools (ADT)
- Android Software Development Kit (SDK)

Alternativu představuje Android Studio, které však je v současnosti k dispozici jen ve verzi Early Access Preview. Aplikace pro Android je možné vyvíjet na platformě Windows, Linux i Mac. Vstupním bodem k získání vývojářských nástrojů, návodů a příkladů je stránka *http://developer.android.com*.

Instalace programovacího jazyka Java

Jazyk Java byl vytvořen vývojářským týmem firmy Sun Mircosystems pod vedením Jamese Goslinga. Původně měl být určen pro spotřební elektroniku. Pro zajímavost – původní název projektu byl Oak (dub). Po zjištění, že už existuje programovací jazyk s tímto jménem, byl přejmenován na Javu. Vývoj první verze byl ukončen v roce 1995.

Java je objektově orientovaný programovací jazyk, jehož syntaxe je podobná C nebo C++. Java je interpretovaný jazyk – to znamená, že program se nepřekládá přímo do strojového kódu, ale do takzvaného Java bajtkódu, který je následně interpretován virtuálním strojem Java. To zabezpečuje, že Java je na platformě nezávislý jazyk. O správu paměti se stará automatický

„garbage collector", jenž zabezpečuje, aby objekty, které se už nepoužívají, byly odstraněny. Problém však představuje skutečnost, že dopředu nevíme, kdy bude spuštěn.

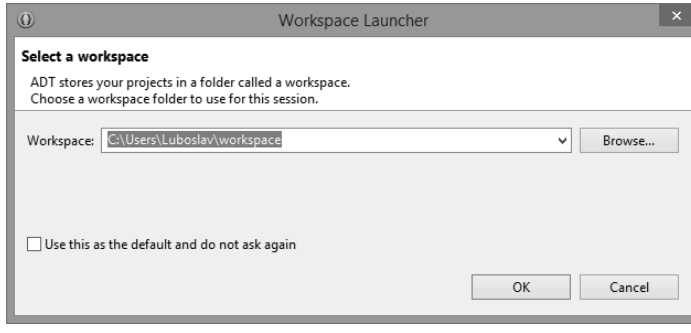
Existují čtyři edice jazyka Java zaměřené na různá prostředí s různě rozsáhlým obsahem.

- **JavaCard** určená hlavně pro čipové karty
- **Java ME (Micro Edition)** zaměřená na mobilní telefony
- **Java SE (Standard Edition)** čili standardní verze pro klasické počítače
- **Java EE (Enterprise Edition)** určená pro rozsáhlé podnikové informační systémy

Poznámka: Pod taktovkou Javy běží více než 3 miliardy zařízení, aplikace celosvětově vytváří 9 milionů vývojářů.

Tip: Programovací jazyk Java, konkrétně Java Development Kit, stáhnete pro platformy Windows a Linux z webu Oracle. Pro platformu Mac OS stáhnete Javu ze stránek společnosti Apple. Platforma Java byla produktem společnosti Sun, tuto společnost však se všemi produkty a řešeními získala společnost Oracle, která Javu momentálně spravuje. Z jejich stránek je možné stáhnout a nainstalovat Java SDK.

Pokud se nejste jistí, zda máte JDK nainstalované, nainstalujte balík Android SDK. Ve složce *eclipse* spusťte aplikaci *eclipse*. Pokud JDK máte, spustí se vývojové prostředí normálně. Pokud tento doplněk nainstalovaný nemáte, aplikace se nespustí a vypíše chybové hlášení. Java Development Kit najdete snadno pomocí vyhledávače, do kterého zadáte frázi „Java JDK". V době psaní publikace byla k dispozici verze Java 8. Android SDK však v aktuální verzi dokázal pracovat jen se starší verzí JDK 7. Podle verze operačního systému, který máte na vývojářském počítači, si nainstalujte 32- nebo 64bitovou verzi Java Development Kitu.



Obrázek 1.1: Upozornění na nutnost instalace Java Development Kitu

JDK se na platformě Windows instaluje implicitně do složky *C:\Program Files\Java*.

Java 8

Přes konstatování, že Android SDK v aktuální verzi dokázal v době psaní publikace pracovat jen se starší verzí JDK 7, je jen otázkou času, kdy bude podporovat i verzi Java 8. Proto si v hrubých rysech představíme nejvýznamnější novinky této verze.

Podpora pro platformu Java SE 8 je tak jako u všech předchozích verzí automaticky dostupná ve vývojovém prostředí NetBeans v den uvedení na trh. Ostatní vývojová prostředí si také uvědomila důležitost a výjimečnost této verze Javy a začlenila její podporu také od prvních dní (například Eclipse IDE podporuje Java 8 od verze 4.3 Kepler).

Vývojáři budou schopní psát v nové Javě kód, který je kompaktnější a snáze udržovatelný. Klíčovými novinkami v JDK 8 významně redukujícími objem kódu jsou lambda výrazy nad velkými objemy dat, nový interpret jazyka JavaScript známý pod kódovým označením Nashorn a eliminované permanentní generování kódu z virtuálního stroje. Nashorn běží jako součást Java Virtual Machine a umožňuje javovým aplikacím využívat komponenty napsané v JavaScriptu, případně spouštět v JVM celé javascriptové aplikace. K snazší lokalizaci aplikací přispějí i nové možnosti práce s datem a časem.

Java 8 by měla podstatně urychlit vývojové cykly všech typů aplikací včetně mobilních a aplikací pro různé spotřebiče a výrobky v rámci „Internetu věcí". K dosažení avizovaných cílů přispívá i spolupráce ARM a Oraclu na definici a integraci technologií. Jedním z výsledků této spolupráce je i platforma Oracle JDK 8. V blízké budoucnosti se očekává expanze různých „wearable" zařízení, tedy inteligentních náramků, brýlí a podobně. Pro tato zařízení umožňuje Java 8 zjednodušení komunikace, lehkou škálovatelnost a zvýšenou robustnost vyvíjených aplikací. K efektivitě vývoje přispěje i vývojové prostředí pro vestavná (embedded) zařízení.

Vývojové prostředí Eclipse

Eclipse je open- source vývojové prostředí (IDE – Integrated Development Environment) určené primárně k programování v jazyce Java. Jeho flexibilita vám však umožňuje nainstalovat doplňky pro další programovací jazyky, například PHP, Python, C++, Ruby a další. Jelikož je Eclipse samotné napsáno v Jazyce Java, potřebujete mít nainstalované prostředí JRE (Java Runtime Environment).

Instalace a konfigurace Android SDK a doplňků ADT

Vývojářský balík Android SDK, pro který se často používá i zkratka ADK, je k dispozici zdarma na *http://developer.android.com/sdk*. Stáhněte soubor s názvem *adt-bundle-<os_platforma-datum>.zip*. V našem případě měl soubor pro platformu Windows název *adt-bundle-windows-x86_64-20140321.zip*, kde poslední posloupnost čísel udává datum vydání edice. Balík obsahuje tyto součásti:

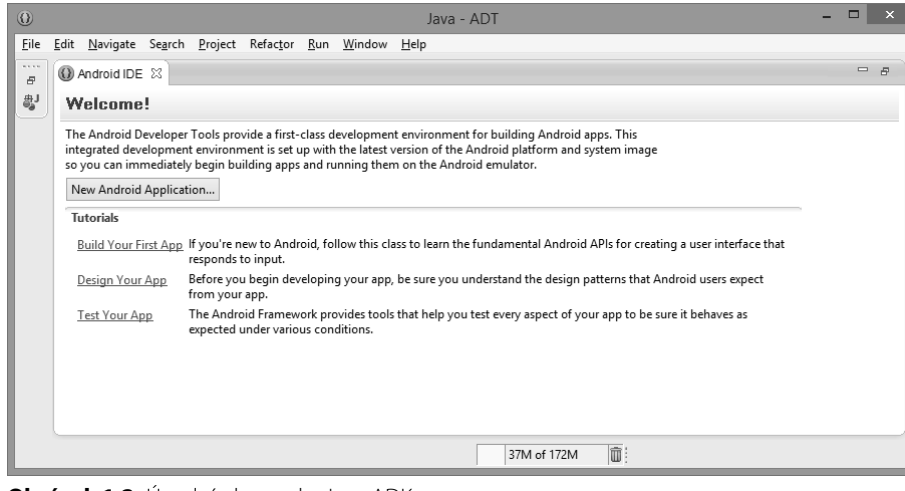
- Vývojové prostředí Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- Obrazy verzí operačního systému pro emulátor

Poznámka: Android SDK není potřeba instalovat, abyste ale mohli začít vyvíjet aplikace, musíte ho rozbalit a hlavně nakonfigurovat. Konfigurace je jednodušší, než se na první pohled zdá, je však zapotřebí dodržet postup.

Námi popisovaný postup je pro operační systém Windows 8, ale úplně stejně můžete postupovat i ve starší verzi Windows 7.

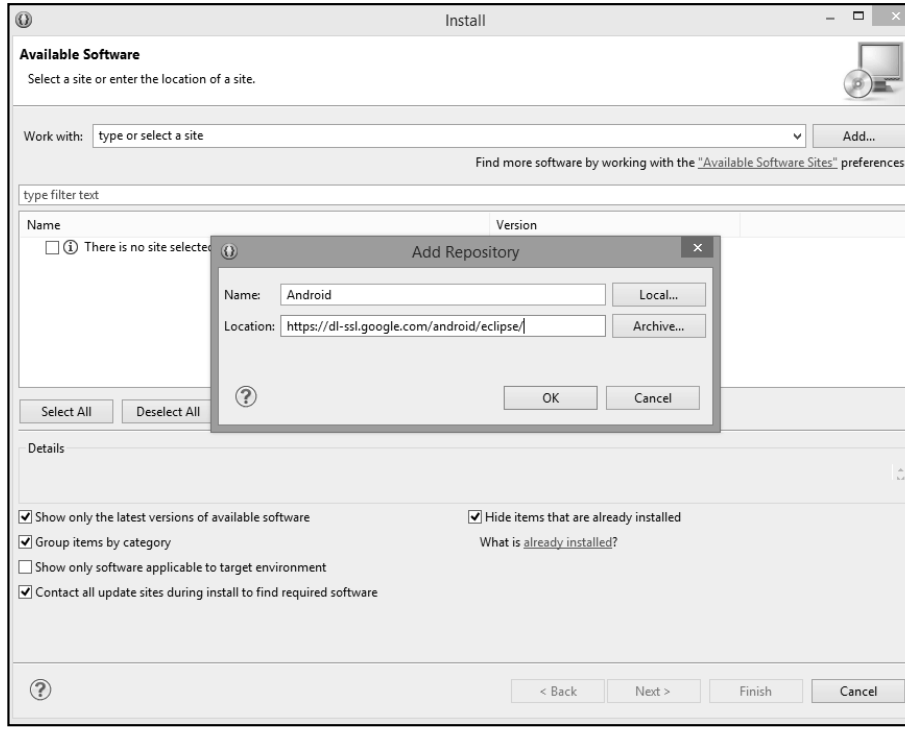
1. V archivu staženém z webu je složka se stejným názvem jako název souboru archivu. Tuto složku rozbalte na vhodné místo. Doporučujeme vytvořit složku s výstižným názvem, například *Android*, *Vývoj* apod. Ve složce *adt-bundle-<os_platforma-datum>* jsou vnořené složky *eclipse*, *sdk* a spustitelná aplikace *SDK Manager*. Vývojové prostředí vyžaduje, aby bylo nainstalované prostředí Java Runtime Enviroment (JRE) nebo Java Development Kit (JDK).
2. Ve složce *eclipse* spusťte aplikaci *eclipse.exe*. Zobrazí se uvítací obrazovka s aktuálními informacemi, návody, příklady a podobně. V této fázi úvodní obrazovku zavřete.

Tip: Eclipse můžete spustit z jakékoliv složky, do níž jste jej rozbalili – nevyžaduje žádnou instalaci. Jelikož je Eclipse vstupním bodem pro vývoj aplikací, doporučujeme vytvořit pro něj na ploše operačního systému zástupce.



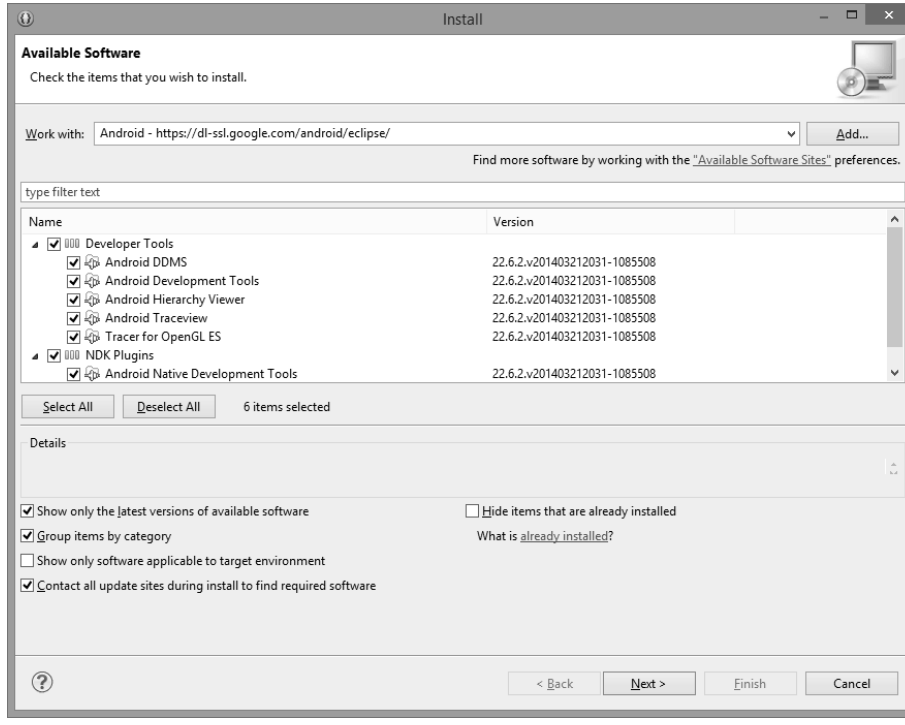
Obrázek 1.2: Úvodní obrazovka Java ADK

3. Aktivujte v nabídce položku **Help → Install New Software**. V instalačním dialogu klepněte na tlačítko **Add** a přidejte nový zdroj modulů *https://dl-ssl.google.com/android/eclipse/*. Zdroj modulů nějak nazvěte, například **Android**. Přidají se moduly Android Developers Tools (ADT).

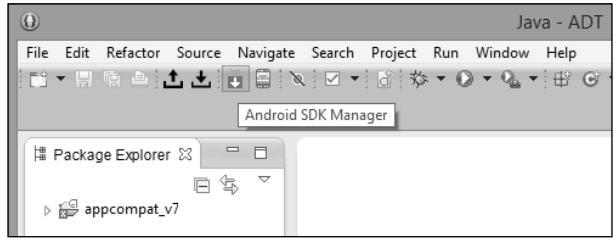


Obrázek 1.3: Instalace zásuvných modulů Android Developers Tools (ADT)

4. V seznamu modulů přibydou dvě položky: **Developer Tools** a případně i **NDK Plugins**. Obě položky zaškrtněte a klepněte na tlačítko **Next**. Dokončete instalaci modulů. Průběh jejich stahování a instalace můžete sledovat v dialogu.
5. Po ukončení instalace modulů je potřeba ukončit aplikaci Eclipse a znovu ji spustit. V námi instalované verzi stačilo souhlasit s nabídkou na restartování a vývojové prostředí se automaticky ukončilo a znovu spustilo.
6. Po nainstalování Android Developers Tools (ADT) je potřeba do vývojového prostředí Eclipse nakonfigurovat propojení na Android SDK. Tím se propojí moduly ADT nainstalované v předchozích krocích s SDK. Spusťte Android SDK Manager. Můžete ho spustit přímo z Eclipse pomocí ikony panelu nástrojů (ikona se symbolem zelené šipky směřující dolů).



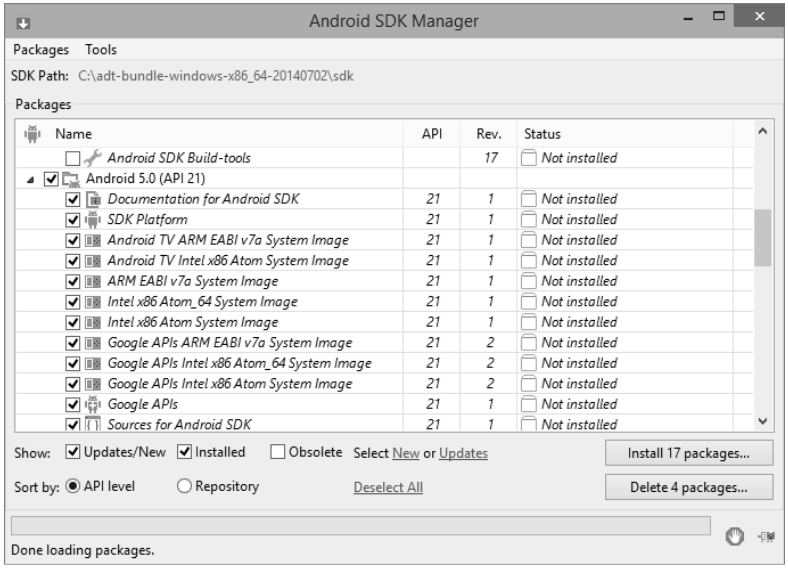
Obrázek 1.4: Konfigurace zásuvných modulů Android Developers Tools (ADT)



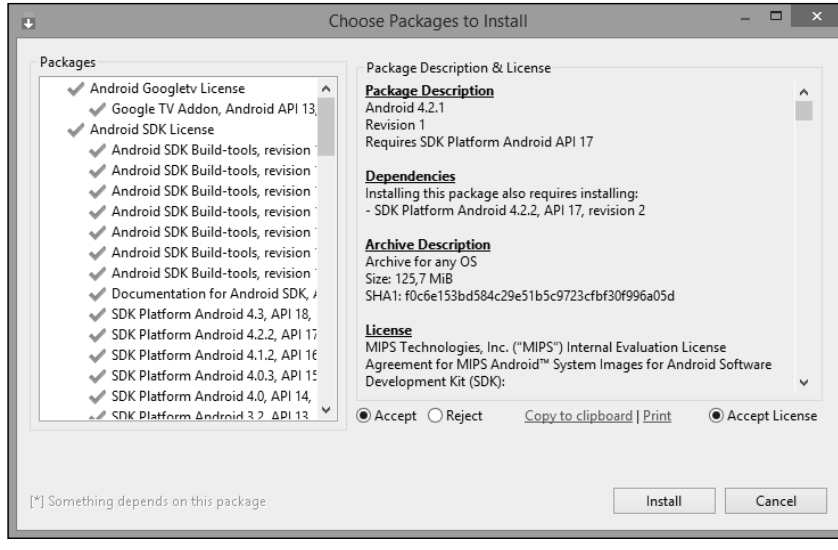
Obrázek 1.5: Spuštění nástroje Android SDK Manager přímo z prostředí Eclipse

7. V aplikaci Android SDK Manager označte položky, které chcete instalovat. Označte složky **Tools**, **Extras** a všechny verze Androidu, pro které chcete vyvíjet a testovat aplikace. Pokud chcete pokrýt i starší zařízení, nainstalujte vývojářskou podporu počínaje verzí 2.2. Tak pokryjete nejširší spektrum momentálně dostupných zařízení. Chytré telefony nemají dlouhou životnost, přístrojů s kdysi velmi rozšířenou verzí 2.2 je stále méně a méně. V současnosti převládají telefony a tablety s verzí 4.0 a vyšší. Verze 3.0 HoneyComb byla určena jen pro tablety a přístrojů s touto verzí v našich končinách není mnoho. Doporučujeme vydat se spíše cestou inovací. V době psaní publikace byla k dispozici verze ADT, která už podporovala novou platformu Android 5.0 Lollipop a Android 4.4W, kde W znamená „wearable“, čili variantu Androidu pro zařízení, která můžeme nosit na so-

bě. Typickým příkladem jsou hodinky od různých výrobců či populární brýle Google Glass, které však zatím nejsou v České republice ani na Slovensku oficiálně dostupné. Povzbuzujeme vás, abyste si SDK pro Android 5.0 Lollipop a Android 4.4 Wearable nainstalovali, vytvořili si pro tyto platformy emulátory a jako vývojáři se s nimi v předstihu seznámili.



Obrázek 1.6: Instalace balíčků v aplikaci Android SDK Manager



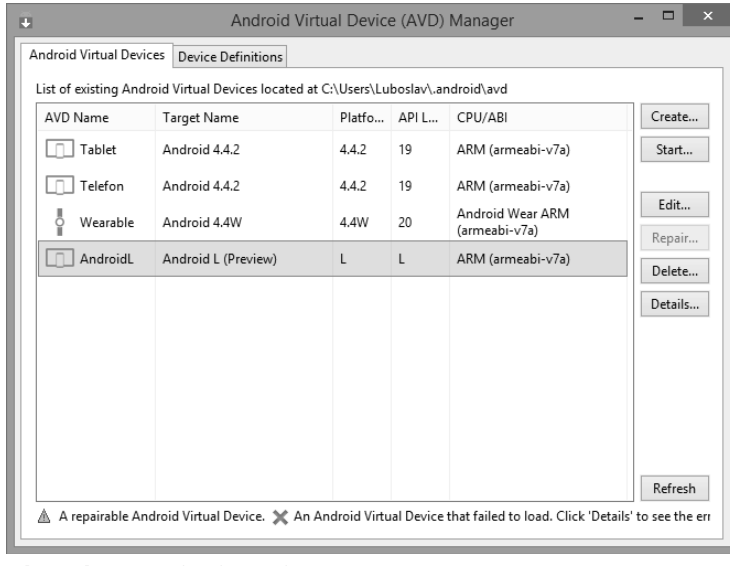
Obrázek 1.7: Souhlas s licenčními podmínkami při instalaci balíčků v aplikaci Android SDK Manager

8. V následujícím dialogu musíte postupně akceptovat licence všech balíčků, které hodláte instalovat. Prohlédněte si celý seznam, aby všude byly jen zelené ikonky bez červených křížků.

Vytvoření emulátoru

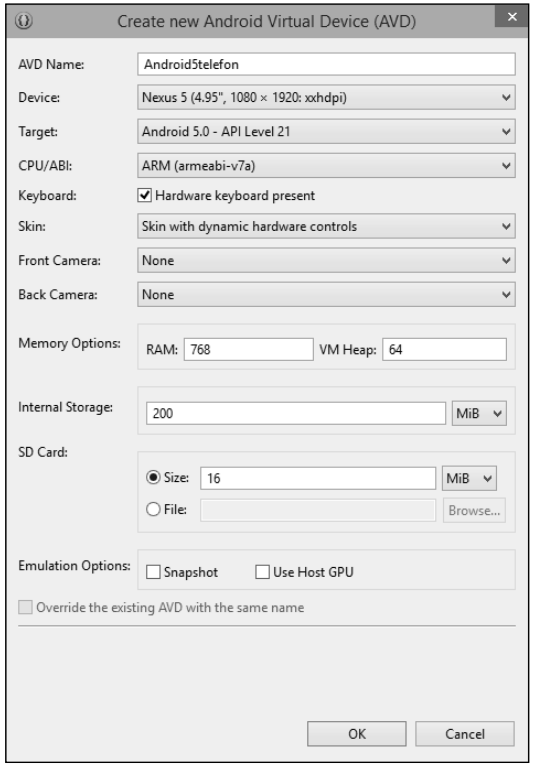
Zdálo by se, že pokud máte k dispozici jeden moderní chytrý telefon a jeden tablet s Androidem, dokážete pokrýt návrh, testování a ladění aplikací. Opak je pravdou. Výhodou a zároveň nevýhodou Androidu je variabilita různých zařízení s různými verzemi systému a různým rozlišením displeje. Neodmyslitelnou pomůckou vývojáře je proto emulátor, na kterém je možné otestovat aplikaci ve více verzích operačního systému Android a na obrazovkách s různým rozlišením.

V aplikaci Android SDK Manager v nabídce **Tools** zvolte položku **Manage AVDs**. Zkratka AVD znamená Android Virtual Devices. Zobrazí se okno aplikace **Android Virtual Device Manager**. Přepněte se na **Android Virtual Devices** a nainstalujte emulátor pro příslušnou verzi Androidu. Vyberte si verzi, kterou disponuje vaše nebo zamýšlené mobilní zařízení, pro nějž je aplikace určena.

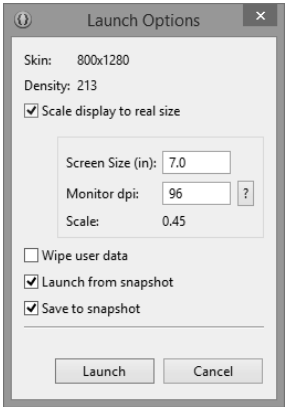


Obrázek 1.8: Android Virtual Device Manager

Pro účely této publikace doporučujeme vytvořit minimálně dva emulátory, oba pro verzi Android 4.2.2 (API Level 17) nebo 4.4: jeden emulátor telefonu s úhlopříčkou 4 palce a rozlišením 480 × 800, druhý emulátor tabletu s úhlopříčkou 7 až 10 palců. Pokud označíte možnost **Snapshot**, druhé a další spuštění emulátoru proběhne velmi rychle, protože AVD po zavření ukládá svůj aktuální stav.



Obrázek 1.9: Vytvoření emulátoru pro platformu Android 4.2.2



Obrázek 1.10: Dialog nastavení škálování zařízení

Abyste se seznámili s množnostmi nové verze Android 5.0 Lollipop, doporučujeme vytvořit i emulátor této platformy, jelikož nejnovější přístroje se budou postupně na tuto verzi upgradovat.

Při vytváření emulátoru nezapomeňte nakonfigurovat dostatečnou kapacitu paměti **SD Card**. Emulátor můžete spustit přímo z dialogu **Android Virtual Device Manager** tlačítkem **Start**. První spuštění emulátoru trvá trochu déle, u dalších spuštění je už doba náběhu přiměřená.

Při používání emulátoru může nastat problém s jeho zobrazením. Základní mód většiny telefonů a tabletů s Androidem je totiž „na výšku“, naproti tomu monitory vývojářských počítačů jsou orientované „na šířku“. Jak zobrazit tablet s rozlišením 800 × 1 200, případně vyšším, na monitoru s vertikálním rozlišením 768 pixelů? Po spuštění emulátoru tlačítkem **Start** se zobrazí dialogové okno umožňující nastavení měřítka zobrazení. Doporučujeme označit volbu **Scale display to real size**.

Poznámka: Kvůli kompatibilitě vyberte nejnižší předpokládanou verzi systému. Takovéto aplikace budou na vyšších verzích fungovat bez problémů, opačně to ale neplatí.

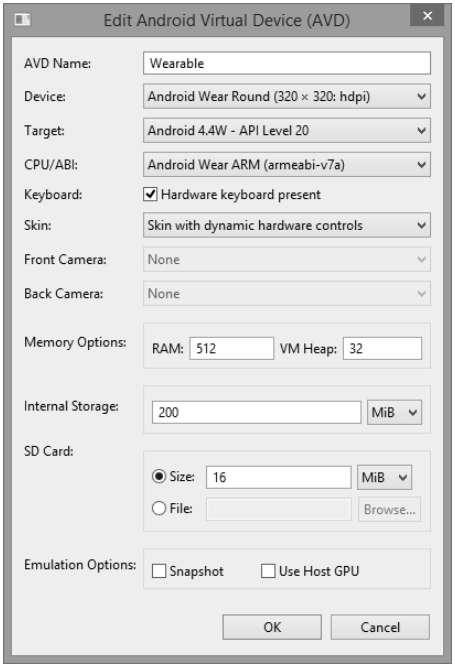
Obrázek 1.11: Spuštění emulátoru pro platformu Android 4.2.2

Pro hloubavější čtenáře je diskový obraz emulátoru v operačním systému Windows 8 uložen v adresáři *C:\Users\<uživatel>\.android*\. V souboru *config.ini* jsou základní parametry emulátoru.

Vytvoření emulátoru platformy Android 4.4 Wearable

Pro platformu Android 4.4 Wearable jsou k dispozici šablony **Android Wear Round** (320 × 320; hdpi) a **Android Wear Square** (280 × 280; hdpi).

Při prvním spuštění emulátoru probíhá inicializace operačního systému. Na emulátoru to trvá několik minut a během této doby doporučujeme, aby byl váš počítač připojen k Internetu. Po spuštění emulátoru pokračujte v konfiguraci zařízení podle pokynů.



Obrázek 1.12: Vytvoření emulátoru platformy Android 4.4 Wearable

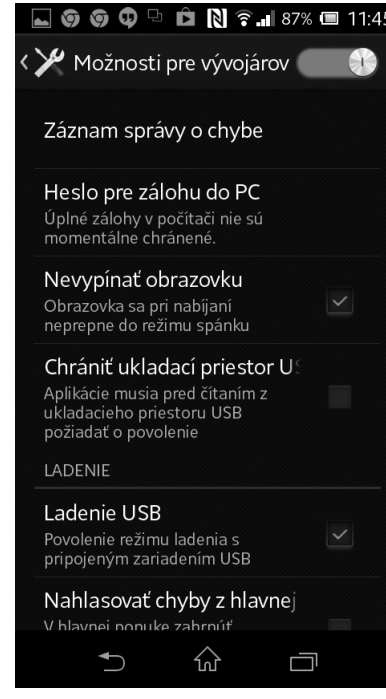
Obrázek 1.13: Emulátor Android 4.4 Wearable

Spouštění aplikací na reálném zařízení

Ke spuštění a ladění aplikací na reálném zařízení je potřeba přepnout jej do vývojářského módu. Klepněte v **Nastavení** na položku **Možnosti pro vývojáře**. Zaškrtněte položku **Ladění USB**. Doporučujeme zaškrtnout i položku **Nevypínat obrazovku**. Po označení této položky se v režimu, kdy je přístroj připojený přes USB kabel k počítači, nebude vypínat obrazovka.

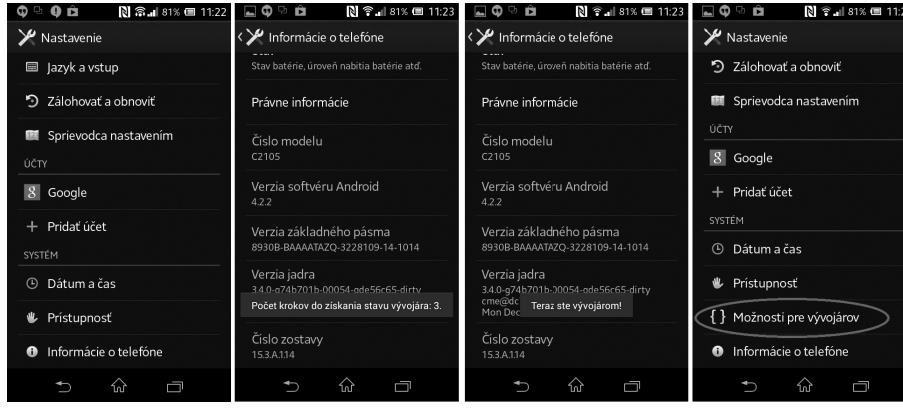
Na zařízeních s operačním systémem Android 4.2 a vyšším jsou možnosti pro vývojáře implicitně skryty a musíte je nejprve zobrazit:

1. V **Nastavení** klepněte na položku **Informace o telefonu**.
2. Následně najděte položku **Číslo sestavy**. Zpravidla je poslední v seznamu.



Obrázek 1.14: Povolení ladění přes USB

3. Opakovaně na tuto položku klepejte. Klepnutí musíte provést sedmkrát. Nejprve se zobrazí oznámení ve tvaru **Počet kroků do získání stavu vývojáře: 2**, které avizuje, kolikrát je potřeba ještě klepnout. Když se zobrazí oznámení **Nyní jste vývojářem**, přibude v **Nastavení** položka **Možnosti pro vývojáře**.



Obrázek 1.15: Postup odemknutí nabídky Možnosti pro vývojáře

Na straně počítače potřebujete ke komunikaci se zařízením s Androidem USB ovladače pro ADB (Android Debug Bridge). Buď použijete ovladač *google_usb_driver*, který je součástí Android SDK, nebo ovladač dodaný výrobcem zařízení.

Ověření konfigurace na cvičném projektu

Možná se ptáte, proč se pouštět do vytváření projektu mobilní aplikace dříve, než se seznámíte aspoň v hrubých rysech s architekturou operačního systému Android. Důvod je jednoduchý. Pokud se aplikace dá přeložit a spustit nejprve na emulátoru a následně na reálném zařízení, máte jistotu, že máte správně nainstalované a nakonfigurované vývojové prostředí, překladač jazyka Java, Android SDK, emulátor a propojení na reálné zařízení.

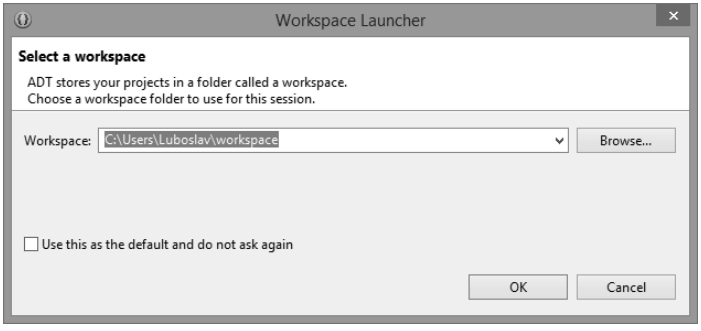
Poznámka: První projekt má v tomto případě i motivační význam, jelikož doslova na jedno klepnutí a bez jakéhokoliv programování vytvoříte aplikaci typu „Hello World", která vypíše na obrazovku text.

Začátečníci se v tomto příkladu nemusí snažit pochopit souvislosti. Návod na vytvoření cvičného projektu je uveden jako obrázkový postup krok za krokem.

Spusťte vývojové prostředí Eclipse. Zobrazí se dialogové okno s výběrem pracovního prostoru, tedy složky na disku, kam se budou ukládat projekty vývojového prostředí Eclipse a konfigurační soubory, které obsahují informace o rozmístění oken na pracovní ploše vývojového prostředí, informace o konfiguraci zásuvných modulů a podobně.

Poznámka: Praktickým důsledkem ukládání konfiguračních údajů je, že po opětovném spuštění se zobrazí pracovní plocha vývojového prostředí v takovém stavu jako při posledním ukončení práce.

Můžete mít společný pracovní prostor pro všechny projekty nebo můžete kvůli vyšší přehlednosti zvolit pro každý projekt samostatný pracovní prostor. Mezi pracovními prostory se můžete přepínat pomocí volby **File → Switch Workspace**. Po volbě se vývojové prostředí restartuje. Implicitně je nastavena složka *C:\Users\<uživatel>\workspace*. V prvním projektu můžete nechat nastavenou tuto složku, případně můžete postupovat systematicky a vytvořit i složku, do které budete umísťovat svoje projekty.



Obrázek 1.16: Výběr pracovního prostoru

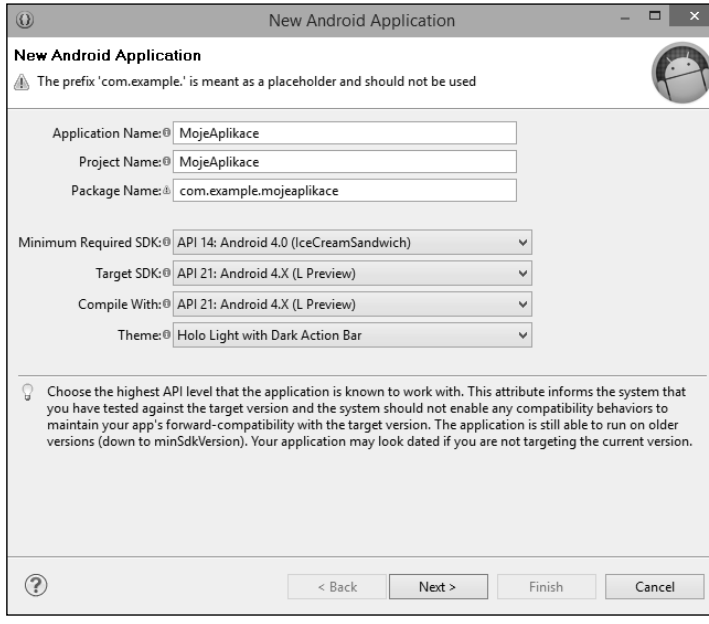
Pomocí nabídky **File → New → Android Application Project** vytvořte nový projekt. Můžete jej nazvat například MojeAplikace. Všimněte si tří polí pro zadání názvu. Název zadaný do pole **Application Name** se bude zobrazovat při spouštění projektu. Do pole **Package Name** se zadává název javového balíčku, do kterého bude projekt aplikace přibalen, například *com.example.mojeaplikace*. Stačí zadat název „MojeAplikace“ do pole **Application Name**, ostatní pole se automaticky vyplní implicitními názvy. Ponechejte i implicitně zadaný výběr maximální a minimální verze SDK. Implicitně je jako aktuální verze nastavená nejvyšší dostupná verze.

V době psaní publikace to byl Android 4.4 (KitKat). Jako nejnižší verze je nastaven Android 2.2 (Froyo). V poli **Theme** vybíráte barevné schéma aplikace. Implicitně je nastaveno světlé pozadí pro aplikaci a tmavé pozadí pro aplikační lištu (**Holo Light with Dark Action Bar**).

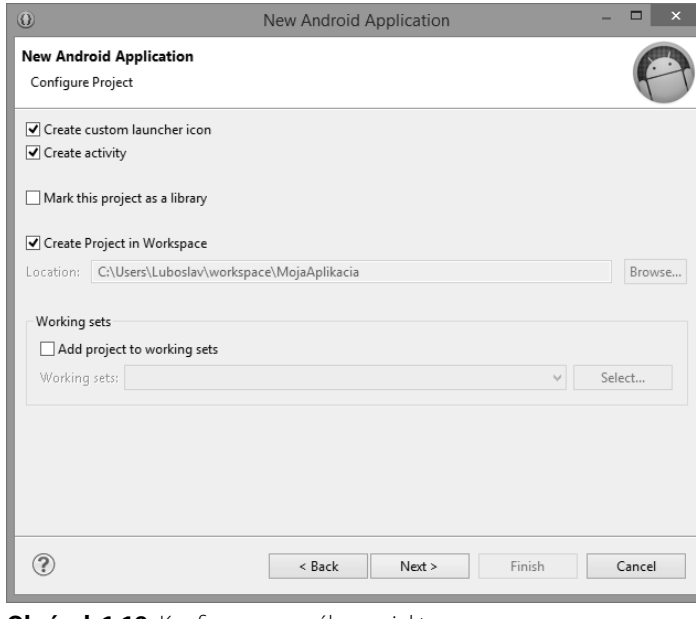
Vyberte verzi Androidu a vyplňte povinné položky **Project name**, **Application name** a **Package name** (ve tvaru *com.example.mojeaplikace*).

V dalším dialogovém okně ponechejte zaškrtnuté položky **Create custom launcher icon**, **Create Activity** a **Create Project in Workspace**.

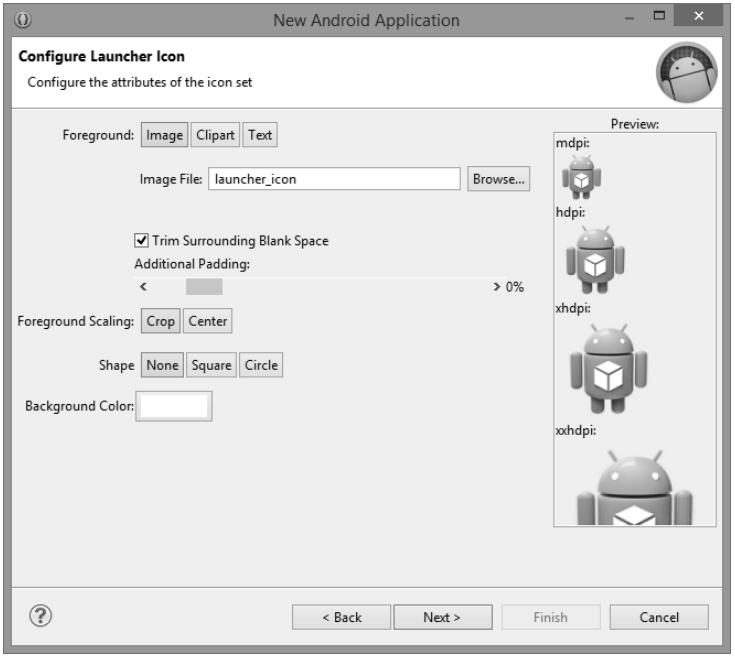
V následujícím dialogovém okně můžete změnit ikony aplikace. Jelikož je tato první aplikace cvičná a slouží k ověření správnosti instalace a konfigurace vývojového prostředí a SDK, není potřeba se v této fázi zdržovat návrhem ikon. Ponechejte implicitně nastavenou ikonu postavy Androida.



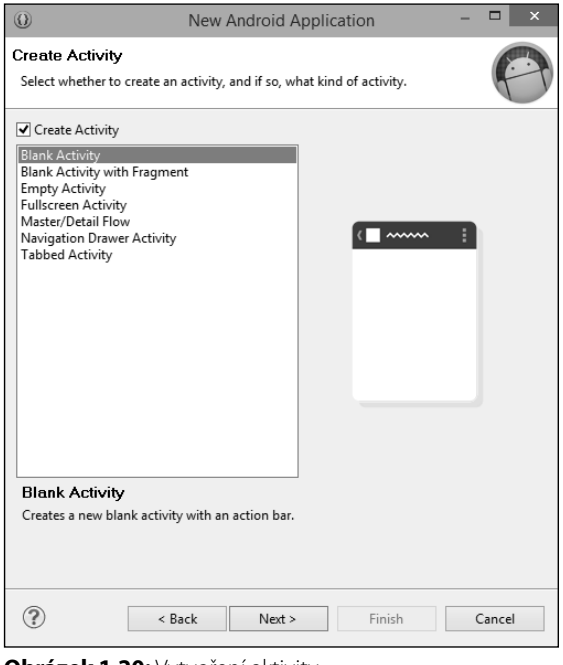
Obrázek 1.17: Vytvoření nového projektu



Obrázek 1.18: Konfigurace nového projektu



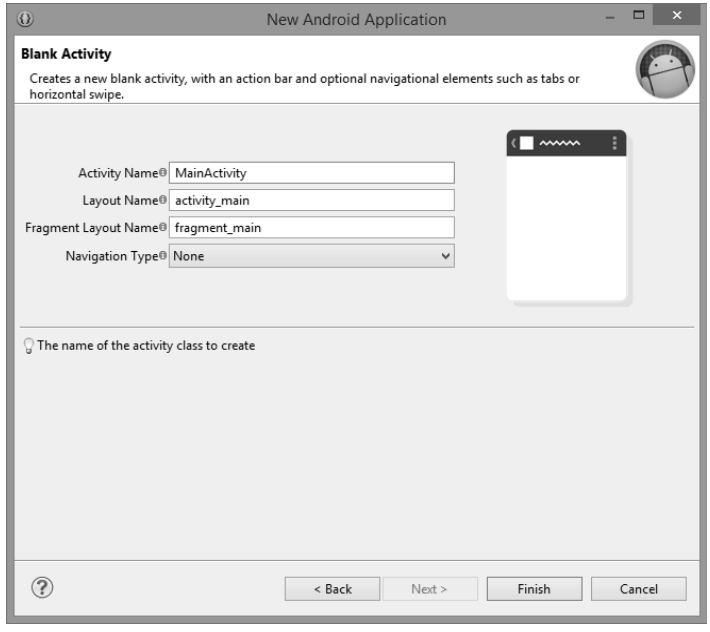
Obrázek 1.19: Ikony pro nově vytvořený projekt



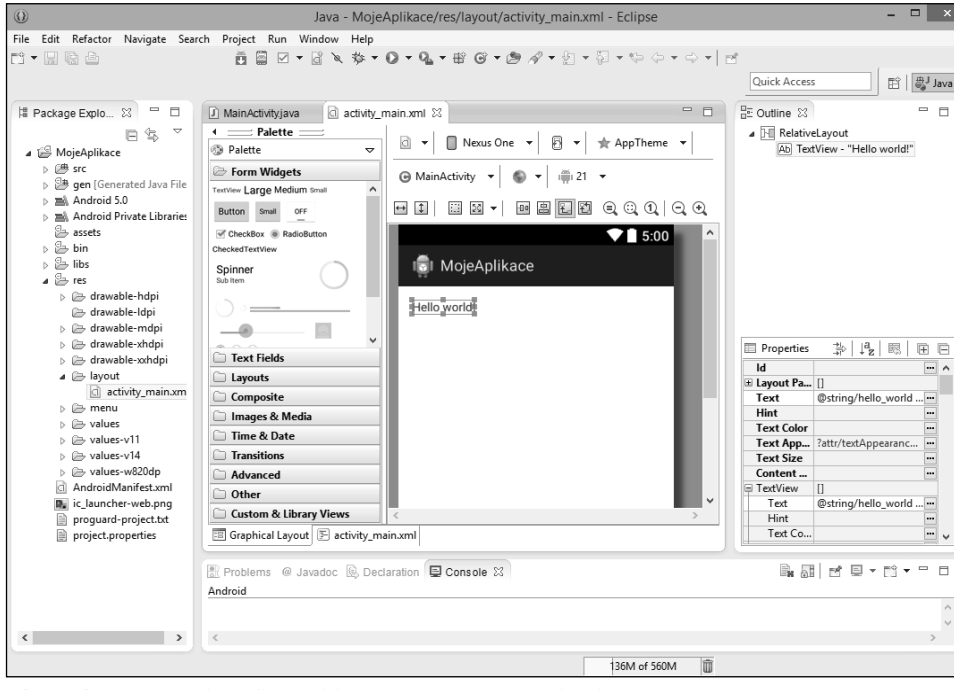
Obrázek 1.20: Vytvoření aktivity

V dalším dialogovém okně vytvoříte objekt typu **Activity**. V tomto projektu ponechejte implicitně vybranou volbu **Blank Activity**.

V následujícím dialogovém okně můžete nakonfigurovat detaily pro vybraný typ aktivity. Ponechejte implicitně nastavené hodnoty.



Obrázek 1.21: Konfigurace vybraného typu aktivity

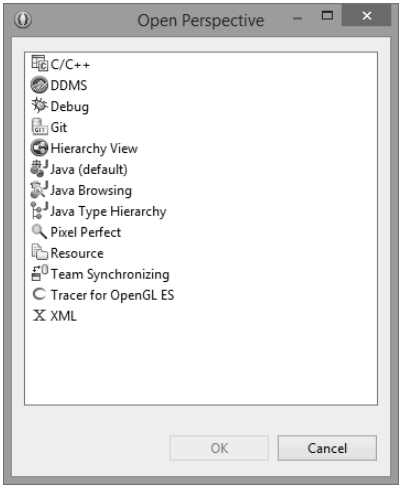


Obrázek 1.22: Projekt Hello World ve vývojovém prostředí Eclipse

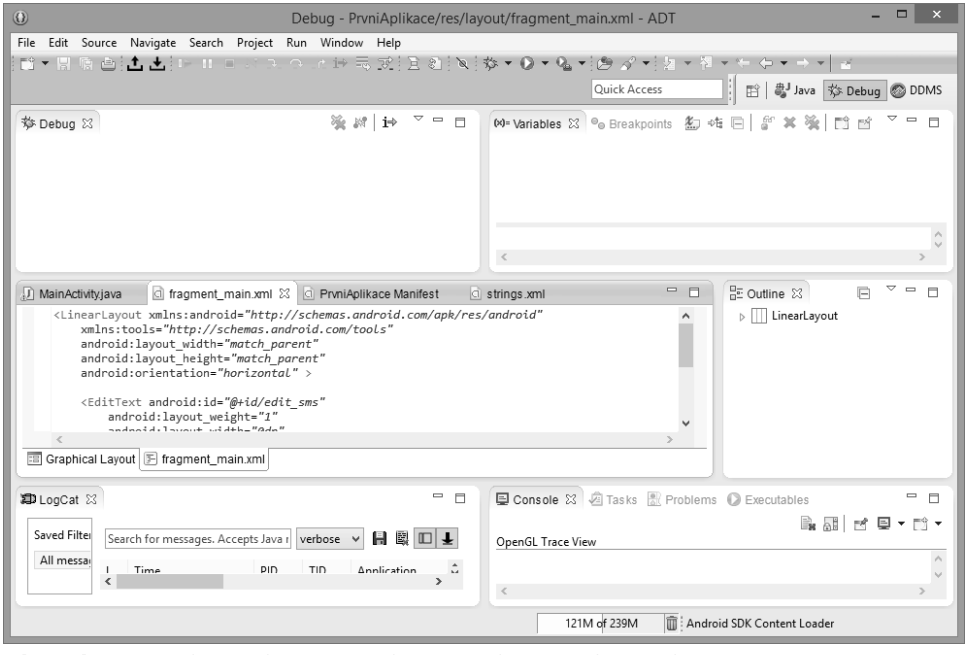
Tlačítkem **Finish** vytvořte projekt. S anatomií projektu se seznámíte v dalších kapitolách. V této fázi se pokusíte projekt spustit, nejprve na emulátoru a následně na reálném zařízení.

Expresní seznámení se s vývojovým prostředím

Po vytvoření projektu a jeho zobrazení ve vývojovém prostředí nastal vhodný okamžik na seznámení se s uživatelským rozhraním vývojového prostředí Eclipse. V levé části je úzké okno **Package Explorer**. Package (balík) slouží k vytváření nových jmenných prostorů – namespaces. Fyzická reprezentace balíku je adresář, který v souborech s příponou *.class* obsahuje přeložené třídy jazyka Java. Ve střední části je okno pro zdrojový kód, případně pro návrhové zobrazení. Vpravo a v dolní části jsou okna pro zobrazení hodnot parametrů, výpisy a podobně. Rozmístění pracovních oken vývojového prostředí se mění v závislosti na činnosti. Jiné je při návrhu prvků uživatelského rozhraní aplikace na platformě Android nazývaných widgety, jiné při psaní kódu v Javě, jiné při ladění.



Obrázek 1.23: Dialogové okno na přepínání perspektiv ve vývojovém prostředí Eclipse

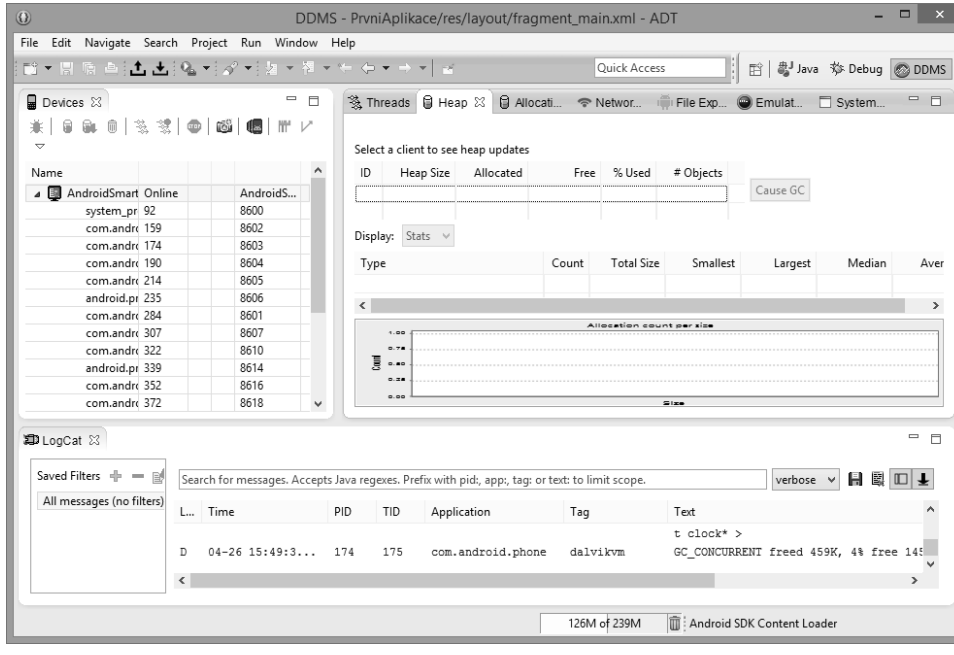


Obrázek 1.24: Rozložení oken vývojového prostředí v perspektivě Debug

Všimněte si v pravé horní části panelu nástrojů sekce implicitně se dvěma tlačítky: čtvercovým tlačítkem se symbolem plus v okně a obdélníkovým tlačítkem s nápisem **Java**. Tato sekce slouží k přepínání perspektiv, tedy rozmístění oken vývojového prostředí. Po stisknutí tlačítka se symbolem plus se zobrazí dialogové okno, pomocí něhož můžete na panel přidat tlačítka pro další perspektivy. Doporučujeme přidat perspektivy **Debug** a **DDMS** (Dalvik Debug Monitor Server).

Dalvik Debug Monitor Server (DDMS)

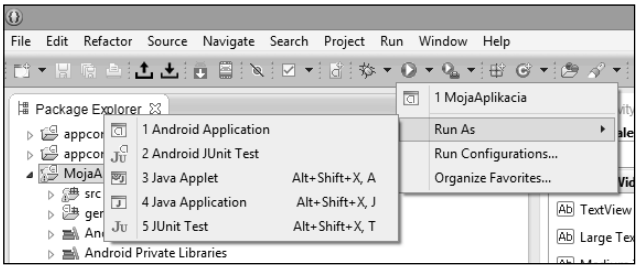
DDMS umožňuje sledovat fungování vaší aplikace na reálném zařízení se systémem Android. Vestavěný modul LogCat umožňuje v reálném čase sledovat všechny procesy, které probíhají na připojeném zařízení. Pro lepší přehlednost je možné události filtrovat a sledovat jen ty, které vás v daném okamžiku zajímají. DDMS umožňuje prohlížení alokované paměti připojeného zařízení a vytvořených objektů, kontrolu paralelně běžících vláken či sledování síťové komunikace.



Obrázek 1.25: Rozložení oken vývojového prostředí v perspektivě DDMS (Dalvik Debug Monitor Server)

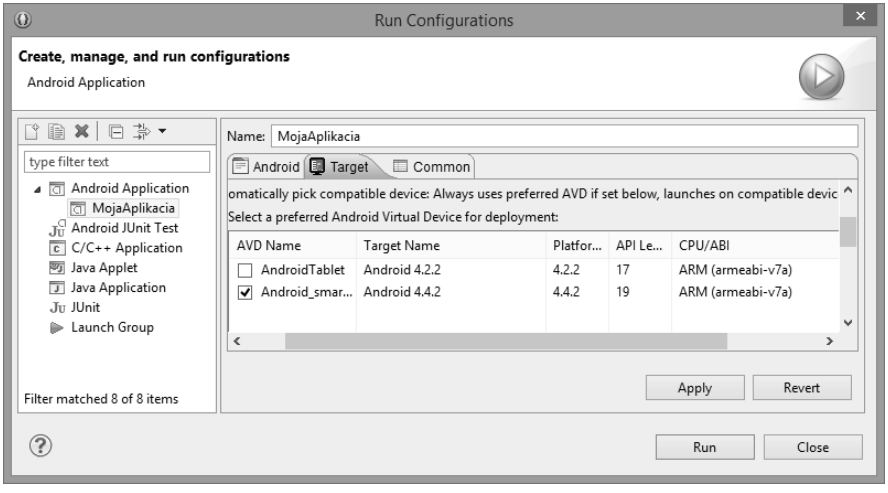
Spuštění aplikace v emulátoru

Klepnutím na zelenou šipku můžete aplikaci spustit v emulátoru mobilního zařízení. V dialogovém okně **Run As** vyberte možnost **Android Application**.



Obrázek 1.26: Spuštění aplikace

Pokud jste vytvořili více emulátorů, například v tomto případě emulátor zařízení typu chytrý telefon a emulátor zařízení typu tablet, pomocí položky **Run Configurations** zobrazíte dialogové okno s výběrem emulátoru, na kterém chcete aplikaci spustit.



Obrázek 1.27: Konfigurace spuštění aplikace

Po náběhu emulátoru a jeho odemknutí se vaše aplikace automaticky spustí.

V okně **Console** v dolní části pracovní plochy můžete sledovat průběh sestavení projektu a jeho zavedení do emulátoru.

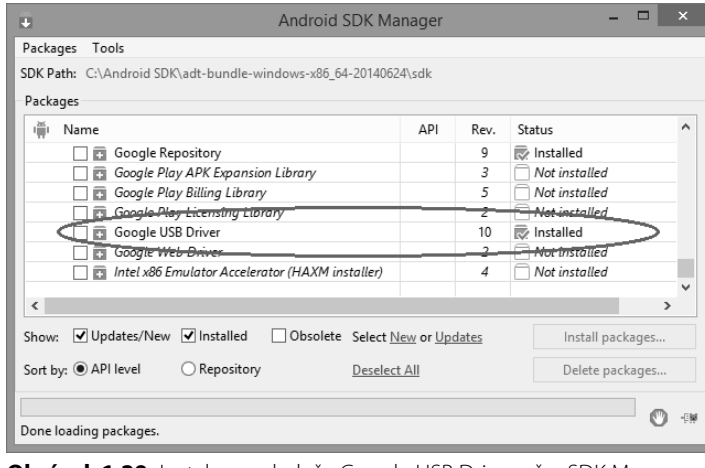
```
[2014-04-06 20:37:25 - MojeAplikace] ------------------------------
[2014-04-06 20:37:25 - MojeAplikace] Android Launch!
[2014-04-06 20:37:25 - MojeAplikace] adb is running normally.
[2014-04-06 20:37:25 - MojeAplikace] Performing
  com.example.mojeaplikace.MainActivity activity launch
[2014-04-06 20:37:25 - MojeAplikace] Automatic Target Mode:
  Preferred AVD 'Android_smartfon' is not available. Launching new emulator.
[2014-04-06 20:37:25 - MojeAplikace] Launching a new emulator
  with Virtual Device 'Android_smartfon'
[2014-04-06 20:37:31 - MojeAplikace] New emulator found: emulator-5554
```

```
[2014-04-06 20:37:31 - MojeAplikace] Waiting for HOME
  ('android.process.acore') to be launched...
```

Obrázek 1.28: Spuštění aplikace v emulátoru

Spuštění aplikace na reálném zařízení

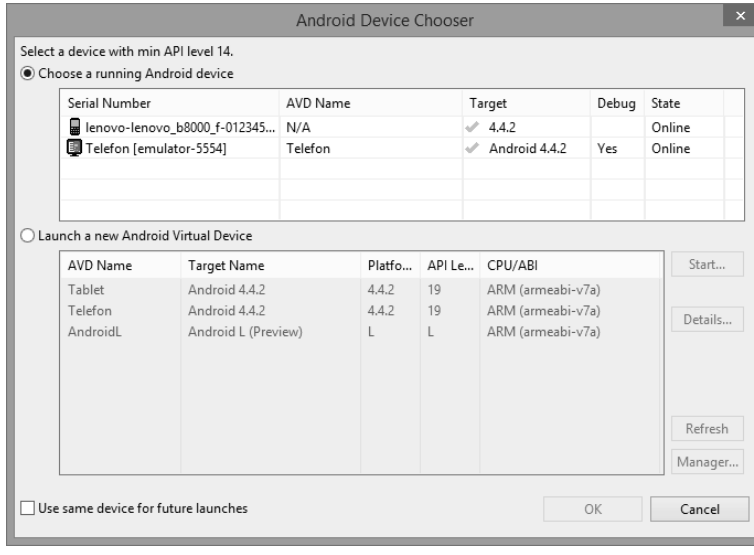
Aby bylo možné aplikaci spustit na reálném zařízení připojeném přes USB, je potřeba na vývojářském počítači nainstalovat USB ovladače pro ADB (Android Debug Bridge). Potom stačí



Obrázek 1.29: Instalace ovladače Google USB Driver přes SDK Manager

připojit zařízení, které má povolené ladění přes USB. Pro zařízení Nexus, případně některá další, postačí Google USB Driver, který doinstalujete přes SDK Manager. Pro ostatní zařízení je potřeba ovladač doinstalovat ze stránky výrobce. Některá zařízení, například Lenovo Yoga, mají možnost po připojení zařízení přes USB nastavit, aby se zařízení chovalo jako virtuální CD ROM, na kterém jsou ovladače.

Po správném nakonfigurování USB ovladače pro ADB se při pokusu o spuštění aplikace zobrazí nabídka možností.



Obrázek 1.30: Nabídka možností spuštění aplikace

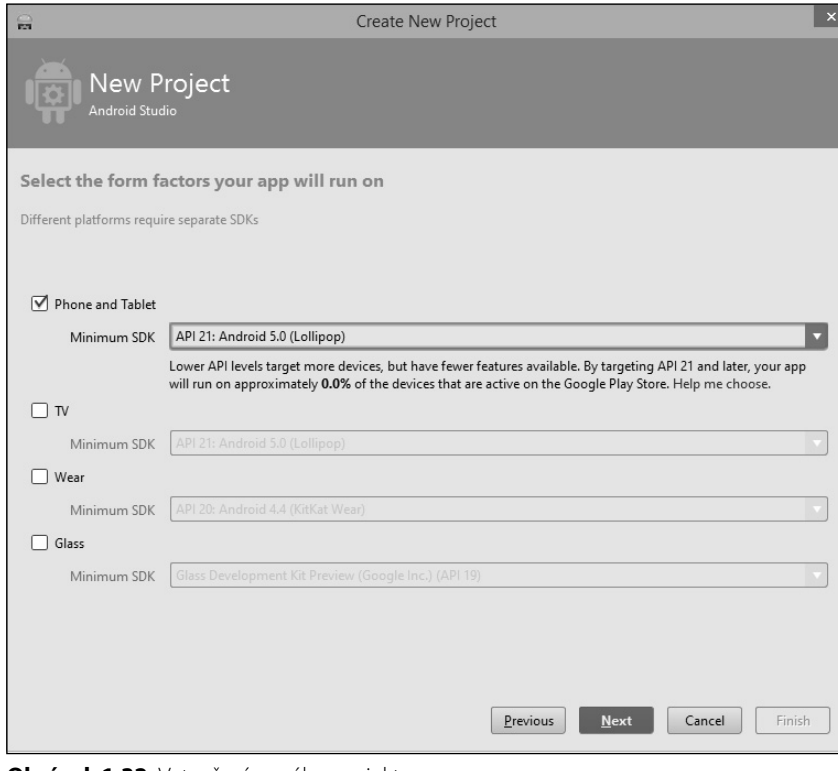
Android Studio

Z předchozí části o instalaci a konfiguraci vývojového prostředí Eclipse je zřejmé, že se nejedná o nástroj pro začátečníky. Také emulátory používané v Eclipse jsou velmi pomalé. Pro ilustraci: ani na vývojářském počítači se čtyřjádrovým procesorem Intel i7 nedokáže emulátor běžet tak rychle jako nejpomalejší reálné zařízení.

Na druhé straně je tu velký zájem o vývoj aplikací pro tuto nejpoužívanější platformu. Východiskem by mohlo být vývojové prostředí Android Studio vyvíjené Googlem na bázi komunitní platformy IntelliJ. Jednodušší návrh uživatelského rozhraní pro různá rozlišení obrazovky přispívá k výraznému zvýšení produktivity práce.

V době psaní publikace byl tento nástroj ve verzi Early Access Preview, tedy před finální verzí. Námi použitá verze 0.8.0 byla stabilní a plně funkční.

Obrázek 1.31: Úvodní obrazovka nástroje Android Studio

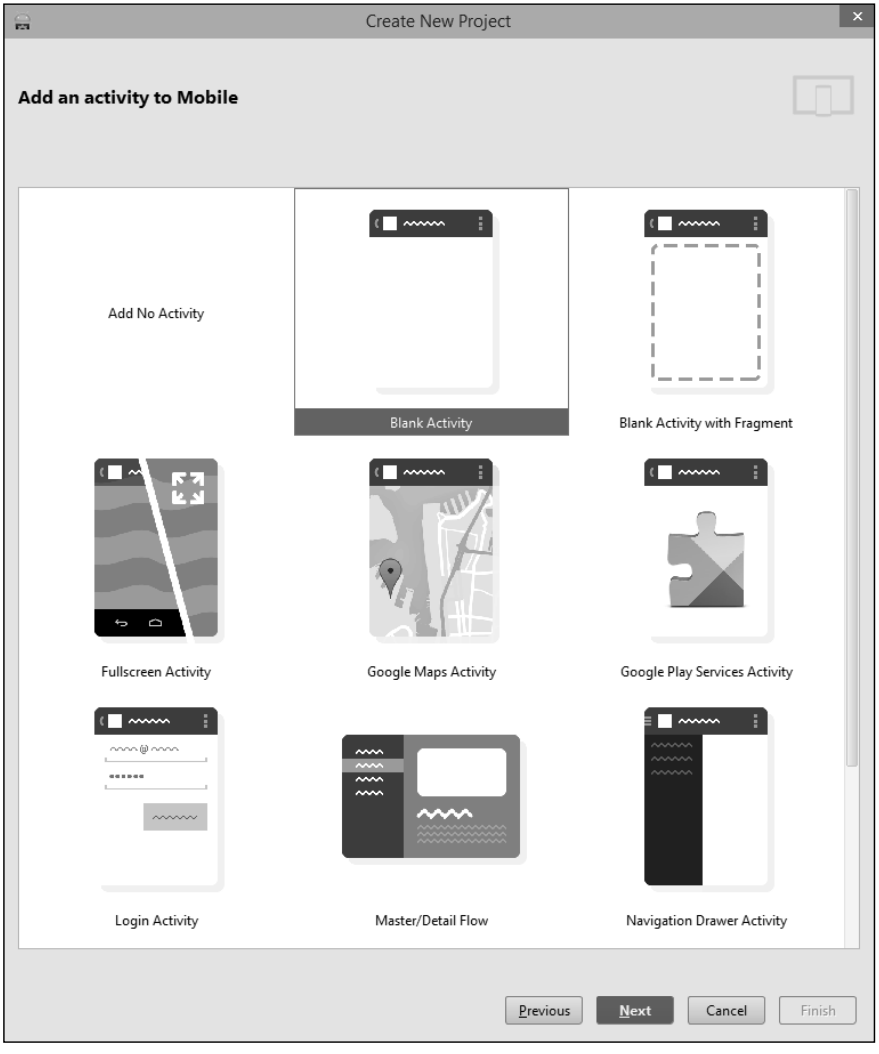


Obrázek 1.32: Vytvoření nového projektu

Instalace Android Studia je jednoduchá a proběhne doslova na jedno klepnutí. Stačí stáhnout instalační soubor a spustit ho. S výjimkou Javy (JDK od Oracle) není potřeba nic doinstalovávat ani konfigurovat. Součástí instalace jsou i kvalitní emulátory zařízení s operačním systémem Android. Po nainstalování se zobrazí okno se základní nabídkou, jejíž součástí je i vytvoření nového projektu.

Postup vytvoření nové aplikace je analogický s postupem v Eclipse. Android Studio využívá méně dialogových oken, která jsou však přehlednější a komplexnější.

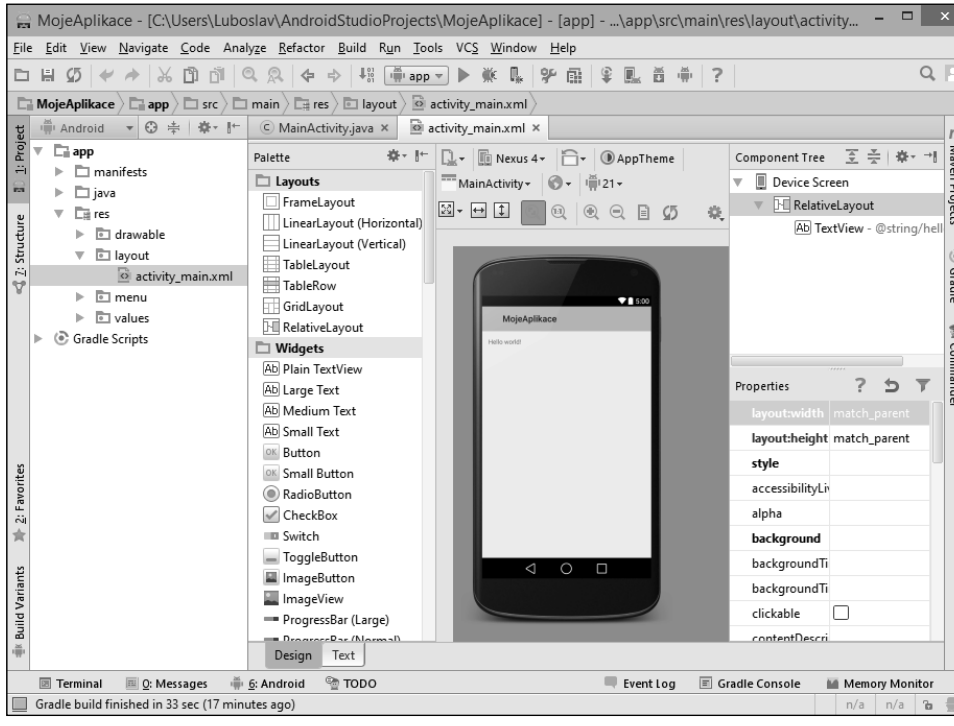
Můžete zvolit několik typů pro hlavní aktivitu aplikace včetně mapové či aktivity typu master-detail.



Obrázek 1.33: Výběr typu hlavní aktivity

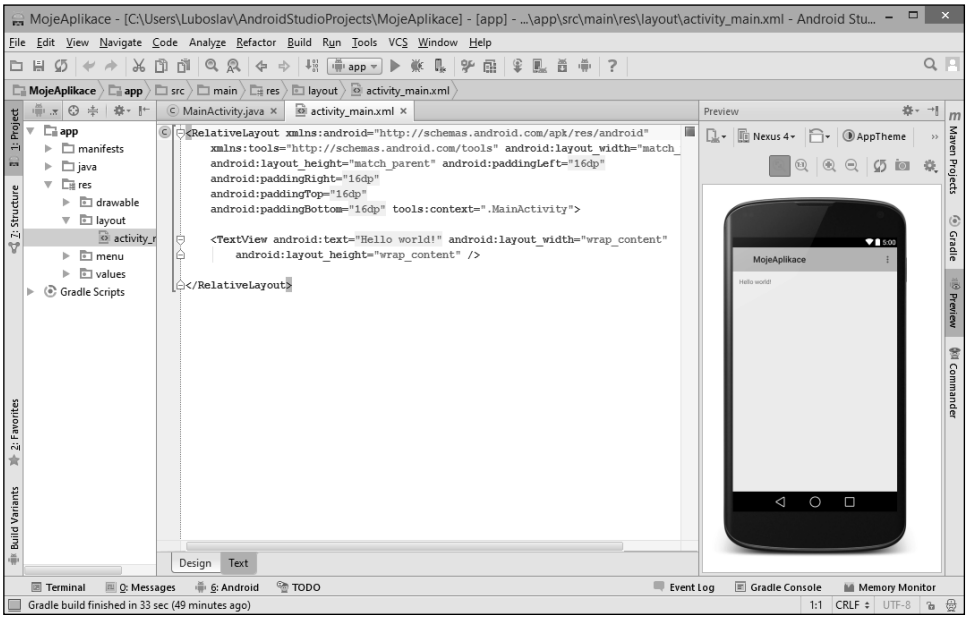
Další postup se odvíjí od nastaveného typu hlavní aktivity. Například pokud jste zvolili typ master-detail, systém vás vyzve k definování názvu položky a skupiny položek, přičemž skupina položek je plurálem od názvu položky; například objednávka/objednávky a podobně.

Podobně jako v Eclipse je i v Android Studiu možné navrhovat design buď v návrhovém módu, nebo přímo v XML souboru. Pokud zvolíte druhou možnost, Android Studio vám automaticky zobrazuje náhled změn. Na rozdíl od Eclipse se nemusíte stále přepínat mezi okny Design a XML.

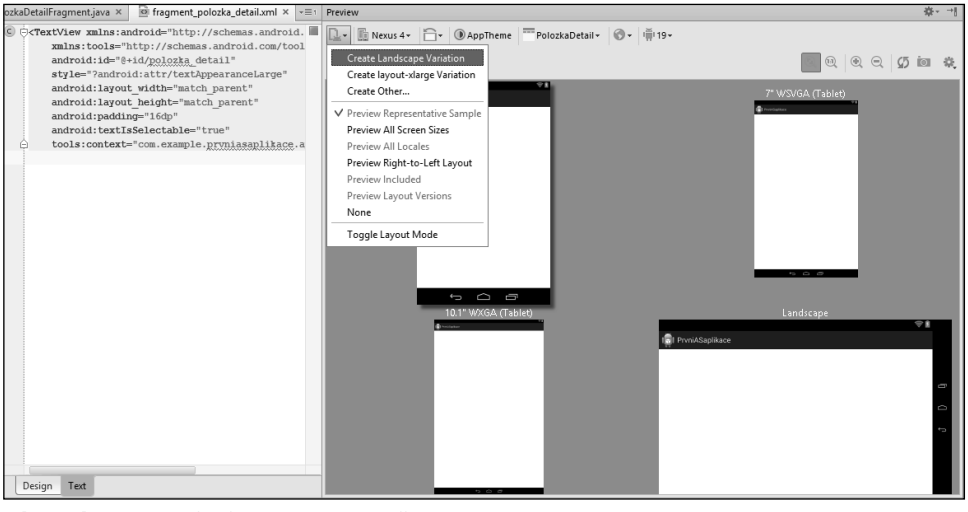


Obrázek 1.34: Uživatelské rozhraní Android Studia

Jelikož je Android implementovaný na množství zařízení s různou úhlopříčkou displeje, různým rozlišením a různou orientací (na výšku, na šířku), vývojáři ocení režim **Preview All Screen Sizes**, který zobrazí náhled změn uživatelského rozhraní v nejčastěji používaných rozlišeních.



Obrázek 1.35: Při tvorbě uživatelského rozhraní přímo v XML kódu se automaticky zobrazuje jeho náhled



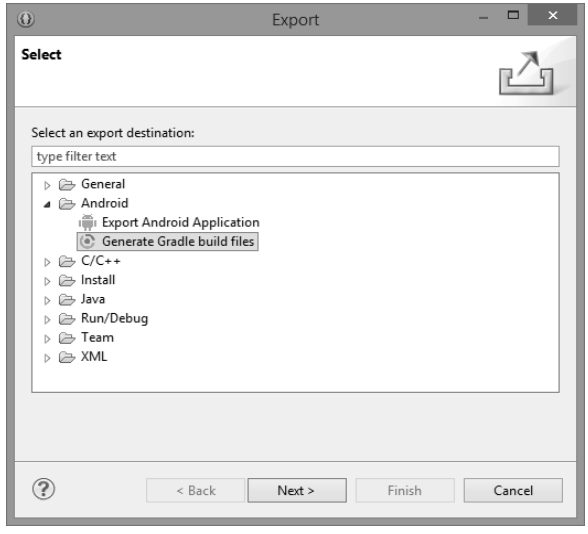
Obrázek 1.36: Mód zobrazení Preview All Screen Sizes

Import projektů z Eclipse

Projekty vytvořené v Eclipse můžete jednoduše importovat do Android Studia. V Eclipse klepněte na **File** → **Export**. Zobrazí se dialogové okno **Export**. Ve složce **Android** zvolte **Generate**

Gradle build files. Vyberte projekt, který chcete exportovat z Eclipse za účelem následného importu do Android Studia.

Poznámka: Nejnovější verze Android Studia umožňuje už přímý import ADT projektů z Eclipse bez nutnosti předchozího exportu.



Obrázek 1.37: Export projektu z Eclipse. Dialogové okno je z vývojového prostředí Eclipse a aktivuje se volbou File → Export.

Embarcadero RAD Studio XE6

Komu by se zdál produkt Embarcadero RAD Studio neznámý, stačí připomenout dva pojmy: Delphi a Borland C++. Vývojové prostředí Delphi, využívající populární programovací jazyk Pascal, vzniklo ve společnosti Borland. Ta se později přejmenovala na Inprise a po pár měsících se znovu vrátila k osvědčenému názvu Borland. Firma se však začala více věnovat ALM (Application Lifecycle Management) řešením a dosavadní vlajkové produkty, tedy vývojářské nástroje a populární databázi InterBase, odstavila na vedlejší kolej, přesněji je předala dceřiné společnosti CodeGear. Tuto nakonec koupila společnost Embarcadero, která se specializovala na vývoj databázových nástrojů pro velké firmy.

Tato část by také mohla mít název „jeden kód pro všechny platformy". Vývojáři v současnosti bojují se dvěma hlavními problémy – efektivitou a různorodostí platforem, ať už serverových, klientských nebo mobilních. Vývojové prostředí RAD Studio od společnosti Embarcadero, které je momentálně k dispozici ve verzi XE6, se snaží řešit oba naznačené problémy, jelikož do značné míry vzájemně souvisí. Vývoj a správa aplikace pomocí specifických nástrojů pro každou platformu jsou nákladné a časově náročné. Zkratka RAD znamená Rapid Application Development a nová verze umožňuje efektivně ve vizuálním prostředí vytvářet společný kód aplikací určených pro platformy Windows, Android, iOS a Mac OS X, všechno v rámci jed-

noho časového plánu bez nutnosti obětovat cokoliv z výkonu aplikací, jelikož tyto jsou kompilovány do nativního kódu.

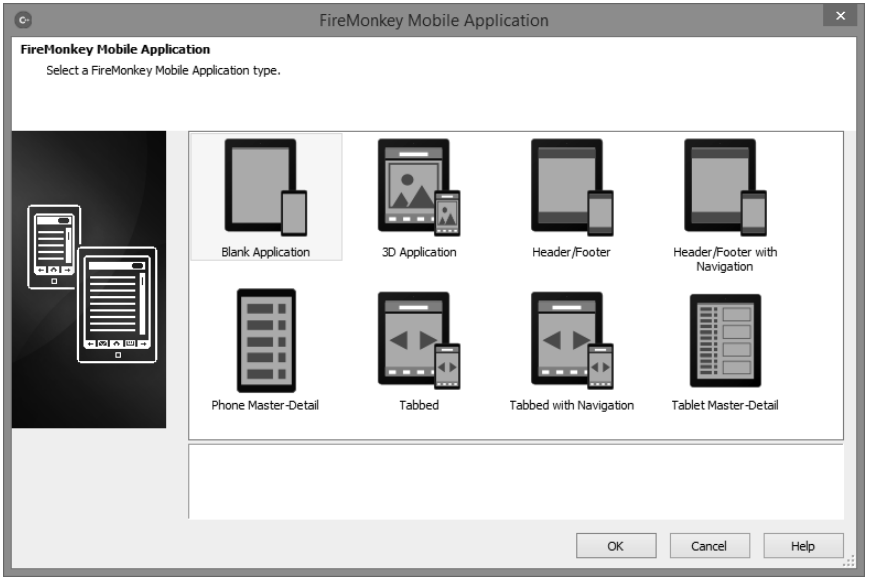
Firmy zabývající se vývojem aplikací si snadno dokáží spočítat, jakým koeficientem by se u nich násobila produktivita vývojářských týmů, pokud by mohly pro nejrůznější platformy použít jeden nástroj, jednotný programovací jazyk a jednotný framework. Multiplatformovost oceníte i po ukončení vývoje v dalších fázích životního cyklu aplikace.

Vizuální návrh aplikací v C++ pro Android

Verze XE6 přináší vizuální aplikační vývojové prostředí jazyka C++ pro platformu Android, což je důležité pro migrující vývojáře, jelikož nativně se aplikace pro Android vytvářejí ve vývojovém prostředí Eclipse v programovacím jazyku Java. RAD Studio XE6 však umožňuje aplikaci jednoduše portovat na všechny nejpoužívanější platformy. Odhadujeme, že bude možné využít přibližně 90 procent společného aplikačního kódu a 60–70 procent návrhu uživatelského rozhraní. Je logické, že bude potřeba změnit uspořádání vizuálních prvků při migraci aplikace z Windows či Mac OS na tablety s iOS či Androidem. Mění se i filozofie ovládání směrem k dotykům. Migrace aplikace na chytré telefony bude mnohem složitější, v mnoha případech bude potřeba kompletně změnit filozofii ovládání. Samozřejmostí je podpora v současnosti nejrozšířenějších verzí Androidu včetně 4.4 KitKat.

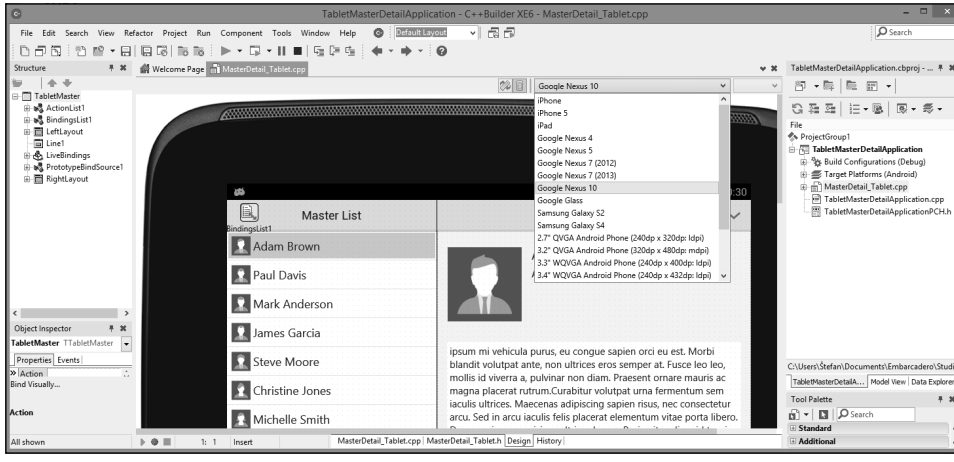
FireMonkey

Cílem tvůrců knihovny Fire Monkey je vývoj rychlých a vizuálně působivých obchodních aplikací na platformách Windows, Mac a iOS. Výsledkem překladu je nativní kód využívající

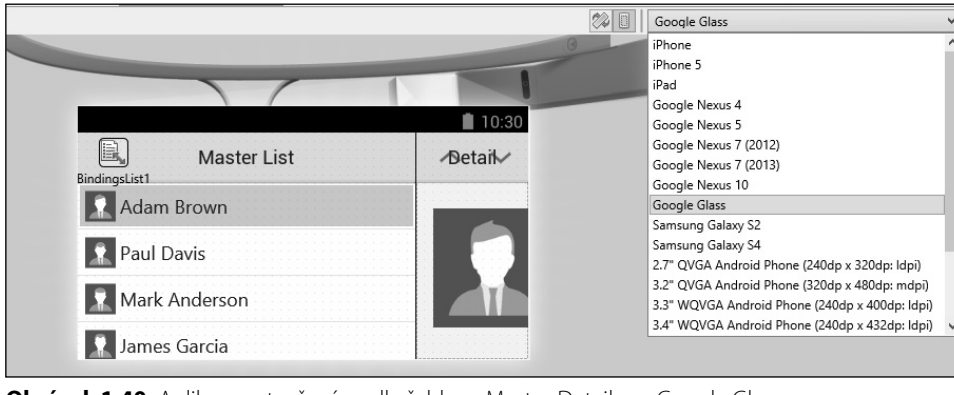


Obrázek 1.38: Šablony pro FireMonkey mobilní aplikace

nejen procesor, ale i grafický akcelerátor, což se výrazně projeví na rychlosti graficky bohatých aplikací. Hardwarovou akceleraci je možné využít i v mobilních zařízeních. FireMonkey umožňuje vývoj komplexních aplikací nejen s vysokým rozlišením grafické prezentační vrstvy na bázi vektorové grafiky, ale pro lepší znázornění „byznys grafiky" je možné využít i trojrozměrné (3D) zobrazení, animaci a širokou škálu obrazových efektů.



Obrázek 1.39: Aplikace vytvořená podle šablony Master-Detail pro tablet



Obrázek 1.40: Aplikace vytvořená podle šablony Master-Detail pro Google Glass

Bez ohledu na to, pro kterou platformu aplikaci vytváříte a který z programovacích jazyků C++ nebo Delphi (populární programovací jazyk vycházející ze syntaxe Pascalu) použijete, kompilátory v RAD XE6 vždy generují nativní, binárně spustitelné soubory pro procesory Intel nebo ARM. To je zárukou vysokého výkonu aplikace a díky rychlé odezvě i pozitivní uživatelské zkušenosti.

App Tethering

Aby se zjednodušila migrace na mobilní platformy a adaptace vývojářů využívajících knihovny vizuálních komponent VCL pro Windows bez vynaložení velkého úsilí, XE6 přináší nové komponenty označené „App Tethering". Tento pojem se dá nejvýstižněji přeložit jako provázání aplikací. V praxi to znamená možnost rozšířit existující aplikace využívající VCL i na mobilní platformy včetně nositelných zařízení, aniž by bylo nutné migrovat celou aplikaci pro Windows. Bude samozřejmě potřeba přizpůsobit uživatelské rozhraní a pro platformy s menším displejem adaptovat jen takové vlastnosti, které dávají smysl na mobilních zařízeních.

Xamarin MonoTouch a Mono for Android

Společnost Xamarin byla založena roku 2011 a zaměřuje se na vývoj multiplatformových nástrojů založených na Mono open-source – Mono Touch (iOS) a Mono for Android. Open-source projekt Mono byl uveden roku 2001 jako open-source verze výkonného prostředí Microsoft .NET. Knihovna Mono Touch byla uvedena na trh roku 2009, Mono for Android následně v březnu 2011. Nástroj Mono umožňuje vývojářům v .NET zaměřeným na jazyk C# vytvářet aplikace pro iOS a Android.

Aplikace vytvořené v nástroji Mono Touch jsou předkompilované do nativní JIT (Just in Time) kompilace a vytváří tím vnořené výkonné prostředí v rámci nativní aplikace. Xamarin také plánuje vydat nový designérský UI nástroj pro Android a poskytnout tak nativní vzhled a dojem z androidových aplikací.

Mnohé z existujících komponent knihoven pro .NET jsou dostupné i pro Mono a Xamarin. Tyto umožňují řešení včetně grafiky, analýzy a vrstvy datové abstrakce. Nástroje Mono Touch a Mono for Android jsou dostupné ve formě bezplatné permanentní trial verze, která nepovoluje publikování aplikací. Cena licence za profesionální verzi začíná na 400 USD za produkt a rok.

Jazyky C# a .NET mají významné zastoupení v podnikové sféře vývoje aplikací. Xamarin se zaměřuje právě na tyto vývojáře, kteří potřebují vytvářet aplikace pro iPhone, iPad a zařízení s Androidem.

Game Maker Studio na vývoj her

Specializovaný nástroj Game Maker Studio na vývoj her získáte na adrese *www.yoyogames.com/studio*. Jedná se o multiplatformní prostředí na jednoduchý vývoj her a aplikací. Vytvoření hry je možné publikovat pro všechny populární platformy: Android, iOS, Windows 8 a Windows Phone 8, HTML5, Facebook i klasický Windows desktop.

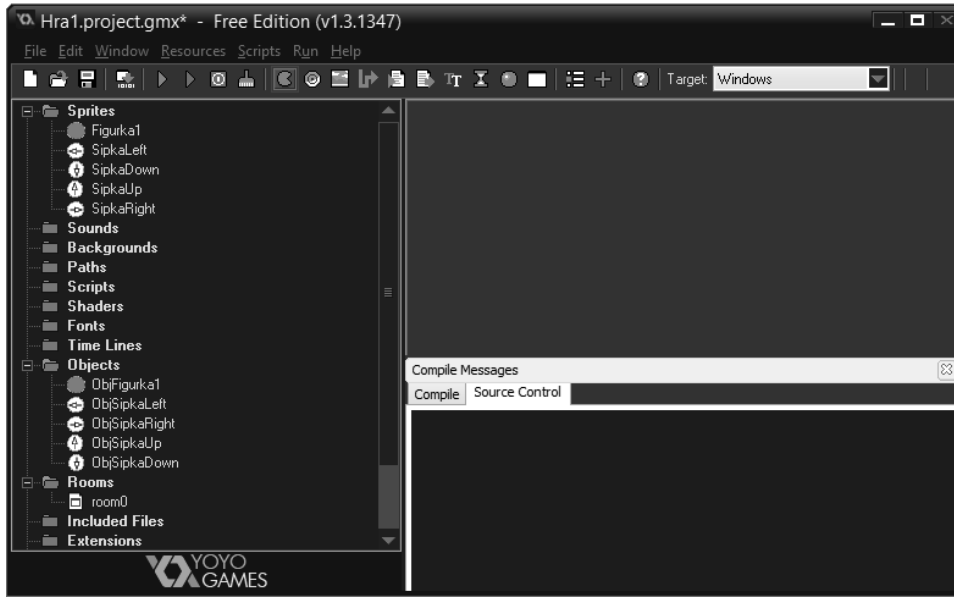
Poznámka: Game Maker Studio Edice Standard je bezplatná, verze Pro, která disponuje pokročilejšími vlastnostmi, například správou textur či možností ladění na mobilním zařízení s Androidem, je k dispozici za 99 dolarů.

Tip: V operačním systému Windows doporučujeme spustit aplikaci Game Maker Studio jako správce.

Příklad vytvoření nejjednodušší hry

Cílem příkladu je postup vytvoření projektu hry, proto jako námět použijeme nejjednodušší „atrapu" hry, jaká se vůbec dá vymyslet – hra umožňuje řídit pohyb figurky pomocí kurzorových kláves bez jakéhokoliv herního algoritmu.

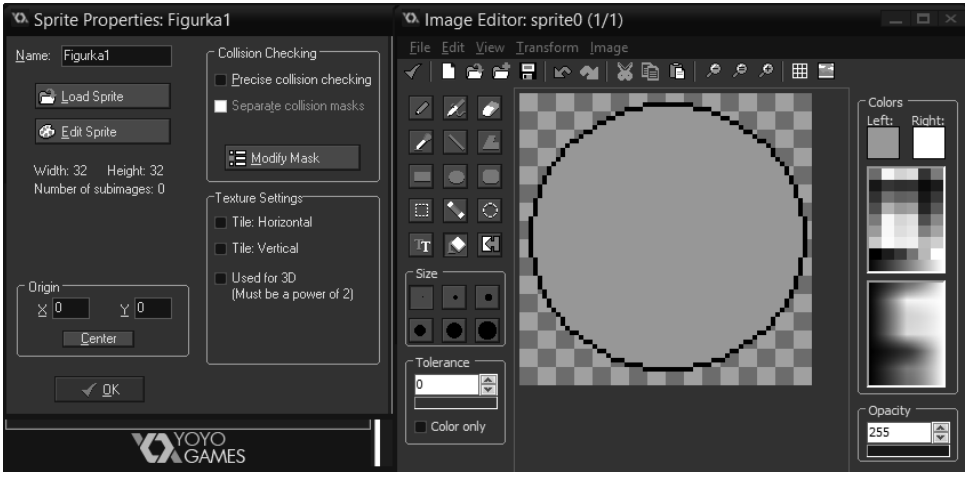
Vytvořte nový projekt s vhodným názvem. Po spuštění si všimněte vlevo složek pro jednotlivé typy objektů, které je možné v nástroji Game Maker Studio vytvořit. Nejprve vytvořte hráče. Klepněte na panelu nástrojů na ikonku se symbolem zelené figurky PacMan.



Obrázek 1.41: Game Maker Studio – složky pro jednotlivé typy objektů

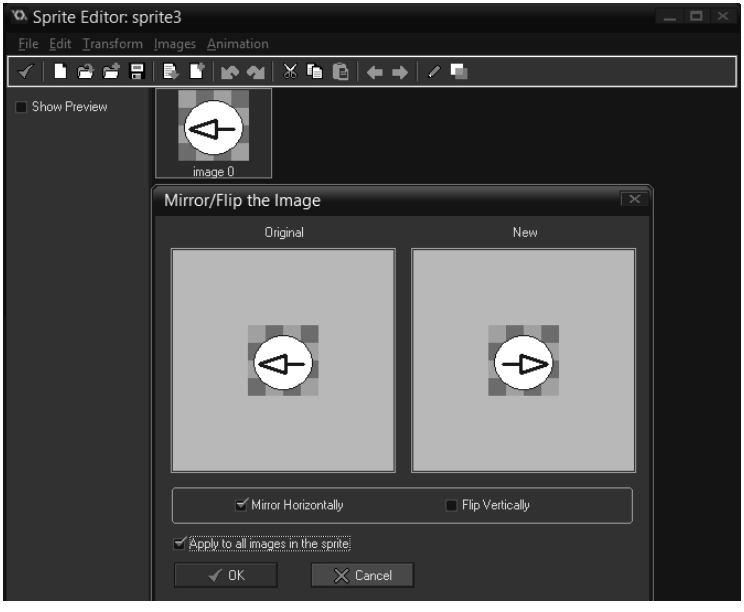
Zobrazí se dialogové okno pro vytvořený sprite (v doslovném překladu skřítek, v terminologii Game Maker Studia grafický návrh herních figurek). Dialogové okno obsahuje i informace o parametrech. Objekt hry, který právě vytváříte, nejprve vhodně pojmenujte. Pomocí tlačítka **Edit Sprite** a položky nabídky **New** vytvořte nový objekt, kde zadáte jeho rozměry. Následně můžete klepnutím na prázdný obrázek nakreslit objekt hry, například postavičku, překážku, stěnu, podlahu a podobně. Doporučujeme pomocí tlačítka s lupou zvětšit měřítko zobrazení. V našem příkladu jsme vytvořili jako symbol figurky zelený kroužek.

Stejným postupem je potřeba vytvořit i ovládací prvky, jelikož zařízení s Androidem nedisponují klávesnicí, pouze dotykovou obrazovkou, na které je potřeba ovládací prvky, například šipky, zobrazit.



Obrázek 1.42: Vytvoření objektu typu sprite

Při vytváření ovládacích prvků se šipkami můžete s výhodou využít funkce **Duplicate** v místní nabídce objektu ve složce **Sprites** v levém úzkém svislém okně. Stačí vám tedy vytvořit jednu šipku na ovládání směru pohybu herní figurky a zbylé tři vytvoříte jejím klonováním pomocí funkce **Duplicate** a následnou vhodnou transformací, například pootočením nebo vytvořením zrcadlového obrazu.



Obrázek 1.43: Vytváření šipek pro ovládání směru pohybu figurky klonováním

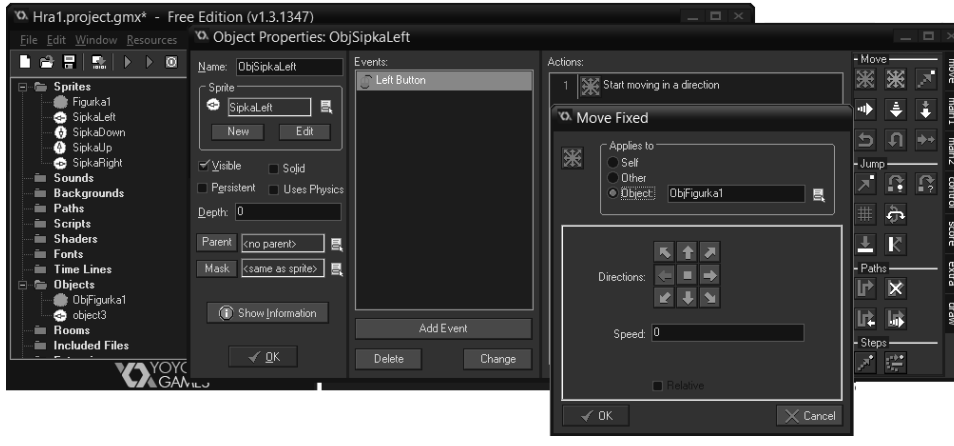
Dalším krokem je na základě figur (sprite), tedy grafických návrhů prvků hry, vytvořit objekty. Objekty je možné vytvářet pomocí ikony se symbolem zeleného kruhu. Do objektu přidejte předtím vytvořený sprite. Parametr **Depth** udává polohu hráče v ose Z.

Pro objekty je potřeba definovat obsluhu události klepnutí na objekt hry, například na šipku. Pomocí tlačítka **Add Event** vytvořte událost. Jako ekvivalent klepnutí se hodí událost **Mouse** → **Left button**.

Obrázek 1.44: Nabídka typů událostí pro objekt

Akci definujete vizuálně v okně **Actions**. Přesuňte na plochu objekt **Move** (ikona se zelenými šipkami, první v sekci **Move**). Následně specifikujte požadovaný směr. Definujte rychlost pohybu, například 5, parametr **Applies to** nastavte na volbu **Object** a zadejte název objektu – figurky, která se má pohybovat.

Herní figurce definujte událost **Mouse** → **GlobalLeftRelease**. Jako akci přidejte ikonu stop, která je mezi směrovými šipkami, a rychlost 0.



Obrázek 1.45: Vytváření objektů a definování akcí

Ve složce **Rooms** vytvořte pomocí volby **Create Room** novou herní místnost – herní plán. Místnost vhodně pojmenujte, například Level 1. Objekt herní figurky umístěte na vhodné místo hracího plánu. Na vhodné místo umístěte také šipky na ovládání pohybu herní figurky. Dbejte na ergonomii, aby bylo možné hru dobře ovládat.