

Vývoj aplikace v J2ME

A development of applications in J2ME

Ondřej Sapík

Bakalářská práce
2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Univerzita Tomáše Bati ve Zlíně
Fakulta technologická
Institut řízení procesů a aplikované informatiky
akademický rok: 2004/2005

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej SAPÍK**
Studijní program: **B 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Vývoj mobilních aplikací v programovacím jazyce Java**

Zásady pro vypracování:

- 1) Vytvořte rešerši na téma způsobu tvorby mobilních aplikací s využitím programovacího jazyka Java.
- 2) Dokument bude mimo jiné popisovat:
 - a. Způsob návrhu implementace mobilní aplikace
 - b. Průběh vývoje aplikace
 - c. Testování vytvořené aplikace
 - d. Způsoby nasazení a distribuce mobilních aplikací
- 3) Proveďte srovnání různých nástrojů, vývojových prostředí a programovacích technik používaných při vývoji mobilních aplikací.
- 4) Na základě vypracované dokumentace vytvořte vzorovou aplikaci "Advanced Calculator", která bude demonstrovat postupy uvedené v teoretické části práce.
- 5) Rozeberte pokud možno všechna omezení která plynou ze základních vlastností použité mobilní platformy.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

- 1) Qusay H. Mahmoud : Naučte se Java 2 Micro Edition pohotová příručka, Grada
- 2) Kim Topley: J2ME v kostce, pohotová referenční příručka, Grada Publishing, 2004
- 3) Bruce Eckel : Myslíme v jazyce Java, knihovna programátora, Grada
- 4) Bret Spell: Java – programujeme profesionálně, Computer Press
- 5) Zdroje z internetu (<http://java.sun.com/j2me/>)

Vedoucí bakalářské práce:

Ing. Michal Bližňák

Institut řízení procesů a aplikované informatiky

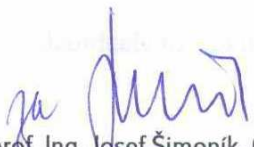
Datum zadání bakalářské práce:

21. února 2005

Termín odevzdání bakalářské práce:

10. června 2005

Ve Zlíně dne 18. února 2005


prof. Ing. Josef Šimoník, CSc.
děkan




prof. Ing. Vladimír Vašek, CSc.
ředitel institutu

ABSTRAKT

Abstrakt česky

Cílem této bakalářské práce je vysvětlit proces vývoje aplikace pro mobilní zařízení v programovacím jazyce Java. Stručně je charakterizována samotná oblast J2ME. Přesněji se věnuji konfiguraci CLDC a profilu MIDL, který je určen pro běžné mobilní telefony. Vznik aplikace je popsán od návrhu, stanovení koncového zařízení, výběru vývojového prostředí, přes samotné programování, testování a ladění aplikace až po distribuci výsledného softwaru. V průběh celého textu jsou rozebrány možnosti a omezení jež z této platformy plynou pro programátory i samotné uživatele.

Abstrakt ve světovém jazyce

Aim of this bachelor thesis is explain suit of development of application for mobile provision in programming language Java. Shortly is characterized alone area J2ME. More precisely I deal with configuration CLDC and profile MIDL, which is specified for current mobile telephones. Origin of application is described from proposal, assessment of end provision, and choice of development environment over alone programming, testing and tuning application up to distribution final software. In all text is out of print possibilities and limitations which from this platform are resulting for programmer and users also.

Poděkování:

Děkuji za užitečné rady z oblasti programování a návrhu aplikací vedoucímu bakalářské práce, panu Ing. Michalu Bližňákovi.

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 08. 06. 2005

.....

jméno bakaláře

OBSAH

OBSAH	6
ÚVOD.....	8
I. TEORETICKÁ ČÁST	10
1 ÚVOD DO J2ME	11
1.1 HISTORIE	11
1.2 KONFIGURACE.....	13
1.3 PROFILY	14
1.3.1 Popis profilů	15
1.4 SPECIFIKACE.....	16
2 NÁVRH APLIKACE	17
2.1 VÝBĚR TELEFONU.....	17
2.1.1 Profil MIDP	17
2.1.2 Hardwarové požadavky	18
2.1.3 Softwarové požadavky	20
2.1.4 Testování telefonů	20
2.2 VÝVOJ KVALITNÍ APLIKACE	23
3 VÝVOJ APLIKACE	24
3.1 UML (UNIFIED MODELING LANGUAGE).....	24
3.1.1 Nástroje pro tvorbu v UML.....	25
3.2 VÝVOJOVÁ PROSTŘEDÍ	26
3.2.1 Souprava J2ME Wireless Toolkit	27
3.2.2 Prostředí IDE.....	27
3.3 PROGRAMOVÁNÍ.....	29
3.3.1 Vytvoření projektu	29
3.3.2 Spuštění aplikace.....	31
4 TESTOVÁNÍ	33
4.1 EMULÁTORY	33
4.2 LADĚNÍ.....	35
4.2.1 Ladění pomocí soupravy Wireless Toolkid.....	36
5 DISTRIBUCE	37
5.1 ZABALENÍ DO BALÍKU JAR.....	37
5.2 DISTRIBUCE.....	39
II. PRAKTICKÁ ČÁST	42
6 APLIKACE BCKALK.....	43
6.1 POPIS A OVLÁDÁNÍ APLIKACE BCKALK	43
6.2 VÝVOJ APLIKACE.....	45
ZÁVĚR	47

SEZNAM POUŽITÉ LITERATURY.....	48
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	50
SEZNAM OBRÁZKŮ	51
SEZNAM TABULEK.....	52
SEZNAM PŘÍLOH.....	53

ÚVOD

V době, kdy se stal mobilní telefon běžnou součástí každodenního života, začalo se rozvíjet také speciální odvětví programovací platformy Java, jež by se uplatňovalo v nových produktech jako jsou osobní počítače PDA a mobilní telefony. Společnost Sun Microsystems (společnost vyvíjející Javu) s výrobcí těchto zařízení proto zavedli pojem J2ME – Java 2 Micro Edition, ten je jedním ze tří balíků Javy (zbylé dvě jsou J2SE – Standard Edition pro běžné použití a J2EE- Enterprise Edition pro podnikové aplikace) dohromady tedy pokrývají celou škálu technologií a specifikací pro jednotlivé skupiny zařízení. Od základu jsou postaveny na vzájemné provázanosti jež má za cíl vytvoření kompletně součinného prostředí (propojení domácností, spotřební elektroniky s internetem nebo mobilními sítěmi). Pro zjednodušení přechodu vývojářů mezi platformami se také dodržuje v největší možné míře vzájemná kompatibilita. Takže se části platform bud' rozšiřují nebo zjednodušují, ale nezdvojují.

Pro zařízení s malou pamětí a nízkým výkonem procesoru je určena oblast tzv. „Micro Java“ do které mimo J2ME patří také platformy JavaCard (technologie chytrých karet) EmbeddedJava (pružně se přizpůsobující platforma pro zakomponovaná zařízení) a PersonalJava (určená pro větší organizéry PDA a komunikační zařízení). Je vidět, že Java je jazyk, který pro každý obor nabídne to své. Pro tento velký rozsah se v této bakalářské práci věnuji pouze platformě J2ME a speciálně pro mobilní telefony stěžejní konfiguraci CLDC a v ní profilu MIDP, který je nejrozšířenější pro svou hardwarovou nenáročnost. A je tedy implementován do většiny nových mobilních přístrojů. Za účelem vývoje aplikace je zde popsán celý postup od návrhu, stanovení koncového zařízení, výběru vývojového prostředí, přes samotné programování aplikace, až po testování a distribuci výsledného softwaru.

Podle určení by se daly aplikace psané pro mobilní telefony rozdělit na tři odvětví. Prvním z nich jsou hry, které jsou velice populární, a jejich vývoj jde rychle kupředu. Rozvíjejí se již také 3D hry jež však mají největší omezení ve výkonu telefonu a nesnadném ovládání. Druhou oblastí jsou aplikace určené pro usnadnění a zpříjemnění života a zpřístupnění známých aplikací používaných na stolních počítačích, jsou to například rozšířené kalkulačky, překladače cizích jazyků, organizéry, kalendáře, aplikace pro čtení knih v mobilním telefonu apod. Poslední odvětví má za cíl dostupnost informací kdekoliv, pouze s pomocí mobilního telefonu. Zde patří internetové prohlížeče, e-mail

klienti, nebo aplikace pro posílání SMS přes GPRS. Toto datové spojení však svou cenou nejvíce brzdí toto zajímavé odvětví.

Tato bakalářská práce má za cíl, popsat celý vývoj mobilní aplikace na platformě J2ME. Zhodnotit všechny prostředky a omezení při návrhu, tvorbě a distribuci midletů. A porovnat softwarové nástroje a pomůcky.

I. TEORETICKÁ ČÁST

1 ÚVOD DO J2ME

1.1 Historie

Firma Sun Microsystems vytvořila na počátku devadesátých let programovací jazyk, který měl být určen pro malou spotřební elektroniku. Výzkumný úkol byl tedy postaven na těchto požadavcích, 1. spolehlivost 2. malé nároky na potřebnou paměť (pro uložení i pro samotný běh programu) a za 3. musí vystačit s malým výkonem mikroprocesoru. Na základě zkušeností vývojového týmu s programovacím jazykem C++ byl vyvinut nový jazyk „Oak“. I přes mnoho předností byl velice náchylný k chybám, které činily programy nespolehlivými. Oak byl navržen tak, aby odstranil či snížil možnost programátorů vytvářet si vlastní problémy. To bylo dosaženo zachycením více chyb při kompilaci a poučením z chyb vyskytujících se v programech psaných v C++. Byly odstraněny např. ukazatele a správu paměti řízenou programátorem.

Prvním prototypem jež využíval jazyk Oak byl Star7, bylo to malé zařízení s dotykovým LCD displejem, bezdrátovou sítí a infračerveným portem. Určeno bylo pro ovládání televize, VCR a také obsahovalo plánovač schůzek. Bohužel na trhu s elektronikou nebyl zájem o tento druh zařízení a tak se neprodal žádný výrobek s touto platformou. Ve stejnou dobu se však začíná rozšiřovat Internet. Firma Sun na tuto skutečnost zareagovala přejmenování Oak na Javu a započala vývoj přenositelného prohlížeče HotJava. Také prodala licenci firmě Netscape (která byla v tu dobu jedničkou na trhu) a ta Javu začlenila do svého prohlížeče, tak se svět internetu poprvé seznámil s Javovskými aplety.

Java vzbudila zájem komerčních koncových uživatelů. A programátorské firmy snížili náklady na vývoj aplikací, které měli běžet na více operačních systémech. V tuto dobu začal firma Sun Javu rozdělovat na několik částí podle určení použití.

Java 2 Standard Edition (J2SE)

Standardní funkcionalita je považována za minimální podporu vyžadovanou pro libovolné prostředí Java. K této edici lze také přidat několik balíčků jako je třeba zabezpečené rozšíření socketů umožňující elektronické obchodování.

Java 2 Enterprise Edition (J2EE)

Tímto Sun reagoval na zvýšený zájem o vývoj na podnikové úrovni. Integruje nové technologie jako např. servlety, Enterprise JavaBeans a JSP (Java Server Pages).

Ironií osudu je, že zatímco společnost Sun vyvíjela Javu pro internet a komerční sféru, začala růst poptávka po menších mobilních zařízeních, což vrátilo Javu ke kořenům. Společnost Sun tedy začala vyvíjet platformy Javy s podmnožinou funkcí specifických pro určité zařízení. Tyto platformy vycházeli z JDK 1.1. Redukováním možností se zajistila možnost programování pro specifické zařízení s malou pamětí a nejrůznějšími periferiemi.

Produkty na bázi JDK 1.1

- JavaCard

Technologie zaměřena na Smart card. Omezení jsou taková, že virtuální stroj JavaCard a malá sada knihoven javových tříd vyžaduje asi 16 kB stálé a 512 bajtů energeticky závislé paměti.

- EmbeddedJava

Určeno pro zakomponované zařízení, které slouží obecně k jednomu účelu, proto není nutné obsáhnout ty části javovské platformy, které daná aplikace nevyžaduje. Umožňuje tedy odstranit jakýkoliv balíček nebo třídu (i metodu třídy) z finální kompilace. Dnes již je vývoj zastaven a přechází se na jeden s profilů CLDC v J2ME.

- PersonalJava

Zařízení podporující tuto platformu(Compaq iPAQ, Nokia 9210) mají až 2 MB ROM a alespoň 1 MB RAM a navíc další paměť pro aplikační software. Personal Java je založena na JDK 1.1.8 a obsahuje plně funkční javový VM. Specifikace určuje jednotlivým balíčkům JDK 1.1.8 zda jsou nezbytné, volitelné nebo upravené. Nezbytný balíček musí obsahovat všechny třídy od svého protějšku JDK 1.1.8. Volitelný balíček, nemusí být implementován, ale je-li přítomen, pak musí být úplný. Upravený balíček musí být přítomen a lišit se může podle pravidel specifikace. Po uvolnění profilů Osobní základ a Osobní přejdou vývojáři na J2ME.

Svět malých mobilních zařízení má na rozdíl od stolních počítačů velkou škálu zařízení s rozdílnými schopnostmi, takže pro něj není možno vyvinout jediný softwarový produkt. J2ME je tedy souborem specifikací, jež definují určitou část platformy. Tyto specifikace se nazývají profily, které rozšiřují základní schopnosti konfigurace. Ty jsou poté přizpůsobeny povaze daného zařízení.

1.2 Konfigurace

Konfigurace má reprezentovat minimální platformu pro dané cílové zařízení, ale nesmí definovat žádné volitelné funkce. Je dána jednotlivými charakteristikami, jako jsou 1. typ a velikost dostupné paměti 2. typ a frekvence procesoru a za 3. typ síťového připojení.

J2ME v současné době definuje dvě konfigurace.

- Konfigurace CLDC (Connected Limited Device Configuration)

Typickou oblastí této konfigurace jsou mobilní telefony a PDA s přibližně 512 kB volné paměti. Vytvořené malé programy se nazývají midlety a pomocí tzv. Wireless Javy si je uživatel může přes mobilní telefon zakoupit i stáhnout.

- Konfigurace CDC (Connected Device Configuration)

Tato konfigurace vyplňuje mezeru mezi CLDC a J2SE. Je tedy určena pro zařízení s většími hardwarovými prostředky (2 MB a více volné paměti, výkonnější procesory). CDC se implementuje do dražších organizátorů, PDA a chytrých telefonů atd.

Každá konfigurace se skládá z virtuálního stroje a standardních tříd. Některá omezení malých zařízení mohou zapříčinit že VM v J2ME nebudou podporovat všechny funkce nebo instrukční bajtové kódy, které poskytuje VM v J2SE. Proto se specifikace doplňuje o JVMs (Java Virtual Machine Specification) a JLS (Java Language Specification) jejíž implementace však není povinná. Například některá zařízení pro něž je určena specifikace CDLC nemají hardware pro výpočty s plovoucí čárkou. VM v CDLC proto nemusí podporovat číselné typy float a double.

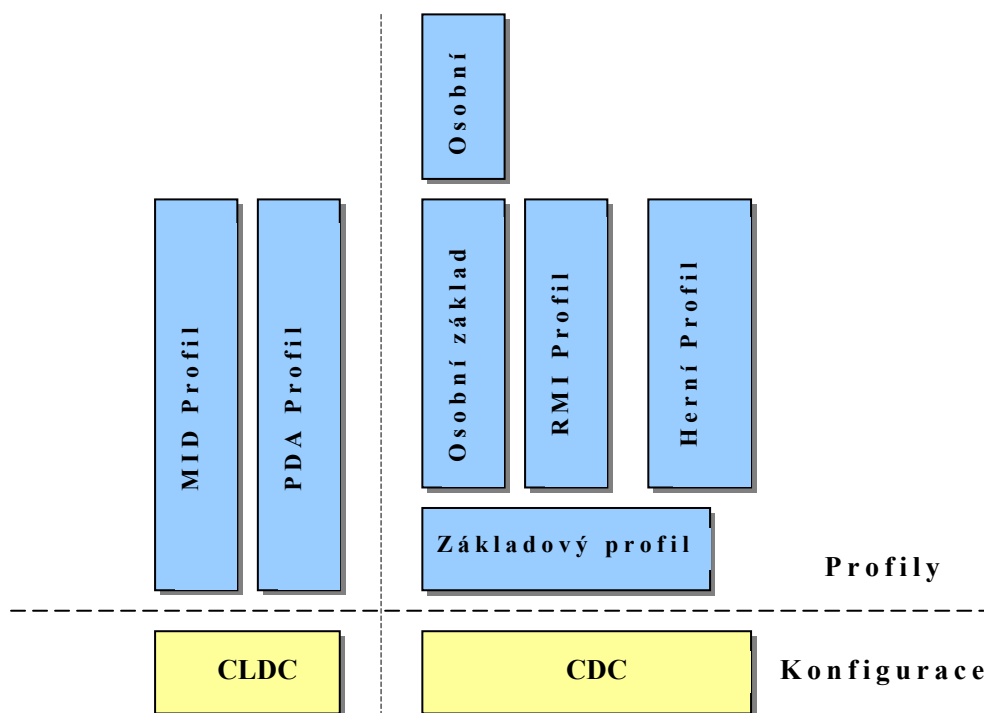
V zařízeních podporujících J2ME se můžeme setkat s VM, které si vytvořili sami výrobci nebo si na VM opatřili licenci třetí strany, jestliže splňuje alespoň minimální požadavky specifikace. Společnost Sun poskytuje referenční implementace s vlastními virtuálními stroji.

Pro CLDC je to KVM. Jde o VM s redukovanými funkcemi, který má velmi malou spotřebu paměti. V referenční implementaci pro CDC je virtuální stroj CVM, který implementuje plný rozsah funkcí VM z J2SE, tak jak je vyžaduje specifikace pro CDC.

Standardní sadu tříd jazyka Java, které konfigurace obsahují, musí být založeny na knihovnách platformy Java 2. Tím je řečeno že J2ME musí využívat třídy a balíčky J2SE, kdekoli je to možné. To podporuje maximální slučitelnost a sdílení zdrojových kódů mezi platformami.

1.3 Profily

Pro určité druhy zařízení nebo pro specifický segment trhu se konfigurace doplňuje o profily, které implementují další třídy. Architektura těchto profilů a návaznost na jednotlivé konfigurace je vidět na obrázku (Obr. 1).



Obr. 1 Konfigurace a profily platformy J2ME

1.3.1 Popis profilů

Profil MIDP (Mobile Information Device Profil)

Tento profil rozšiřuje konfiguraci CLDC o síťové služby. Podporuje základní síťové služby na základě http 1.1. A je základem k tzv. Bezdrátové Javy. Profil je určen pro mobilní telefony s omezeným displejem a pamětí.

Profil PDA (PDAP)

Profil PDA je podobný MIDP, ale je určen pro zařízení s lepším displejem a větší pamětí jako jsou PDA a kapesní počítače např. Palm nebo Handspring.

Základový profil (Foundation Profile)

Tento profil rozšiřuje CDC o téměř všechny standardní knihovny jako je v Java 2 SDK. Je základem pro většinu dalších profilů.

Profil Osobní základ (Personal Basis) a Osobní (Personal)

Rozšiřuje základový profil o funkce uživatelského rozhraní. Pro zařízení s omezenými uživatelskými schopnostmi (např. pouze jednoho aktivní okno) je určen Osobní základ. Platformy, které podporují složitější uživatelské rozhraní, budou naopak používat Osobní profil.

Profil RMI

Přidává knihovny pro vzdálené vyvolávání metod J2SE. Podporuje se jen klientská strana tohoto API.

Herní profil

Tento profil se stále vyvíjí a je určen pro psaní her. A není jasné jak se bude vyvíjet, jestli bude stavět na Základovém profilu nebo bude založen přímo na CDC.

1.4 Specifikace

Konfigurace a profily J2ME se vyvíjely jako součást procesu JCP (Java Community Process), kde se slučují aktéři s různých odvětví s cílem dohodnout specifikaci. Každá konfigurace či profil začala jako žádost JSR (Java Specification Request), která popisuje zadání práce a nastíní oblasti kterým se má věnovat. Skupina odborníků vytvoří specifikaci, která je poté podrobena vnitřní revizí a posléze představena veřejnosti.

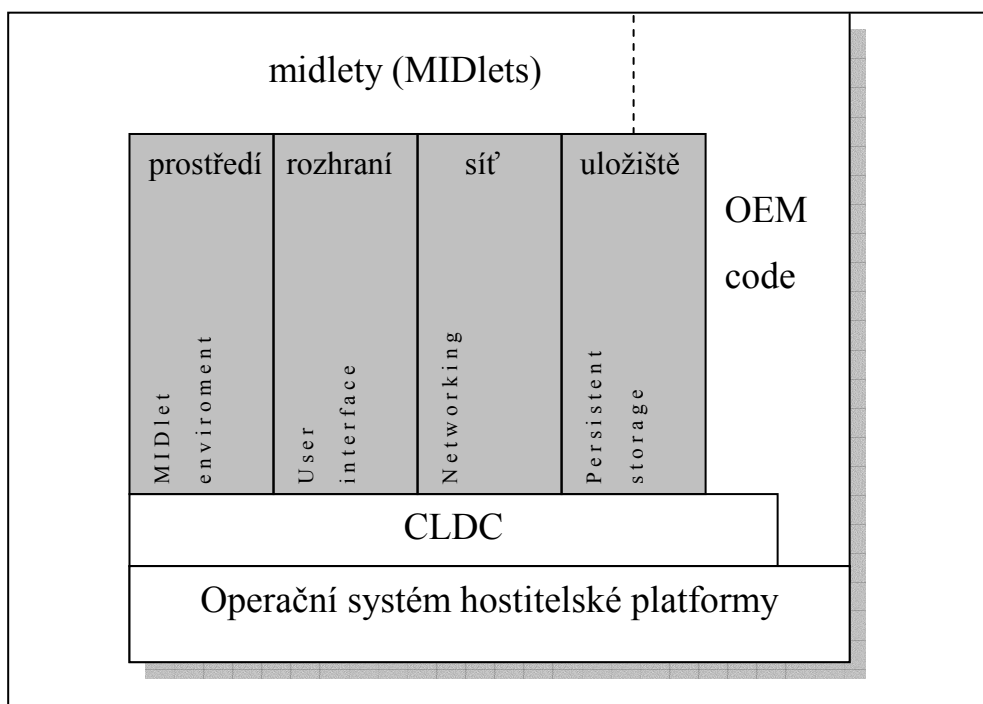
2 NÁVRH APLIKACE

2.1 Výběr telefonu

2.1.1 Profil MIDP

Konfigurace CLDC poskytuje pouze základní vrstvu, jak je vidět na obrázku (Obr. 2), která poskytuje základ pro běh Javy (neobsahuje žádnou interakci s uživatelem, úložnými zařízeními a sítí). Proto se na tuto vrstvu aplikuje řada profilů. Jak již jsem v úvodu zmínil, pro použití na malých zařízeních s omezeným uživatelským rozhraním (malý displej, omezená schopnost vstupů) je určen profil MIDP (Mobile Information Device Profile)

Logické umístění MIDP v softwarové architektuře zařízení ukazuje obrázek (Obr. 2). Software implementující MIDP běží v KVM, který je založen na CLDC. Zároveň poskytuje další služby výhodné pro kód aplikace využívající rozhraní v MIDP. Aplikace se nazývají midlety a nemohou přistupovat k operačnímu systému, protože by se staly nepřenositelnými. Jediným způsob pro přístup k nativním prostředkům platformy je pomocí sestavení nativního kódu do upravené verze virtuálního stroje. Výrobci také do svých produktů zpravidla začlení další kód (OEM code), který poskytuje funkce např. instalace nebo odstranění.



Obr. 2 Softwarová architektura

2.1.2 Hardwarové požadavky

Paměť

Profil definuje minimální požadavky na paměťové místo (které může být a většinou je větší)

- 128 kB RAM pro uložení vlastní implementace MIDP (není započítaná nutná paměť pro CLDC)
- 32 kB RAM pro javovou haldu , což je velmi omezující. Vývojář proto musí dbát na rozumnou alokaci, a jakmile objekt již není nutný, musí se zrušit a zabrané místo vrátit správci paměti.
- 8 kB paměti pro trvalé uložení, toto úložiště však může ztratit data při výměně baterie, proto se očekává, že zařízení poskytne i nějakou paměť pro zálohování.

Výstupní zařízení – Displej

MIDP specifikuje grafické výstupní zařízení tak, aby byla velikost displeje minimálně 96 pixelů na šířku a 54 pixelů na výšku, přičemž by každý pixel měl být čtvercový. Na barevnosti displeje nezáleží, ale musí mít samozřejmě minimálně dvě barvy. Dnes jsou již běžné barevné displeje, které dokáží zobrazit až 65 536 barev (jsou v typových materiálech i větší hloubky barev, ale na malých displejích to lidské oko nerozezná).

Vstupní zařízení – Klávesnice

V tomto směru jsou mobilní telefony a PDA velice rozdílná. Možnosti vstupu uživatele jsou tři základní druhy. Na obrázku (Obr. 3) jsou jejich příklady.

- Úplná alfanumerická klávesnice (+ rychlé zadávání znaku i čísel, - velká velikost)
- Graffiti – psaný text na dotykovém displeji se převádí na znaky (+ nepotřebuje žádnou klávesnici, využívá displej, - chybná detekce)
- Klasická numerická klávesnice 0-9 * # (+ snadné zadávání čísel, - pomalé zadávání znaků)

Vedle kláves pro zadávání znaků a čísel jsou zařízení většinou vybaveny funkčními klávesami, klávesou pro mazání, ale také šipkami nebo jiným polohovacím zařízením s potvrzovacím tlačítkem.



Obr. 3 Druhy vstupů

Připojení k síti

MIDP nevyžaduje trvalé připojení k informační síti, ani podporu protokolu TCP/IP. Požaduje však od výrobce alespoň iluzi, že zařízení podporuje HTTP verze 1.1. To se docílí buď přes zásobník internetového protokolu (IP stack), nebo přemostěním bezdrátového připojení k internetu pomocí brány WAP.

2.1.3 Softwarové požadavky

Zařízení musí obsahovat kód, který přemostí mezery mezi předpoklady referenčního kódu společnosti Sun, samotným hardwarem a operačním systémem. Ten také vytváří softwarovou základnu, jak je vidět na obrázku (Obr. 2). Java od tohoto operačního systému očekává

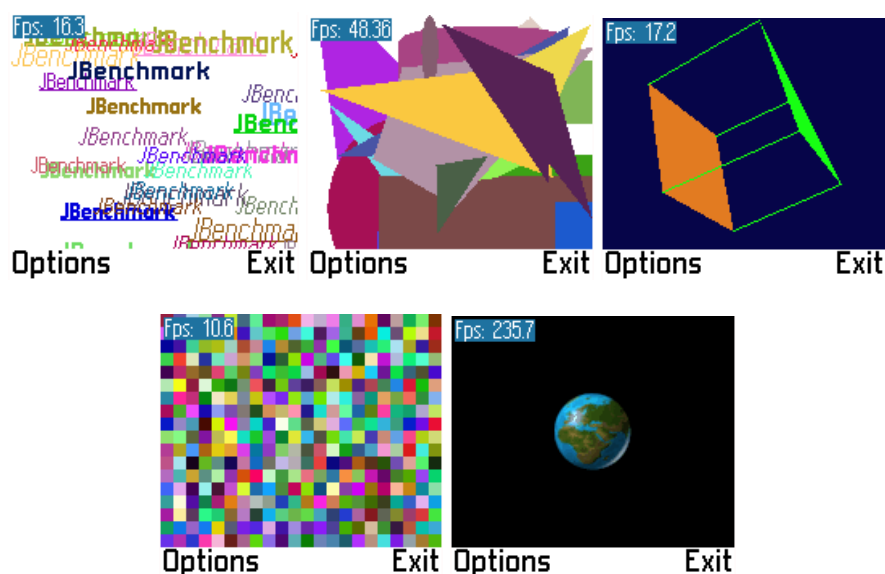
- vytvoření chráněnou oblast v níž bude běžet JVM
- schopnost víceláknového běhu aplikace. Pokud hardwarová platforma tuto možnost nepodporuje musí implementace MIDP poskytnout iluzi. To se docílí sdílením jednoho vlákna operačního systému mezi javovými vlákny.
- podporu síťových funkcí. Na PalmOS je to API na úrovni soketů. U ostatních je to přemostění z vlastní sítě až na internet.
- musí generovat události při stisku a uvolnění klávesy, ukazovacího nebo polohovacího zařízení. Mapování kláves je také pevně dáno.
- zobrazovací zařízení musí připomínat bitově mapované pole, kde se každému pixelu může přiřadit barva v rozsahu jež podporuje dané zařízení.
- hostitelský software musí umožnit přístup k trvalému úložišti pomocí nějakého druhu programového rozhraní.

2.1.4 Testování telefonů

Mobilní telefony podporující javu mají rozdílné hardwarové a softwarové systémové možnosti. Což se projevuje především v rychlosti provádění úkolů dané aplikace. Tato rychlost odezvy je možno experimentálně změřit. K tomuto účelu jsou určeny, ze stolních počítačů již dlouho známé programy - Benchmarky. Pro mobilní telefony je vytvořeno několik testů.

JBenchmark

Naměřené hodnoty jsou vztaženy převážně ke grafickému subsystému. Princip měření pro MIDP 1.0 spočívá ve vykreslování textů, grafických obrazců ve 2D a 3D režimu, vyplňování a animace, jak lze vidět na obrázku (Obr. 4), při níž se měří počet snímků za sekundu. Vyhodnocovací číslo má charakter čím vyšší tím lepší. Pro složitější měření jsou také určeny nové verze JBenchmark2 (pro profil MIDP 2.0) a JBenchmark3D (měření výkonnosti 3D režimu). Program je možno stáhnout z www.jbenchmark.com.



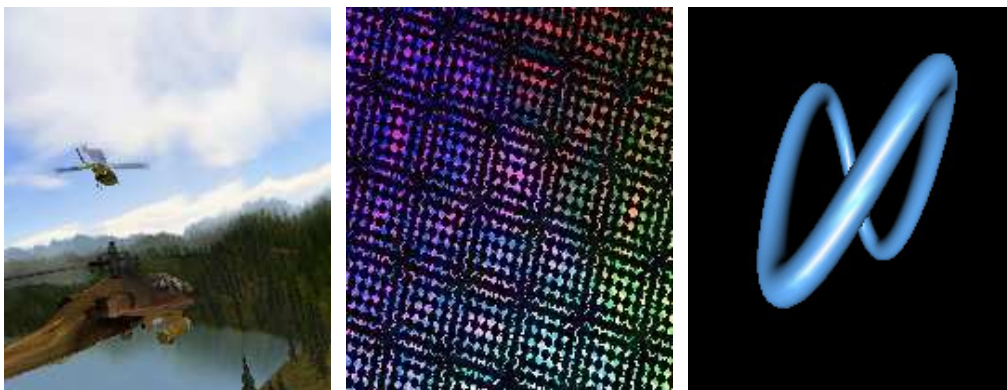
Obr. 4 Testovací obrazce v programu JBenchmark

FloatTest

Tato aplikace měří čas (v milisekundách), který je nutný k několikanásobnému vykonání matematických funkcí. (100krát goniometrické funkce, sčítání násobení a dělení 10 000krát, odmocnina 1 000krát) Jednotlivé časy se sečtou a výsledný čas udává rychlost respektive pomalost Javy. Stáhnou lze z henson.newmail.ru/j2me/Float.htm.

SPMark04 3D Score

Tento test je určen pro telefony s operačním systémem Symbian nebo UIQ. Nároky na paměť telefonu jsou 1,2 MB pro uložení a 3 MB pro samotný běh aplikace. Základní verze otestuje 3D režim (je volně ke stažení). Komerční verze umí navíc otestovat i další hardware telefonu. Na obrázku (Obr. 5) jsou jednotlivé testy 3D režimu. Stránky výrobce jsou www.futuremark.com.



Obr. 5 Testovací obrazce v SPMark 04 3D Score

Výsledky testů na skutečných telefonech

Pro Sony Ericsson T630, který podporuje MDP 1.0 jsem měření provedl v uvedených programech sám a naměřené hodnoty jsou v tabulce (Tab. 1) a (Tab. 2)

Tab. 1 Výsledky testování SE T630 v programu FloatTest

sin	cos	tan	sčítání	násobení	dělení	odmocnina	celkem
9580	5440	15675	11990	32340	47435	35900	157360

Tab. 2 Výsledky testování SE T630 v programu JBenchmark 1.0

text	2D	3D	fill rate	animace	celkem
168	118	66	26	273	651

Tab. 3 Výsledky testování pro různé telefony

	FloatTest	JBenchmark 1.0
Motorola V500	78903	1165
Nokia 6230	11607	1983
Nokia 3100	121261	794
Siemens M55	603884	651
Sony Ericsson T610	164520	696

Za účelem porovnání, jsou v tabulce (Tab. 3) uvedeny naměřené hodnoty pro různé druhy telefonu. Zdroj těchto dat jsou WWW stránky [12] a [9]

2.2 Vývoj kvalitní aplikace

Pokud má aplikace být profesionální, můžeme identifikovat její nároky a při navrhování myslet na následující body.

Bod 1. Shoda původních požadavků zadavatele s navrhovaným řešením.

Bod 2. Přizpůsobitelnost aplikace a schopnost jejího rozšířit.

Bod 3. Možnost opakovaně používat části aplikace.

Bod 4. Spolehlivost aplikace, je stejná reakce při shodných podmínkách.

Bod 5. Robustnost, je zotavení z co nejvíce nedefinovaných stavů.

3 VÝVOJ APLIKACE

3.1 UML (Unified Modeling Language)

Cílem tohoto univerzálního objektově orientovaného modelovacího jazyku je sjednotit používané výrazové prostředky a usnadnit tím návrh a vizualizaci různých typů aplikace. Pro tyto účely obsahuje bohatou sémantiku a syntaxi. Pokud se s tímto nástrojem zachází podle pevně daných pravidel, můžou se výsledky práce sdílet s ostatními návrháři nebo se zadavatelem. A právě tyto pevně daná pravidla a principy jsou největší silou tohoto jazyka. Proto je musí uživatel přesně znát a používat.

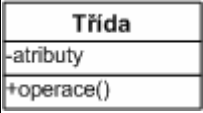
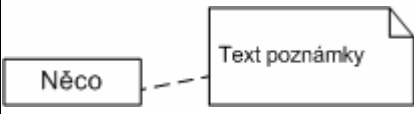
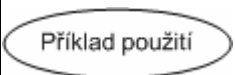
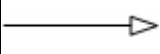
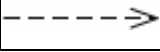
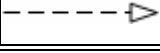
Hlavní charakteristickou vlastností UML je, že softwarový systém je vyjádřen jako struktura spolupracujících částí – modelů.

- Model požadavků na systém - základní požadavky, které jsou kladeny na systém
- Model statické struktury – popisuje druhy objektů a jejich vzájemné vztahy, včetně druhů a vlastností asociací.
- Dynamický model– možné chování objektu (definovány ve statické struktuře)
- Model vzájemného chování – popisuje tok řízení mezi objekty za účelem zajištění potřebného chování celého systému. Charakterizují to tři diagramy, které popisují postup činnosti, průběh a spolupráce objektu na celkové činnosti systému. Charakterizují také obecnou spolupráci objektů mezi sebou.
- Model fyzické implementační struktury – vystihuje technologické a implementační prostředky. (programové moduly)

Modely se skládají ze tří základních stavebních bloků. *Předměty* obsahují samotné elementy modelu. K vzájemnému propojení slouží *Relace*, jež se dále dělí na asociace, závislost, generalizaci a realizaci. A *Diagramy*, které ve vizuální formě vypráví právě jeden konkrétní „příběh“ o aplikaci.

UML je graficky orientovaný modelovací systém, v tabulce (Tab. 4) je několik základních symbolů.

Tab. 4 Základní symboly v jazyce UML

Symbol	Význam
	Třída
	Poznámka k něčemu
	Stav
	Příklad použití
	Balíček
	Začátek (diagram činností)
	Konec (diagram činností)
	vazba
	Zobecnění
	Závislost
	Realizace

Tuto tabulku jsem získal z WWW stránek [8]

3.1.1 Nástroje pro tvorbu v UML

Pro modelování v UML existuje mnoho programů, některé komerční, ale také mnoho open source. Mezi tři základní (tím nemyslím povinné) funkce těchto softwarů patří. Vytváření diagramů z již napsaných zdrojových kódů tzv. reverse engineering. Druhá je samotné

vytváření modelů a tím zapisování svých myšlenek a požadavků. Poslední funkcí, která se dnes bouřlivě rozvíjí je vytváření z těchto diagramů kostry nebo celé aplikace.

ArgoUML – Je program pod volnou BSD licencí, který se stále vyvíjí. Poslední verze podporuje 8 typu diagramů a podle analýzy zdrojového kódu javy umí vytvářet diagramy tříd.

Fujaba – Tento program podporuje pouze tři diagramy (tříd, činností a stavů) má však propracovanou reverzní analýzu a vytváření kostry aplikace. K programu jsou dostupné mnohé pluginy.

Jude Bamboo – Toto je asi nejmenší program vytvářející všech 9 diagramů a reverse engineering. Na svou velikost asi 2 MB poskytuje veškeré základní funkce definované v UML 1.4

Poseidon for UML – tento program je založen na ArgoUML, má však výrazně rozšířené možnosti a podporuje UML 2.0. Verze Community Edition má určité omezení, ale je zdarma. Bohužel od verze 2.0 nepodporuje reverse engineering. Placené verze jsou již velmi vyspělé ale drahé. Cena se pohybuje od 249 USD až po 3 590 USD (90 000Kč)

Umbrello – je pod licencí GPL. Je určen pro operační systém Linux, a v této oblasti je velmi oblíbený. Mezi bezplatnými programy má velké možnosti, podporuje snad veškeré dnes používané programovací jazyky a stále se vyvíjí. Momentální je verze Umbrello 1.4

3.2 Vývojová prostředí

Pro programátory Javy, kteří pracovali s JDK 1.0 a 1.1 nebyly začátky příliš pohodlné. Neexistovalo totiž žádné integrované vývojové prostředí. Museli proto (někteří tak pracují dodnes) psát zdrojový kód v textovém editoru a poté jej kompilovat pomocí příkazového řádku. V posledních letech zejména od vydání platformy Java 2 se staly IDE v J2SE natolik výkonné, že jim můžeme zcela důvěřovat. A většina jich zvládá také rozšířené možnosti jako je průvodce při kompilaci do JAR, nebo exportování dokumentace Javadoc. Sada J2ME Wireless Toolkit (www.java.sun.com/products/j2mewtoolkit/) , stejně jako většina IDE lze jako základní nebo startovací verze stáhnout z internetu zdarma.

3.2.1 Souprava J2ME Wireless Toolkit

Tento vývojový software od společnosti Sun, lze používat buď samostatně (neobsahuje žádné nástroje pro zobrazení nebo editaci zdrojového kódu) nebo ji integrovat do uživatelsky příjemných prostředí IDE. Někteří výrobci mobilů dokonce vyvíjejí vlastní prostředí pro vývoj midletů pro své telefony.

Pro práci v samotném prostředí Wireless Toolkit, je určeno několik pomůcek pro příkazový řádek a GUI.

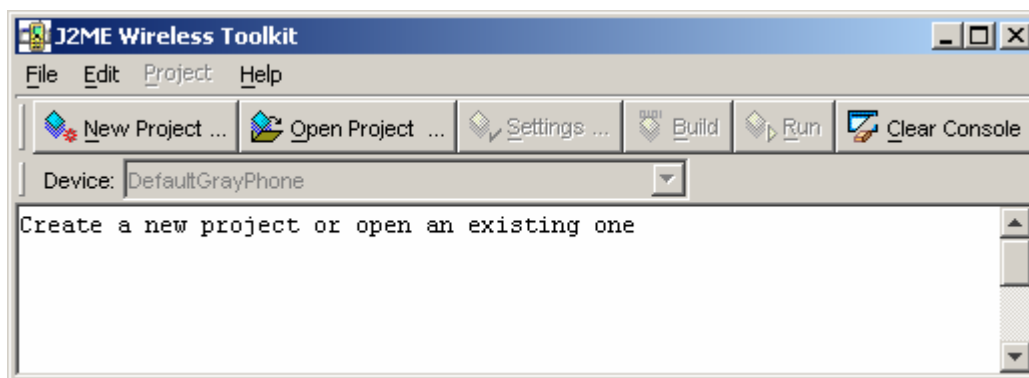
KToolbar - uživatelské rozhraní určené k tvorbě a řízení projektu, umožňuje kompilaci, balení a spouštění midletů. Obrázek (Obr. 6)

Run MIDP Application – spustí vybranou sadu midletů zabalenou v JAR

Default device selection – slouží k výběru typu emulátoru

Preferences – zde lze upravovat funkce emulátorů

Utilities – umožňuje převést sadu midletů na formát .prc (pro zařízení PalmOS). Při emulaci s využitím úložiště je zde možnost tyto soubory odstranit.



Obr. 6 Hlavní okno KToolbar soupravy Wireless Toolkit

3.2.2 Prostředí IDE

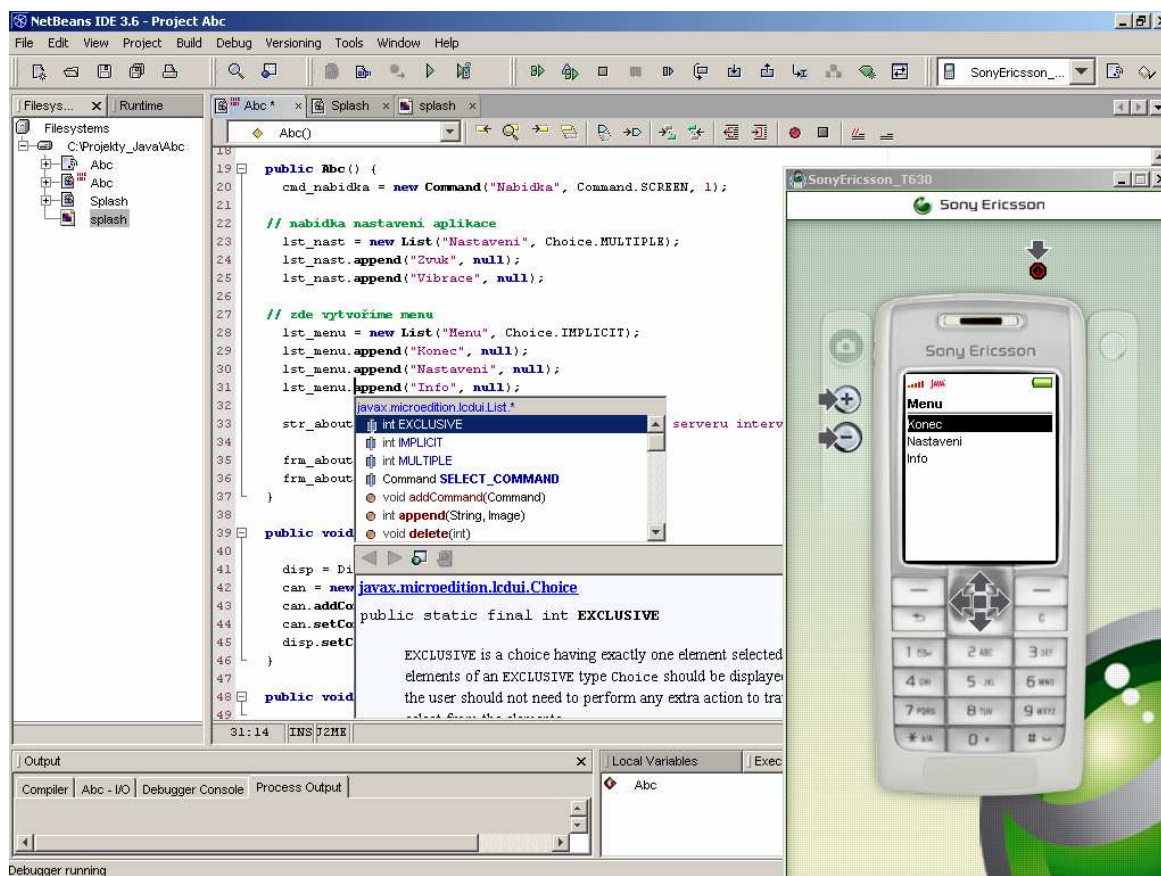
Pro snadnější a pohodlnější programování jsou k dispozici grafické prostředí, které umožňují přehledné psaní a úpravu zdrojového kódu, a další funkce pro programátory. Existují prostředí ve kterých se kód pouze píše a poté se kompiluje v J2MEWTK. Ale většina těchto prostředí v sobě již soupravu Wireless Toolkit implementuje a úzce s ní spolupracuje. To poskytuje používat těchto prostředí k celému vývoji a ladění aplikace.

Funkce IDE

- zvýraznění syntaxe a detekování chyb v syntaxi
- nabídnutí a doplnění očekávané metody, třídy nebo příkazu
- formátování kódu a vyhledávání
- editace grafického rozhraní aplikace
- přehled mezi částmi a soubory projektu
- ladění, vytváření zarážek
- kompilace, vytváření JAR (JAD, manifest), generování JavaDoc dokumentace k projektu
- nápověda k příkazům, zobrazování JavaDoc

NetBeans

Společnost NetBeans vznikla v České republice, a později jí odkoupil Sun, který nastartoval dvě větve vývoje. Forte for Java (dnes Sun ONE studio) je komerční produkt s větší podporou pro J2EE. Druhou je NetBeans, který je opensource projekt a prochází rychlejším vývojem. NetBeans je napsán v Javě, je proto přenositelný na různé operační systémy. Může sloužit jako vývojové prostředí pro jazyky Java, C, C++, XML a HTML. Zde jej popisují samozřejmě z důvodů programování v J2ME. Pro tento účel má implementovanou soupravu J2ME Wireless Toolkit a jako plug-ins lze doinstalovat potřebný emulátor pro daný telefon.



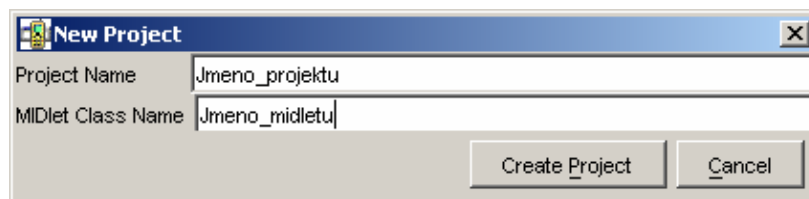
Obr. 7 Prostředí NetBeans IDE 3.6

3.3 Programování

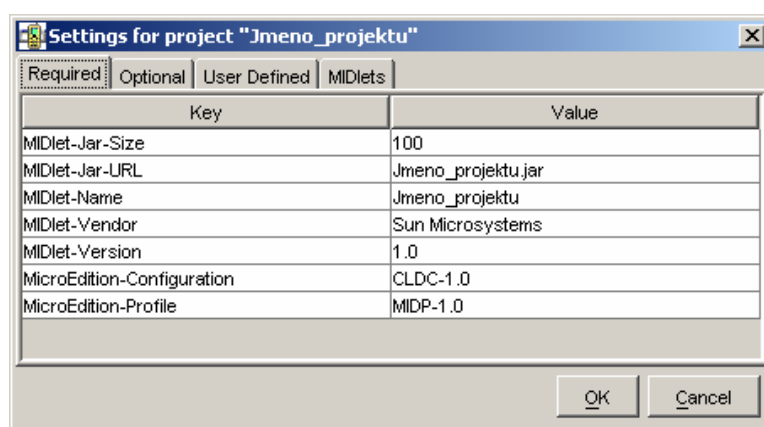
3.3.1 Vytvoření projektu

Pro vytvoření projektu v soupravě J2ME Wireless Toolkit, je určen program KToolBar, který je velmi jednoduchý. Po vytvoření projektu File/New Project a zadání názvu projektu a hlavní třídy (Obr. 8), se otevře nastavení. Které obsahuje čtyři záložky pro zadávání atributů používané k vygenerování manifestu, pro midletové soubory JAR a JAD. Záložka *Required* jak lze vidět na obrázku (Obr. 9) obsahuje povinné atributy (výchozí hodnoty lze použít bez úprav). V záložce MIDlets se definují midlety, jež má sada obsahovat. Pro více midletů se přidávají další řádky, ale důležité aby se v klíších používaly pořadová čísla. Soubory class, které musí JAR obsahovat by se v tomto seznamu neměli objevovat. Záložka User Defined je určena pro uživatelsky definované atributy, které se dají změnit v manifestu nebo souboru JAD. Mohou k nim poté přistupovat jakékoliv

midlety z dané sady a jsou určeny pro snadné přizpůsobení operace midletů, aniž by se vše muselo rekompilovat. Tímto je nastavení dokončeno, změnu lze dodatečně provést v záložce Project/Settings.



Obr. 8 Tvorba nového projektu v KToolbar



Obr. 9 Nastavení povinných atributů pro sadu midletů

Další krok k sestavení midletu je v umístění zdrojového kódu a dalších potřebných souborů do pevně dané hierarchie adresářů. Cesta je určena podle místa instalace Wireless Tolkodu a názvu projektu, například takto: c:\J2MEWTK\apps\Jmeno_projektu. Zde již jsou připraveny tyto čtyři podadresáře:

src – určen pro zdrojové kódy midletů a sdílené třídy

res – doplňující souboru, např. ikony

lib – obsahuje soubory JAR a ZIP s knihovnami třetích stran

bin – zde jsou ukládány soubory JAR, JAD a manifest.

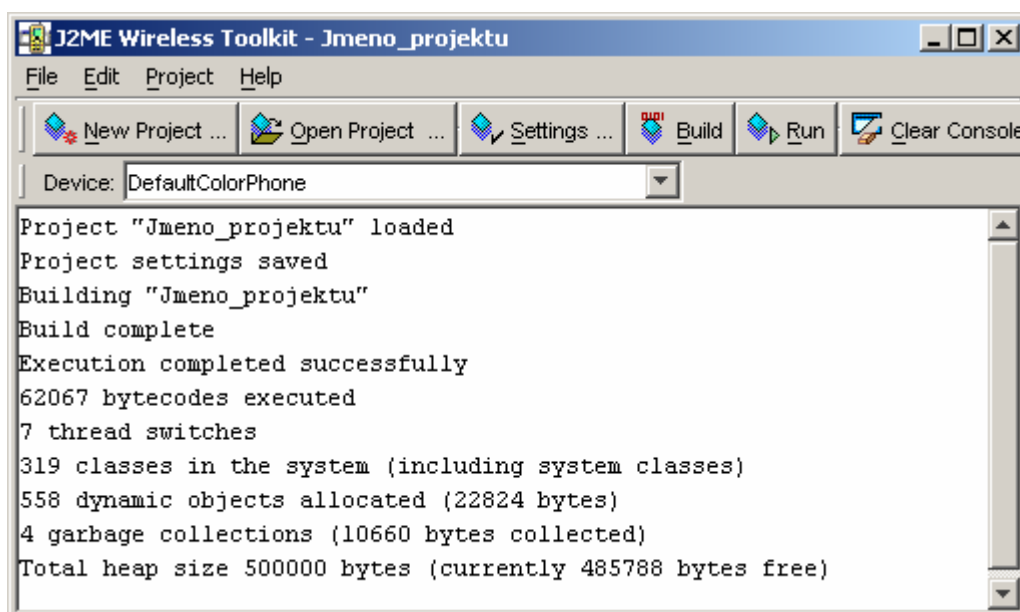
Nyní když jsou všechny potřebné soubory umístěny, může se celý projekt sestavit. Po stisku tlačítka Build se provedou tyto operace.

- Zkompiluje zdrojové soubory z adresáře src a umístí soubory class do nově vytvořeného podadresáře tmpclasses.
- Na všechny soubory v tmpclasses spustí předverifikátor a ověřené soubory umístí do nového adresáře classes. Pokud jsou v adresáři lib umístěny soubory JAR a ZIP zkontroluje je také a umístí do tmplicb.
- do adresáře bin vytvoří soubory manifest a JAD

3.3.2 Spuštění aplikace

Nyní lze midlet poprvé spustit v softwarovém emulátoru, jež lze změnit v programu Default Device Selection. Tlačítkem Run jsou zavedeny třídy midletu, prostředky a všechny přidružené knihovny z podadresářů classes, res a lib. Poté se spustí emulátor a zobrazí seznam midletů, které projekt obsahuje. Tímto se poskytne uživateli možnost zamítnout spuštění midletu, pokud si podle nějakého důvodu myslí, že by midlet mohl ohrozit bezpečnost přístroje. Tato metoda je však velmi slabá protože uživatel vidí pouze název midletu. Po výběru se midlet spustí a konečně je vidět co jsme naprogramovali a jak to funguje (Obr. 11). Pomocí tlačítek emulátoru můžeme samozřejmě midlet ovládat.

Další funkcí Wireless Toolkitu je vytvoření souboru JAR, pomocí Project/ Package / Create Package.



Obr. 10 Výpis při kompilaci a spuštění midletu



*Obr. 11 Spuštěný midlet v
základním emulátoru J2MEWTK*

4 TESTOVÁNÍ

4.1 Emulátory

Pro spuštění midletů nebo jejich testování na stolních počítačích jsou vyvinuty malé programy (moduly) zvané Emulátory. Ty vytvářejí iluzi chování telefonu softwarovou cestou. Jsou implementovány v soupravě J2ME Wireless Toolkit, nebo přímo v grafickém vývojovém prostředí IDE. Výrobci zařízení vyvíjejí emulátory pro jednotlivé typy svých telefonů a jsou volně ke stažení (většinou po bezplatné registraci) na oficiálních internetových stránkách. Tato snaha se výrobcům vrací v prodejnosti telefonu. Protože, kdyby určitý typ telefonu neměl svůj emulátor, nebylo by ani mnoho aplikací, které by mohl využívat. Proto dnes výrobci někdy zveřejní emulátor dříve než samotný telefon, jsou to tzv. návrhové emulátory (které obsahují hlavní charakteristiku), a později zároveň s představením této nové série jsou vydány tzv. užité emulátory, které již napodobují vzhled i chování reálného telefonu.

Rozdíly ve spektru emulátorů jsou vysoké. Na jedné straně jsou emulátory, které jsou pouze graficky přizpůsobeny danému typu. Naproti tomu ty nejvyspělejší emulují veškeré prostředky jako je procesor nebo ostatní čipy v zařízení. Tyto emulátory se poté mnohem více chovají jako skutečný telefon. Jejich vývoj je však velmi zdoluhavý a finančně nákladný, proto takové emulátory neexistují pro všechny druhy zařízení. Mnoho ostatních emulátorů je mezi těmito dvěma extrémy, takže vypadají jako skutečný telefon, ale dokáží vytvořit iluzi jen některých systémových prostředků.

Charakteristické rysy emulátorů založených na J2MEWTK

- Snadné grafické přizpůsobení. Můžeme specifikovat velikost obrazovky v pixelech, barevnou hloubku, druh fontů a možnosti vstupu (rozložení kláves a definování jednotlivých akcí)
- Pokud nepoužíváme výrobcem specifikovaných API je tento nástroj velmi stabilní a robustní.
- Může pracovat v operačních systémech Windows, Linux a Solaris
- Pro vývojáře obsahuje pomocné nástroje simulace, analýza a monitorování síťových funkcí.

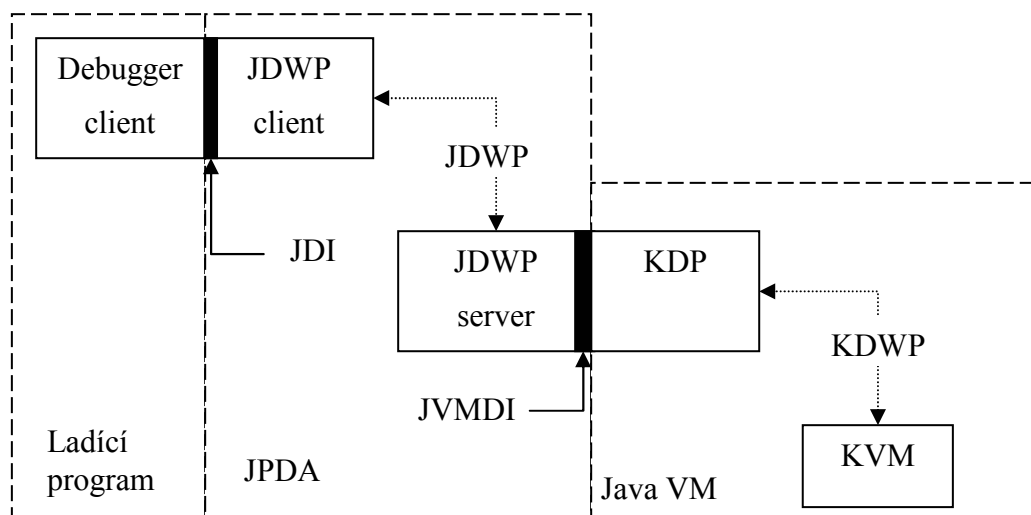
Zajímavý emulátor poskytuje firma SNG. Která podle hesla „Zkusím a koupím“ vyvinula marketingový nástroj pro prodejce mobilních aplikací. Kteří svým zákazníkům umožní odzkoušení nabízených produktu na svých WWW stránkách. Samotný emulátor je založen na technologii Java a dosahuje věrného napodobení chování telefonu a jeho vykreslování. Celá aplikace (webové rozhraní) má velké možnosti nastavení (především dobu, po kterou může zákazník produkt testovat, ale také popis, přizpůsobení ovládání a třeba vložení reklamy). Tento produkt také pečlivě chrání autorská práva na zkoušenou aplikaci. Ihned po nahrání je aplikace transformována do formátu, který může být přehrán pouze SNG emulátorem. Na obrázku (Obr. 12) je ukázka SNG emulátoru ze stránek výrobce www.sng.cz



Obr. 12 SNG emulátor

4.2 Ladění

Pro potřebu ladění aplikace v Javě je zapotřebí do VM umístit prostředky umožňující požadované funkce (vkládání zářezek, vyšetřování a modifikace objektů a zajišťování hlášek). Pro tuto činnost je v Javě 2 definována ladící architektura JPDA (Java Platform Debugger Architecture) zobrazená na obrázku (Obr. 13).



Obr. 13 Architektura JPDA v J2ME

Komunikace mezi ladícím programem JPDA a VM je definována pomocí protokolu JDWP (Java Debug Wire Protocol), který specifikuje zprávy se žádostmi o provedení operací na cílovém VM v souladu s ladícími příkazy. Ty jsou předávány od klienta na server. V opačném směru upozorňuje ladící program na změnu stavu VM. Na každé straně protokolu JPDA jsou izolační vrstvy. JDI (Java Debug Interface) obstarává služby klienta ladícímu programu, který má poté jistotu, že bude fungovat s jakýmkoliv JVM. Druhé rozhraní JVMDI (JVM Debug Interface) umožňuje přijatým operacím, aby byly virtuálním strojem provedeny a změny stavu nahlášeny zpět na server.

Architektura nespecifikuje, jak se mají zprávy JDWP mezi ladícím programem a VM přenášet, zpravidla se používá buď soky (pro vzdálené ladění) nebo sdílená paměť.

Z omezených důvodů KVM se Java VM rozděluje na KDP (KVM Debug proxy – ladící zástupce) a KVM jak je vidět na obrázku (Obr. 13). Komunikaci mezi sebou obstarává KDWP což je speciálně navržená varianta JDWP, která pracuje na soketovém připojení. To umožňuje spouštět ladícího zástupce na stolním počítači a komunikovat

s KVM, který může být na jiném zařízení. V konečném důsledku si ladící program neuvědomuje že nekomunikuje přímo s KVM.

4.2.1 Ladění pomocí soupravy Wireless Toolkit

Wireless Toolkit neobsahuje vlastní ladící program, ale umožňuje přípravu relace mezi emulátorem a ladícím programem, který JDBA podporuje. Standardní KVM také neobsahuje funkce pro ladění, proto J2MEWTK obsahuje kopii (kvm_g), kterou lze pro tuto činnost použít. Umožňuje tedy vytvořit ladicího zástupce, který bude přes architekturu JDBA komunikovat s ladícím programem.

Postup ladění aplikace je následující.

1. Otevřeme projektu v KToolBar a spustíme Debug. Jsme vyzváni k zadání portu, na kterém bude ladicí zástupce čekat na připojení ladicího programu. Do té doby se také emulátor zastaví.
2. Dalším krokem je spuštění ladicího programu. To je většinou součástí IDE, my použijeme NetBeans, kde připojíme adresář src a otevřeme zdrojový soubor midletu a umístíme zarážku.
3. Poté se připojíme na ladicího zástupce pomocí funkce Attach, kde opět vyplníme číslo portu na kterém již ladicí zástupce čeká.
4. Po navázání spojení se v emulátoru rozběhne daná aplikace a probíhá až do chvíle, kdy se dosáhne zarážky. V tu chvíli se program zastaví a my si můžeme prohlédnout proměnné a objekty a případně je modifikovat.

Tento celý postup je samozřejmě jen ilustrativní, a v dnešní době vyspělých IDE rozhraních nemusíme používat takto časově náročný postup. Je totiž celý implementován v grafickém rozhraní a stačí proto celý proces spustit kliknutím na Run in Debugger.

5 DISTRIBUCE

5.1 Zabalení do balíku JAR

Před stáhnutím a instalací aplikace do mobilního telefonu se musí zkompileovaný kód vhodně zabalit. Pro tento účel se vytvoří jediný soubor JAR(Java Archive) ve tvaru `Nazev_midletu.jar`, ve kterém je sada midletů uložena. JAR tedy obsahuje veškeré javové třídy, soubory jež potřebuje midlet ke svému činnosti a v adresáři META-INF soubor MANIFEST.MF. Ve kterém jsou uloženy informace o midletu a souborech, které JAR obsahuje. Podobné informace jsou také uloženy v samostatném souboru JAD (Java Application Descriptor) někdy také nazývány Deskriptor. Oba tyto informační soubory (manifest a JAD) jsou ve formě textového souboru (pokud má být JAD přenosný, měl by být zakódován pomocí ISO-8859-1). Tyto jednoduché texty mají každý řádek ve tvaru:

`název_atributu: hodnota_atributu`

Všechny atributy jež se týkají instalace midletu, mají název začínající MIDlet-. Seznam všech atributů s krátkým popisem je v tabulce (

Tab. 5). Atributy v manifestu (JAR) nebo v JAD jsou buď povinné (P), volitelné (V) nebo jsou ignorovány (I)

Tab. 5 Seznam atributů souboru MANIFEST.MF a JAD

Název atributu	JAR	JAD	Název atributu
MIDlet-Name	P	P	Název midletu zobrazující se uživateli
MIDlet-Version	P	P	Číslo verze midletu ve tvaru a.b.c (1.2.3)
MIDlet-Vendor	P	P	Výrobce midletu
MIDlet-n	P	I	Popis midletu v rámci sady midletů. Místo n je číslo od 1, hodnota je ve tvaru název, ikona, třída
MicroEdition-Profile	P	I	Verze specifikace MIDP (hodnota je např. MIDP-1.0)
MicroEdition-Configuration	P	I	Verze konfigurace J2ME (hodnota je např. CLDC-1.0)

MIDlet-Description	V	V	Popis sady midletů zobrazující se uživateli.
MIDlet-Icon	V	V	Cesta k ikoně reprezentující midlet (format png)
MIDlet-Info-URL	V	V	URL adresa souboru, který podrobněji popisuje sadu midletů.
MIDlet-Data-Size	V	V	Minimální velikost paměti pro data v bajtech. Není-li tento atribut zadán předpokládá se, že midlet nepotřebuje úložiště pro své data.
MIDlet-Jar-URL	I	P	URL adresa souboru JAR. Atribut pouze v JAD
MIDlet-Jar-Size	I	P	Velikost souboru JAR v bajtech.
MIDlet-Install-Nodify	I	V	URL adresa používaná k hlášení úspěchu či neúspěchu midletové instalace.
MIDlet-Delete-Confirm	I	V	Zpráva, která se zobrazí uživateli před odstraněním
MIDlet-specifické atributy	V	V	Atributy ke kterým má midlet přístup a vývojář nebo uživatel je mohou měnit bez rekompilace.

Důvody použití obou souborů

Úkolem manifestu je předat zařízení název a verzi sady midletů v souboru JAR a zároveň specifikovat, které soubory class obsahuje a jak odpovídají jednotlivým midletům. Tyto informace se dají jednoduše z balíčku JAR vyextrahovat a pomocí uživateli se rozhodnout jestli chce midlet nainstalovat. Protože je však soubor JAR docela velký a při použití na mobilních telefonech, kdy je síťový přenos pomalý a drahý, by bylo zbytečné stahovat celý soubor jen aby uživatel dostal informace, podle kterých by se nakonec rozhodl midlet neinstalovat a celý soubor by smazal. Pro tento úkol je určen malý soubor JAD, který uživateli poskytne všechny tyto informace a nabídne adresu a velikost přidruženého JAR souboru. Pokud se hodnoty dublují v obou souborech má přednost hodnota v JAD.

Po zabalení midletu je možno navíc použít funkci Obfuscated Source Code. Která JAR upraví tak, že všechny třídy a jejich proměnné nahradí jednoslovnými názvy. To má poté za důsledek zmenšení souboru a ztížení dekompilace.

5.2 Distribuce

MIDP nespecifikuje jak má uživatel hledat a instalovat midlety. Také se nezmiňuje jak má samotný telefon vybrat a spustit nebo odstranit aplikaci. Tyto funkce jsou z velké části závislé na daném zařízení. Obecně se software v telefonu, který tyto funkce poskytuje nazývá AMS (Application Management Software) v minulosti také používaný termín JAM (Java Application Manager). AMS umožňuje stažení a instalaci midletů ze dvou zdrojů. Synchronizace s počítačem, nebo přes internet (WAP)

Synchronizace s počítačem

Tato možnost přenosu midletu do telefonu je jednoduchá a rychlá. Pro synchronizaci telefonu s počítačem existují tři technologie. Datový kabel, infračervený port a Bluetooth.

Datový kabel - je rozhraní, které využívá sériový přenosu dat. Do počítače se připojuje na COM nebo USB. Tyto kabely jsou specifikovány výrobcem telefonu a některé pouze spojují dva konektory (mobil a PC) a některé obsahují integrovaný obvod k řízení přenosu (hlavně Nokia)

Infračervený port (IrDA) – port komunikuje pomocí infračervených LED diod na vlnové délce světla 875 nm, což předpokládá přímou viditelnost a krátkou vzdálenost mezi zařízeními. Tato technologie je nejvíce využívána u notebooků, ale její rychlost není vysoká.

Bluetooth – je nová bezdrátová technologie, oproti IrDA nevyžaduje přímo viditelnost (využívá radiový přenos) a má rychlejší přenos dat. Pro přenos dat musí mít PC připojen Bluetooth adaptér většinou přes USB. Kromě přenosu dat se může k telefonu připojit i mnoho jiných zařízení (např.handsfree).

Připojení pomocí mobilní sítě

Pro mobilní telefony nebo PDA připojenými přes modem je určen způsob připojení k internetu pomocí WAP. Přes tuto technologii lze stahovat midlety pomocí procesu OTA

(over-the-air provisioning) česky nazývané „poskytování prostřednictvím éteru“. Poskytování OTA není součástí specifikace MIDP 1.0 (v verzi MIDP 2.0 již je specifikována). Ale je dominantní mechanismem pro distribuci midletů. Výrobci midlet umístí na webový server a poté na něj odkazují hypertextovým odkazem.

Proces stahování a instalace midletů pomocí OTA

V prvním bodě si uživatel prohlíží WAP stránku distributora aplikace, zde má poté hypertextový odkaz na danou sadu midletů. Ten ukazuje na soubor JAD. Při jeho aktivaci pošle server soubor JAD a příznak typ MIME text/vnd.sun.j2me.app-descriptor. Na tuto skutečnost reaguje AMS telefonu a zobrazí obsah deskriptoru. Nyní se uživatel může na základě informací rozhodnout jestli zvolenou sadu midletů chce stáhnout a nainstalovat nebo ne. Pokud se uživatel rozhodne pro instalaci, AMS vyhledá v deskriptoru hodnotu MIDlet-Jar-URL a odešle žádost o stažení JAR. Tento soubor server pošle s příznakem typu MIME application/java-archive. Po stažení a uložení do paměti telefonu je midlet nainstalován. Závěrem toho může být odeslání výkazu o instalaci zpět na server (parametr v JAD MIDlet-Install-Notify) ve tvaru stavového kódu a stavové zprávy (příklad stavových kódů: 900 – úspěch, 901 – nedostatek paměti, 902 – zrušeno uživatelem a další)

Kromě instalace midletů poskytuje AMS ještě další funkce pro práci s nimi.

Aktualizace sady midletů

Probíhá stejně jako instalace, ale s tím rozdílem, že na pokyn uživatele se stáhne pouze midlet vyšší verze (parametr MIDlet-Version v JAD)

Výběr a spuštění midletu

AMS poskytuje prostředky k výběru nainstalovaných midletů a jejich spuštění. V mobilních telefonech jsou umístěny v nabídce midletů (většinou sekce „Jiné aplikace“) a v PDA jsou zařazeny jako standardní aplikace telefonu.

Odstraňování midletů

AMS musí také poskytnout funkci pro odstranění sady midletů, a to včetně všech dat v trvalém úložišti, které si aplikace vytvořila. Tím jsou tyto data definitivně ztraceny, a

proto, aby byl uživatel s touto skutečností obeznámen je možno vložit do atribut MIDlet-Delete-Confirm zprávu, která se před odstraněním zobrazí.

II. PRAKTICKÁ ČÁST

6 APLIKACE BCKALK

6.1 Popis a ovládání aplikace BcKalk

BcKalk je midlet pro mobilní telefony v programovacím jazyce J2ME. Využívá konfigurace CLDC a profil MIDP. Je určen k výpočtu matematických výrazů, jež obsahují matematické operace $+$, $-$, $*$, $/$, závorky a reálná čísla. Ovládání je velice jednoduché. Do pole Matematický výraz se zapíše podle stanovených pravidel výraz, jak je vidět na obrázku (Obr. 14). Poté uživatel spustí funkci „Vypočítej“ a v novém okně Výsledek, je vypsán zadaný výraz i s vypočítanou hodnotou výsledku (Obr. 15). BcKalk dále umožňuje zobrazit krátkou nápovědu s popisem programu, jak ukazuje obrázek (Obr. 16)

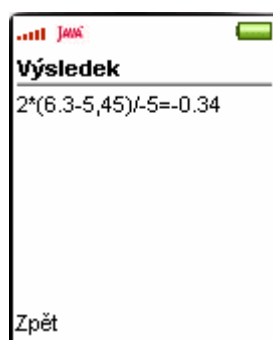
Matematický výraz musí být zadán podle daných matematických pravidel. Je nepřípustné, aby výraz obsahoval rozdílný počet levých a pravých závorek. Při zadávání záporných hodnot nesmí být mínus před závorkou, jak je vidět v tabulce (Tab. 6). A výraz také nesmí obsahovat nepovolené znaky jako jsou třeba písmena. Aplikace na všechny tyto výjimky reaguje zobrazením chybového hlášení s popisem dané chyby, jak je vidět na obrázku (Obr. 17). Poté má uživatel možnost výraz opravit.

Tab. 6 Pravidla pro zadávání matematického výrazu

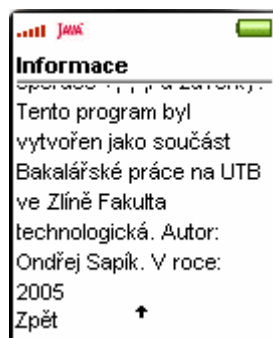
Správně	Špatně	Chybové hlášení
$2*(3-5)$	$2*(3-5$	Závorky nejsou párové.
$-2*(3-5)$ $2*-2$	$2*-(3-5)$	Operátor '-' před závorkou.
$2*5$	$2*5a$	Nečekaný znak 'a'



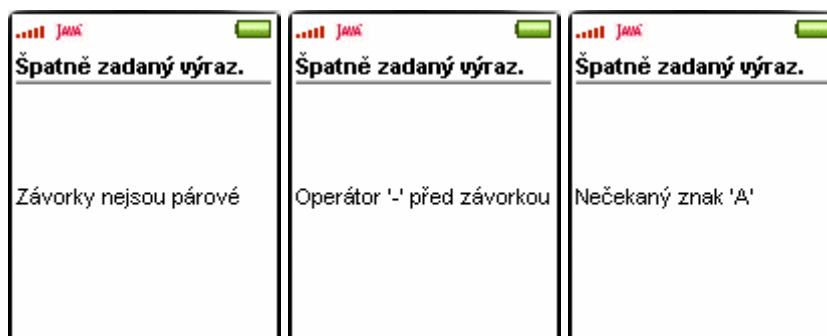
Obr. 14 Pole

matematický výraz

Obr. 15 Výsledek



Obr. 16 Informace



Obr. 17 Chybová hlášení

6.2 Vývoj aplikace

Aplikace je vytvořena pomocí vysokoúrovňových API funkcí, je tedy přenositelná mezi různé telefony. Pro zadávání výrazu a výpis výsledku a informací využívá standardního textového vstupu a výstupu. Ty jsou závislé na daném zařízení a grafická úprava je tedy na různých telefonech jiná a závisí také na schématu, které má uživatel vybrán. Proto také aplikace vypadá jako by byla součástí telefonu. Protože J2ME nepodporuje typ Float ani Double, a neposkytuje ani žádné rozšiřující matematické funkce. Byla vyvinuta knihovna Float, která tyto nedostatky do midletu doplňuje. Aplikace tedy využívá balíček Float pro výpočet matematických operací a pro převod řetězce znaků (které představují číslo) do tohoto typu.

Základní údaje o třídě Float:

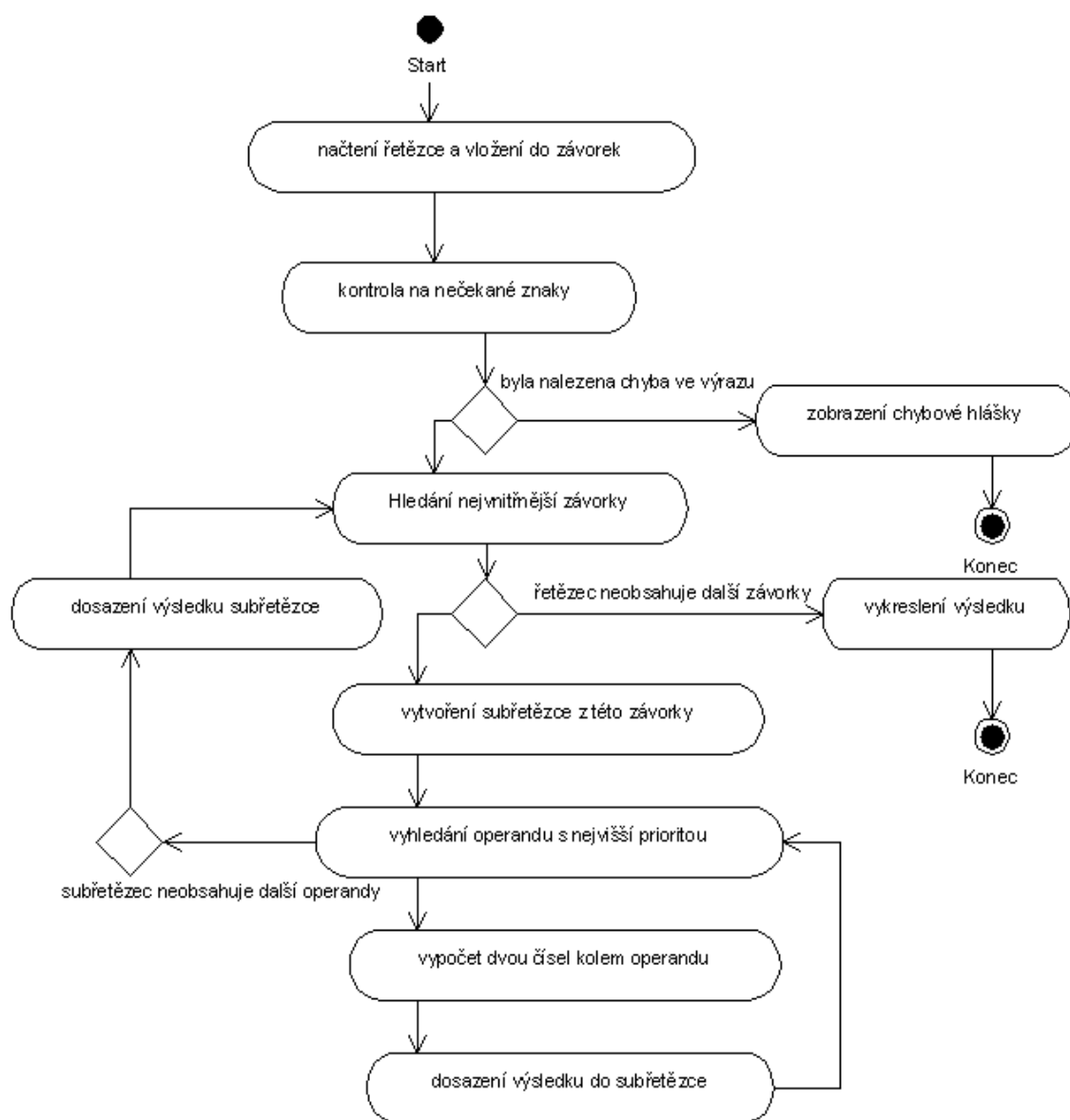
Licence: Free use only for non-commercial purpose.

Copyright: Nick Henson Copyright (c) 2002-2003

Autor: Nikolay Klimchuk (<http://henson.newmail.ru>)

Verze: 0.9

Algoritmus pro rozklad matematického vzorce, je uveden na obrázku (Obr. 18). Princip výpočtu je založen na postupném zjednodušení výrazu až do doby, kdy obsahuje pouze jedno číslo. Program po načtení výrazu, jej vnoří do vnějších závorek, což zjednodušuje kód. Poté provede kontrolu, jestli je výraz zapsán podle stanovených pravidel. Pokud ne, vyvolá chybové hlášení a vrátí se aby uživatel mohl chybu opravit. Pokud je výraz správně, nalezne první výskyt znaku „(“ a poté zpětně v řetězci najde „(,“ Tím získá obsah nejvíce vnořené závorky, a předá jí jako subřetězec další proceduře. Zde je tento krátký výraz dále analyzován způsobem nalezení matematické operace s nejvyšší prioritou. Po nalezení se předají dva parametry (subřetězec a pozice operandu) další proceduře. Ta si již vyextrahuje pravou a levou číselnou hodnotu kolem operandu (které si převede do typu Float) a operand. Poté již je vyhodnocen dílčí výsledek, který je do subřetězce dosazen místo hodnot z kterých byl vypočítán. A program je vrácen zpět na vyšší pozici. Tímto způsobem je vypočítán celý výraz. Po výpočtu poslední závorky (té do které program vložil celý výraz) program vytvoří nové okno a vloží do něj řetězec výrazu a výsledek oddělený znakem „=“.



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Obr. 18 Algoritmus pro rozklad mat. výrazu

ZÁVĚR

V této bakalářské práci jsem se snažil osvětlit jednotlivé kroky při vývoji aplikace pro mobilní telefon. Při návrhu je důležité si nejprve vybrat oblast zařízení pro níž má být aplikace určena. V závislosti na dostupných hardwarových prostředcích zvolit konfiguraci a profil, které J2ME specifikuje. Před samotným napsáním aplikace je dobré vyprojektovat si jak má daná aplikace vypadat a co má uživateli umožnit. Pro tento krok je dobré použít modelovací jazyk UML. Pro práci s tímto jazykem jsem používal program Poseidon for UML CE 3.0, který je freeware pro nekomerční účely. Pro samotný vývoj programu je nezbytné nainstalovat J2SDK a implementaci CLDC a MIDP. Pro snadnější vývoj jsou také k dispozici mnohé IDE prostředí. Já jsem používal balíček j2sdk-1_4_2_04-nb-3_6-bin-windows, který tyto součásti nainstaluje a obsahuje také vývojové prostředí NetBeans 3.6. Pro testování vytvořené aplikace je také potřeba si nainstalovat emulátor mobilního telefonu. Pro SonyEricsson jsem používal balíček semc_j2me_sdk_2_2_0_rc, který je s NetBeans propojen.

Výsledkem praktické části bakalářské práce je program BcKalk. Tato aplikace je podobná kalkulačce, ale umožňuje zadávat celý matematický výraz najednou a podle priority závorek a matematických operací vypočítá výsledek. Neboť J2ME nepodporuje proměnné typu Float a Double, musel jsem použít knihovnu Float, kterou napsal Nikolay Klimchuk. Ta obsahuje převod řetězce čísel na vlastní typ Float a poté s těmito hodnotami umožňuje provádět matematické operace.

Vývoj aplikací pro drobná a mobilní zařízení má v dnešní době velkou budoucnost. Uživatelé potřebují pracovat s informačními technologiemi v terénu a mobilní telefon je pro tento účel dobrou platformou. Jeho stále se vyvíjející výkon dovoluje programátorům vyvíjet stále složitější a propracovanější programy a díky mobilní síti třeba využívat data s firemního serveru. Velkou hnací silou je však zábava. V prostředí J2ME se dnes pro mobilní telefony vyvíjejí mnohé hry, které si uživatel může přímo z internetu za nízkou cenu stáhnout. Podle stále narůstajícímu počtu firem jež se J2ME zabývají lze vyvodit závěr, že tato platforma bude mít v blízké budoucnosti velké uplatnění.

SEZNAM POUŽITÉ LITERATURY

- [1] TOPLEY, Kim. J2ME v kostce : Pohotová referenční příručka. Redaktorka Irena Ranbusová; přeložil Pavel Makovec. 1. vyd. Praha : Grada Publishing, 2004. 536 s. ISBN 80-247-0426-9.
- [2] BITTNEROVÁ, Lucie Rút. J2ME v kostce [online]. Brno : ZONER software, 2002 ,14.10.2002[cit.2005-05-29].Dostupný z WWW: <<http://interval.cz/serial.asp?serial=45>>. ISSN 1212-8651.
- [3] ŠEDA, Jan . Co umožňuje Java v mobilních telefonech? [online]. MAFRA, 06.06.2002 [cit. 2005-05-29]. Dostupný z WWW: <http://mobil.idnes.cz/mob_prakticky.asp?r=mob_prakticky&c=A020605_5070886_mob_prakticky#top>.
- [4] Sun Microsystems. Java Technology [online]. c1994-2005 [cit. 2005-05-29]. Dostupný z WWW: <<http://java.sun.com/>>.
- [5] KAŠPAR, Jiří. NetBeans : Integrované vývojové prostředí [online]. Hewlett-Packard Development Company, c2004 [cit. 2005-05-29]. Dostupný z WWW: <<http://www.hp.cz/sw/ovms/pdf/netbeans.pdf>>.
- [6] SNG. Emulátor [online]. c2003 [cit. 2005-05-29]. Dostupný z WWW: <<http://www.sng.cz/cz/emulator.html>>.
- [7] Programy pro práci s UML diagramy [online]. [2004] [cit. 2005-05-29]. Dostupný z WWW: <<http://java.vse.cz/UMLNastroje.htm>>.
- [8] OKsystem. UML [online]. [2004] [cit. 2005-05-29]. Dostupný z WWW: <http://portal.oksystem.cz/portal/page?_pageid=53,42336&_dad=portal&_schema=PORTAL>.
- [9] KLIMCHUK, Nikolay. FloatTest [online]. c2002-2003 [cit. 2005-05-29]. Dostupný z WWW: <<http://henson.newmail.ru/j2me/Float.htm>>.
- [10] SNÁŠEL, Jaroslav. Jak jsou rychlé vaše mobily [online]. 27.7.2004 [cit. 2005-05-29]. Dostupný z WWW: <<http://www.mobilmania.cz/Profi/AR.asp?ARI=107806>>.

-
- [11] PEŠEK, Jiří. Jak rychlý je váš Sony Ericsson? [online]. 24.02.2005 [cit. 2005-05-29]. Dostupný z WWW: <http://semania.mobilmania.cz/index.php?option=com_content&task=view&id=773&Itemid=2>.
- [12] JBenchmark. Browse JBenchmark result database [online]. [2004] [cit. 2005-05-29]. Dostupný z WWW: <<http://www.jbenchmark.com/result.jsp>>.
- [13] HODEK, Václav. Programujeme v J2ME [online]. P.E.S. consulting, c2000-2004 [cit. 2005-05-29]. Dostupný z WWW: <<http://www.pcsvet.cz/art/article.php?id=4180>>.
- [14] STEIN, René. Návrh aplikací v jazyce UML [online]. Brno : Zoner software, 2003-2004 [cit. 2005-05-29]. Dostupný z WWW: <<http://interval.cz/serial.asp?serial=18>>. ISSN 1212-8651.
- [15] PITNER, Tomáš. Programování v jazyce Java [online]. Masarykova univerzita, Fakulta informatiky c2001-2003 [cit. 2005-05-29]. Dostupný z WWW: <<http://www.fi.muni.cz/~tomp/slides/pb162/printable.html#about>>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

J2ME	Java 2 Micro Edition	Java pro malé zařízení.
CLDC	Connected Limited Device Configuration	Konfigurace J2ME, určená pro mobilní telefony a PDA.
CDC	Connected Device Configuration	Konfigurace J2ME, určená pro výkonnější zařízení než CLDC.
MIDP	Mobile Information Device Profile	Profil konfigurace CLDC.
JVM, VM, KVM	Java Virtual Machine	Virtuální stroj Javy.
JAR	Java Archive	Balíček v němž je aplikace zabalená.
JAD	Java Application Descriptor	Soubor ve kterém jsou uloženy informace o midletu.

SEZNAM OBRÁZKŮ

Obr. 1 Konfigurace a profily platformy J2ME.....	14
Obr. 2 Softwarová architektura.....	18
Obr. 3 Druhy vstupů	19
Obr. 4 Testovací obrazce v programu JBenchmark.....	21
Obr. 5 Testovací obrazce v SPMark 04 3D Score	22
Obr. 6 Hlavní okno KToolbar soupravy Wireless Toolkit.....	27
Obr. 7 Prostředí NetBeans IDE 3.6.....	29
Obr. 8 Tvorba nového projektu v KToolbar	30
Obr. 9 Nastavení povinných atributů pro sadu midletů	30
Obr. 10 Výpis při kompilaci a spuštění midletu	31
Obr. 11 Spuštěný midlet v základním emulatoru J2MEWTK.....	32
Obr. 12 SNG emulátor	34
Obr. 13 Architektura JPDA v J2ME.....	35
Obr. 14 Pole matematický výraz.....	44
Obr. 15 Výsledek	44
Obr. 16 Informace.....	44
Obr. 17 Chybová hlášení.....	44
Obr. 18 Algoritmus pro rozklad mat. výrazu.....	46

SEZNAM TABULEK

Tab. 1 Výsledky testování SE T630 v programu FloatTest	22
Tab. 2 Výsledky testování SE T630 v programu JBenchmark 1.0	22
Tab. 3 Výsledky testování pro různé telefony	22
Tab. 4 Základní symboly v jazyce UML	25
Tab. 5 Seznam atributů souboru MANIFEST.MF a JAD	37
Tab. 6 Pravidla pro zadávání matematického výrazu	43

SEZNAM PŘÍLOH

P I CD-ROM – Aplikace BcKalk (obsahuje zdrojové kódy a soubory JAR a JAD)