



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

## VYUŽITÍ BEZDRÁTOVÝCH PŘENOSŮ V J2ME APLIKACÍCH

UTILIZATION OF WIRELESS TRANSFERS IN J2ME APPLICATIONS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

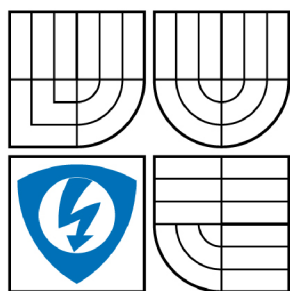
MILOŠ MIKULECKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN JEŘÁBEK

BRNO 2008



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor

Teleinformatika

**Student:** Mikulecký Miloš

**ID:** 78523

**Ročník:** 3

**Akademický rok:** 2007/2008

## NÁZEV TÉMATU:

**Využití bezdrátových přenosů v J2ME aplikacích**

## POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku programování J2ME aplikací a zaměřte se zejména na jejich využití pro bezdrátové přenosy (ovládání, komunikace či přenos souborů mezi zařízeními). Vytipujte vhodnou bezdrátovou technologii. Navrhněte a následně v programovacím jazyku J2ME realizujte aplikaci, která bude umožňovat bezdrátový přenos zpráv mezi mobilními telefony, případně i další funkce. Funkčnost aplikace otestujte na několika vybraných mobilních telefonech.

## DOPORUČENÁ LITERATURA:

[1] ROBERT, Morrow. Bluetooth operation and use. New York : McGraw-Hill, 2002. 215 s.

[2] MAHMOUD, Qusay. Naučte se Java 2 Micro Edition. Praha : Grada Publishing, 2002. 160 s

**Termín zadání:** 11.2.2008

**Termín odevzdání:** 4.6.2008

**Vedoucí práce:** Ing. Jan Jeřábek

**prof. Ing. Kamil Vrba, CSc.**

*předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

# LICENČNÍ SMLOUVA POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

## 1. Pan/paní

Jméno a příjmení: Miloš Mikulecký

Bytem: Cerekvice nad Loučnou

Narozen/a (datum a místo): 9.12.1985 v Litomyšli

(dále jen „autor“)

a

## 2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....

(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- disertační práce
- diplomová práce
- bakalářská práce
- jiná práce, jejíž druh je specifikován jako

.....

(dále jen VŠKP nebo dílo)

Název VŠKP: Využití bezdrátových přenosů v J2ME aplikacích

Vedoucí/ školitel VŠKP: Ing. Jan Jeřábek

Ústav: Telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v\*:

tištěné formě – počet exemplářů 1

elektronické formě – počet exemplářů 1

---

\* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: 2.6.2008

.....  
Nabyvatel

Miloš Mikulecký  
Autor

## **Anotace**

Tato bakalářská práce se zabývá problematikou programování J2ME aplikací se zaměřením na bezdrátové přenosy. Jako bezdrátová technologie byla vybrána technologie Bluetooth, která je dnes implementována na většině dnešních mobilních telefonech. Byla navržena a realizována aplikace umožňující přenos krátkých zpráv z jednoho telefonu na druhý přes rozhraní Bluetooth. Aplikace byla vyvíjena v programovacím prostředí NetBeans. Aplikace byla testována na několika mobilních telefonech různých značek a typů. Zde se ukázalo, že každý mobilní telefon je jiný, i když jde třeba o mobilní telefon jedné značky.

## **Klíčová slova**

J2ME, Bluetooth, aplikace, mobilní telefon, midlet

## **Abstract**

This Bachelor's thesis consider to problemation of programing J2ME aplication with specialized on wireless transfers. As the wireless technology was selected technology Bluetooth, what is implemented on the majority mobile phones at now. Aplication was projected and realized enabling transfers short messages between mobile phones over interface Bluetooth. The aplication was developed in programation enviroment NetBeans. The aplication was tested on some mobile phones different marks and types. It turned out, that some mobile phones is differnet, though mobile phones are of the same mark.

## **Keywords**

J2ME, Bluetooth, application, mobile phone, midlet

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Využití bezdrátových přenosů v J2ME aplikacích jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne 2.6.2008

Miloš Mikulecký  
(podpis autora)

## **PODĚKOVÁNÍ**

Děkuji vedoucímu bakalářské práce Ing. Janu Jeřábkovi, za cenné rady při zpracování bakalářské práce a všem známým, kteří byly ochotní zapůjčit svůj mobilní telefon k testování aplikace.

V Brně dne 2.6.2008

Miloš Mikulecký  
(podpis autora)

Bibliografická citace mé práce:

MIKULECKÝ, M. Využití bezdrátových přenosů v J2ME aplikacích. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 47 s. Vedoucí bakalářské práce Ing. Jan Jeřábek.



<b>1.</b>	<b>Úvod</b> .....	<b>12</b> -
<b>2.</b>	<b>Bluetooth</b> .....	<b>13</b> -
<b>2.1</b>	<b>Využití v praxi</b> .....	<b>13</b> -
<b>2.2</b>	<b>Parametry</b> .....	<b>13</b> -
<b>2.3</b>	<b>Technická specifikace</b> .....	<b>13</b> -
<b>2.4</b>	<b>Zásobník Bluetooth protokolu</b> .....	<b>14</b> -
2.4.1	Rádiová vrstva (Bluetooth radio) .....	14 -
2.4.2	Řadič základního pásma (Baseband).....	15 -
2.4.2.1	Fyzické linky .....	15 -
2.4.2.2	Logické linky .....	16 -
2.4.2.3	Adresování.....	16 -
2.4.2.4	Vytvoření spojení.....	17 -
2.4.3	Link Manager Protocol (LMP).....	17 -
2.4.4	Host Controller Interface (HCI) .....	18 -
2.4.5	Logical Link Control and Adaption protocol (L2CAP).....	18 -
2.4.6	Protokol RFCOMM.....	18 -
2.4.7	Service Discovery Protocol .....	18 -
<b>2.5</b>	<b>Bluetooth profily</b> .....	<b>18</b> -
<b>3.</b>	<b>Java 2 Micro edition</b> .....	<b>20</b> -
<b>3.1</b>	<b>Architektura J2ME</b> .....	<b>20</b> -
<b>3.2</b>	<b>Konfigurace</b> .....	<b>21</b> -
3.2.1	CDC.....	21 -
3.2.2	CLDC .....	22 -
<b>3.3</b>	<b>Virtuální stroje</b> .....	<b>22</b> -
<b>3.4</b>	<b>profily</b> .....	<b>22</b> -
3.4.1	MIDP.....	22 -
3.4.2	Profil PDA.....	23 -
3.4.3	Základní profily .....	23 -
3.4.4	RMI profil .....	23 -
<b>3.5</b>	<b>CLDC (Connected Limited device configuration)</b> .....	<b>23</b> -
3.5.1	Vlastnosti.....	24 -
3.5.2	KVM – Kilo Virtual Machine .....	25 -
<b>3.6</b>	<b>Midp 1.0 (mobil information device profil)</b> .....	<b>25</b> -

3.6.1	MIDlet .....	- 26 -
<b>3.7</b>	<b>MIDP 2.0 .....</b>	<b>- 27 -</b>
3.7.1	Novinky oproti MIDP 1.0 .....	- 27 -
<b>4.</b>	<b>Java for Bluetooth.....</b>	<b>- 28 -</b>
<b>4.1</b>	<b>Použití JABWT .....</b>	<b>- 28 -</b>
<b>4.2</b>	<b>Bluetooth protocol stack .....</b>	<b>- 29 -</b>
<b>4.3</b>	<b>profily J2ME a bluetooth .....</b>	<b>- 29 -</b>
<b>5.</b>	<b>Vytvořená aplikace .....</b>	<b>- 30 -</b>
<b>5.1</b>	<b>Základní Vlastnosti aplikace .....</b>	<b>- 30 -</b>
<b>5.2</b>	<b>Realizace v J2ME .....</b>	<b>- 32 -</b>
5.2.1	StartMidlet.....	- 33 -
5.2.1.1	Server.....	- 34 -
5.2.1.2	Klient .....	- 34 -
5.2.1.3	Nastavení .....	- 35 -
5.2.1.4	Info.....	- 35 -
5.2.1.5	Konec.....	- 35 -
5.2.1.6	Další funkce .....	- 35 -
5.2.2	ServerSluzby .....	- 35 -
5.2.3	Server .....	- 35 -
5.2.3.1	PoleAaO.....	- 36 -
5.2.3.2	poleO .....	- 36 -
5.2.3.3	polePrMSG .....	- 36 -
5.2.3.4	poleOdMSG.....	- 36 -
5.2.3.5	PoleZav .....	- 36 -
5.2.4	ServerSpojeni .....	- 37 -
5.2.4.1	close .....	- 37 -
5.2.4.2	frontaZprav .....	- 37 -
5.2.4.3	run.....	- 37 -
5.2.4.4	readFully .....	- 37 -
5.2.4.5	Odeslat .....	- 37 -
5.2.4.6	Writer.....	- 38 -
5.2.5	SpojeniServer .....	- 38 -
5.2.6	HledaniZarizeni .....	- 38 -
5.2.6.1	init.....	- 38 -
5.2.6.2	commandAction.....	- 38 -
5.2.6.3	servicesDiscovered .....	- 39 -

5.2.6.4	serviceSearchCompleted.....	- 39 -
5.2.6.5	run.....	- 39 -
5.2.6.6	hledaniSluzby.....	- 39 -
5.2.7	Klient.....	- 39 -
5.2.7.1	vytvoreniSpojeni.....	- 39 -
5.2.7.2	Další metody.....	- 39 -
5.2.8	KlientSpojeni.....	- 40 -
5.2.9	SpojěníServer a SpojeníKlient.....	- 40 -
5.2.10	Výsledek.....	- 40 -
<b>5.3</b>	<b>Testování aplikace.....</b>	<b>- 41 -</b>
<b>6.</b>	<b><i>Závěr.....</i></b>	<b>- 42 -</b>
<b>7.</b>	<b><i>Seznam použité literatury.....</i></b>	<b>- 43 -</b>
<b>8.</b>	<b><i>Zkratky.....</i></b>	<b>- 44 -</b>
<b>9.</b>	<b><i>Seznam Obrázků.....</i></b>	<b>- 45 -</b>
<b>10.</b>	<b><i>Seznam Tabulek.....</i></b>	<b>- 46 -</b>
<b>11.</b>	<b><i>Seznam Příloh.....</i></b>	<b>- 47 -</b>
<b>11.1</b>	<b>Obsah CD.....</b>	<b>- 47 -</b>

## 1. ÚVOD

Dříve jsme si mohli z mobilního telefonu pouze zavolat nebo poslat SMS. Rychlý rozvoj elektroniky a silná konkurence si však vyžádala implementaci velkého množství nových funkcí. Nejdříve jednoduché funkce jako budík, poznámkový blok, kalendář, stopky a další užitečné pomůcky. Dnes najdeme v mobilním telefonu např. MP3 přehrávač, digitální fotoaparát nebo dokonce GPS (satelitní navigace). Již samozřejmostí je i podpora Javy, což je programovací jazyk, který má tři edice: J2SE, J2EE a J2ME [9], která je určena pro malá zařízení, jako například mobilní telefony. Aby bylo možné spouštět na mobilním telefonu Java aplikace, je nutné aby telefon podporoval profil MIDP [1], [3]. Starší telefony mají podporu MIDP 1.0. Novější telefony již obsahují podporu MIDP 2.0 nebo už i MIDP 2.1. Oba profily jsou kompatibilní s MIDP 1.0.

Bezdrátové technologie, se kterými se dnes můžeme setkat u mobilních telefonů jsou např. GPRS, EDGE, HSCSD, IrDA, Bluetooth nebo Wi-Fi [6]. Díky těmto technologiím i ve spojení s Java aplikacemi je dnes možné proměnit mobilní telefon v mobilní zdroj informací či centrum zábavy.

## **2. BLUETOOTH**

Bluetooth [2], [13] je rádiová síť pro bezdrátovou komunikaci na krátkou vzdálenost. Název Bluetooth je odvozen od dánského krále Haralda zvaného „Modrý zub“, který před mnoha lety sjednotil vikingské kmeny. Také technologie Bluetooth měla za úkol vyřešit vzájemnou nekompatibilitu různých druhů zařízení. Dnes se rozšiřuje mnoho dalších bezdrátových radiových technologií, jejichž velikou výhodou je jejich všesměrovost a to, že nepotřebují ke svému fungování přímou viditelnost mezi komunikujícími stranami. Dosah technologie Bluetooth je teoreticky až 100 metrů při přímé viditelnosti. U mobilních telefonů bývá maximálně 10 metrů.

### **2.1 VYUŽITÍ V PRAXI**

Technologie Bluetooth se dnes používá hlavně tam, kde je výhodné nahradit kabel bezdrátovým spojením, hlavně u malých zařízení jako jsou třeba mobilní telefony, PDA, headsety, bezdrátové myši, bezdrátové klávesnice, tiskárny a další zařízení.

### **2.2 PARAMETRY**

Bluetooth je provozován v kmitočtovém pásmu 2,4 – 2,4835 GHz, což je ve většině zemí (včetně ČR) tzv. bezlicenční pásmo. V tomto pásmu se nacházejí i další bezdrátové systémy, proto je třeba zajistit aby se navzájem nerušili. Vysílací elektromagnetický výkon je asi 1 mW, což je zhruba 800x až 1200x větší [2] než vysílací výkon mobilního telefonu. Radiové vlny na těchto frekvencích projdou většinou materiálů, takže není problém spojit se se zařízením za zdi. Přenos může komplikovat nebo úplně znemožnit tzv. Faradayova klec, za kterou můžeme považovat i běžné železobetonové zdi. Spojení u technologie Bluetooth je plně duplexní, takže dovoluje přenos dat v obou směrech najednou. Bluetooth může pracovat jak v asymetrickém režimu s přenosovou rychlostí v jednom směru 723,3 kb/s a v druhém směru 56,7 kb/s tak i v symetrickém režimu s přenosovou rychlostí 433,9 kb/s v obou směrech.

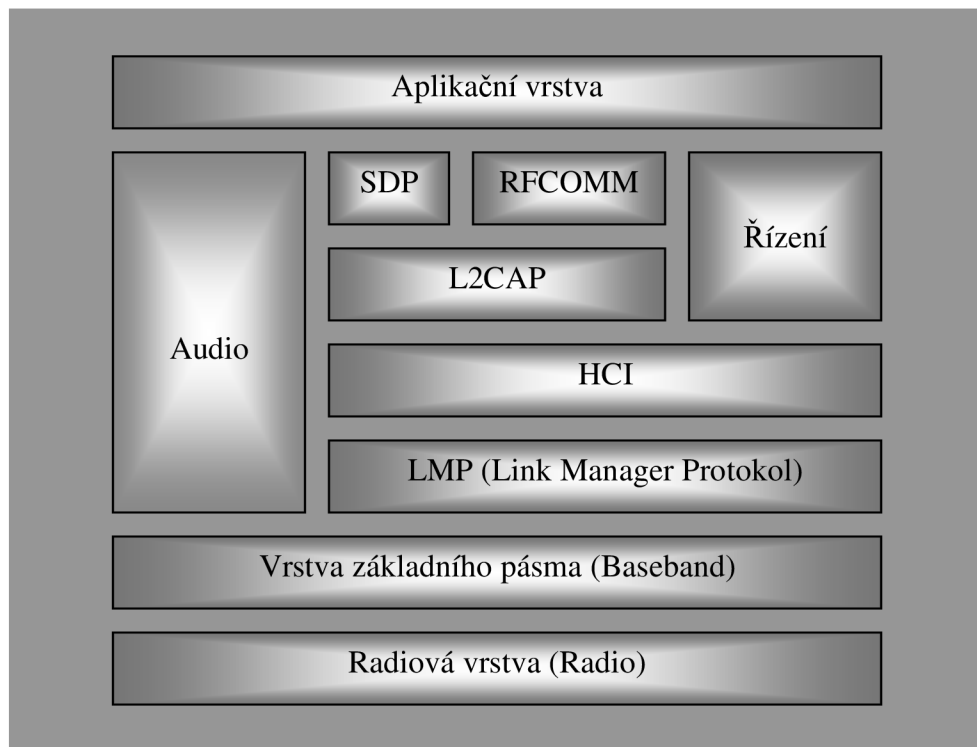
### **2.3 TECHNICKÁ SPECIFIKACE**

Bluetooth se skládá ze tří základních částí [2]:

- Bluetooth™ Radio - Zajišťuje rádiový přenos (vysílač, přijímač...)
- Bluetooth™ Link Controller - Linkový řadič (řízení navázání spojení, komunikace, identifikace a přístupu)
- Bluetooth™ Link Manager - Linkový manažer (připravuje data a zaručuje komunikaci se zařízením)

## 2.4 ZÁSObNÍK BLUETOOTH PROTOKOLU

Na obr. 1 je naznačen zásobník Bluetooth protokolu, v dalších podkapitolách jsou jednotlivé vrstvy rozebrány.



Obr.1. Zásobník Bluetooth protokolu

### 2.4.1 Rádiová vrstva (Bluetooth radio)

Rádiová vrstva definuje technickou specifikaci rádiového vysílání systému Bluetooth. Přesný frekvenční rozsah Bluetooth je od 2,402 GHz až do 2,480 GHz v tzv. bezlicenčním pásmu. Je určeno pro vědu, výzkum, průmysl a další. Činnost Bluetooth je v tomto frekvenčním pásmu definována způsobem frekvenčních skoků, tzv. frequency hopping, v 79 krocích po 1 MHz. Změna kmitočtu nastává 1600x za sekundu. Výhodou frekvenčních skoků je odolnost vůči rušení širokopásmového i úzkopásmového [8].

Bluetooth můžeme rozdělit do tří tříd podle výkonu (viz tab.1.). Výkon řídí LMP (Link Manager Protocol). Vysokofrekvenční signál je modulován GFSK (Gaussian Frequency Shift Keying) modulací.

Tab.1. Výkonové třídy Bluetooth

Výkonové třídy	Maximální výstupní výkon	Nominální výstupní výkon	Minimální výstupní výkon	Dosah
1	100 mW (20 dB)	-	1 mW (0 dB)	až 100 m
2	2,5 mW (4dB)	1 mW (0 dB)	0,25 mW (-6 dB)	až 50 m
3	1 mW (0 dB)	-	-	až 10 m

### 2.4.2 Řadič základního pásma (Baseband)

Řadič základního pásma definuje horní část fyzické vrstvy, která se stará o kódování, umožňuje synchronní nebo asynchronní mód, duplexní provoz, přeskokování kmitočtů a případně i základní kryptovací funkce. Baseband je zastoupen jako Link Controller (LC), který například vytváří linky nebo řídí spotřebu. Také řídí synchronní a asynchronní linky, přijímá pakety, vykonává jejich analýzu nebo hledá další Bluetooth zařízení v okolí.

Přenosový kanál se dělí do slotů, reprezentuje ho pseudonáhodná posloupnost frekvencí, které se používají pro vysílání v jednotlivých slotech. Délka časového slotu je 625  $\mu$ s, paket ale využívá jen část tohoto intervalu. Paket může být rozprostřen až do pěti slotů.

Dvě až osm zařízení může používat tentýž fyzický kanál a vytvořit tím pikosíť. V této pikosíti se jedno zařízení chová jako řídicí zařízení – *master* a další zařízení jako podřízená – *slave*. *Master* síť udržuje. V pikosíti je jen jedno zařízení typu *master*. Je však možné, aby tato funkce byla předávána mezi zařízeními v této síti.

#### 2.4.2.1 Fyzické linky

Baseband poskytuje dva druhy linek:

- SCO (Synchronous Connection-Oriented) linky
- ACL (Asynchronous Connection-Less) linky

SCO je synchronní point-to-point (bod-bod) spojení mezi zařízením *master* a zařízením typu *slave*. Je určené pro hlasovou komunikaci mezi Bluetooth zařízeními s rychlostí 64 kb/s. Mezi dvěma Bluetooth zařízeními mohou existovat až 3 spojení SCO, resp. jedno zařízení může komunikovat najednou se dvěma zařízeními pomocí SCO linek. SCO linky se používají hlavně pro přenos hlasu, tedy například mezi mobilním telefonem a Bluetooth headsetem.

ACL je asynchronní linka point-to-multipoint (bod–více bodů) spojení, která poskytuje paketově přepínané spojení mezi *master* a *slave* zařízením. Mezi *master* a *slave* zařízením může být pouze jedna ACL linka. Většinou se používá opakování přenosu chybně přenesených paketů.

V asynchronním módu se dosahuje přenosové rychlosti 721 kbps ve vysílacím směru a 57,6 kbps v přijímacím směru. V synchronním módu je to 432,5 kbps v obou směrech.

#### 2.4.2.2 Logické linky

Technologie bluetooth má 5 druhů logických linek:

- LC – Link Control – Kontrolní linka
- LM – Link Manager – Linkový manager
- UA – User Asynchronous data chanel – Uživatelský asynchronní datový kanál
- UI – User Isochronous data chanel - Uživatelský isochronní datový kanál
- 
- US – User Synchronous data chanel - Uživatelský synchronní datový kanál

LC a LM používají se v linkové vrstvě a ostatní se používají pro přenos asynchronních, isochronních a synchronních uživatelských informací.

#### 2.4.2.3 Adresování

Existují 4 druhy adres:

- BD\_ADDR – Bluetooth Device Address : 48-bitová fyzická adresa zařízení,
- AM\_ADDR – Aktive Member Address : 3-bitová adresa, která slouží k adresaci zařízení typu *slave* v rámci pokosítě. Platí jen po dobu existence pokosítě,



- PM\_ADDR – Parked Member Address : 8-bitová adresa, kterou využívá zařízení *master* k adresaci zařízení *slave*,
- AR\_ADDR – Access Request Address : Je používána zařízením *slave*.

#### 2.4.2.4 Vytvoření spojení

Než se vytvoří spojení, musí být zařízení ve stavu Standby, tedy ve stavu malé spotřeby energie a bez komunikace s ostatními zařízeními. Ale zařízení stále sleduje okolí, jestli se neobjeví nějaký signál iniciující vytvoření spojení. Když zachytí nějaký podnět pro spojení, přihlásí se dle určitého schématu jedno ze zařízení jako *master* a vyšle zprávu Page, pokud zná adresu BD\_ADDR druhého zařízení nebo zprávu Inquiry pro ověření adresy a pak následuje vyvolávací zpráva.

Na začátku této zprávy vysílá *master* 16 identických telegramů na 16-ti různých kmitočtech. Nepřijde-li odpověď, vyšle zařízení *master* zprávu na dalších 16-ti kmitočtech. Tato dotazovací zpráva se využívá pro detekci Bluetooth zařízení v dosahu.

Módy zařízení ve stavu spojeno:

- Aktive – Zařízení se aktivně zúčastňuje komunikace,
- Hold – Zařízení *slave* udržuje synchronizaci jen pomocí interních hodin. Tento stav se ale mění hned, jakmile je zapotřebí přenést další informaci,
- Shift – Zařízení *slave* se synchronizuje jen minimálně, což umožňuje úsporu energie,
- Parked – Zařízení *slave* je stále synchronizováno s pikosítí ale nezúčastňuje se komunikace.

#### 2.4.3 Link Manager Protocol (LMP)

Link Manager zajišťuje konfigurování linky, synchronizaci zařízení při vytváření spojení, autentizaci zařízení na základě privátních klíčů, šifrování spojení, detekci a korekci chyb a další. Vyhledává zařízení v okolí a komunikuje s nimi pomocí Link Manager Protokolu a vytváří spojení. Na vykonávání těchto funkcí používá Link Controller.

#### **2.4.4 Host Controller Interface (HCI)**

Host Controller Interface poskytuje programové rozhraní pro přístup k funkcím, které poskytuje hardwarová část Bluetooth. Prakticky se jedná o rozhraní, které spojuje software a hardware.

#### **2.4.5 Logical Link Control and Adaption protocol (L2CAP)**

Logical Link Control and Adaption Protocol je vrstva nad Baseband protokolem a slouží pro funkcionalitu data link layer. L2CAP poskytuje vyšším vrstvám datové služby bez spojení i se spojením. Zajišťuje funkce umožňující vytváření pikosítí. Specifikace L2CAP je definovaná jen pro ACL linky.

#### **2.4.6 Protokol RFCOMM**

Protokol RFCOMM je jednoduchý protokol, který poskytuje emulaci sériového portu RS232 pomocí L2CAP. RFCOMM podporuje až 60 simulovaných sériových linek mezi dvěma zařízeními. Počet současně běžících linek současně je specifikován pro každou implementaci různý.

#### **2.4.7 Service Discovery Protocol**

Service Discovery Protocol poskytuje aplikacím informace, jaké služby jsou dostupné a určuje charakteristiky služeb. SDP je nenáročný na kapacitu fyzického přenosového kanálu. Využívá request/response model komunikace, kde každá komunikace se skládá z jednoho požadavku a jedné odpovědi. Protokol obsahuje omezení, že může existovat jen jeden požadavek bez odpovídající odpovědi na jedno připojené zařízení.

### **2.5 BLUETOOTH PROFILY**

Bluetooth se skládá z několika protokolů, které jsou rozděleny do vrstev, čemuž se říká vrstevový model. Bluetooth profil je řez vrstvami protokolů a určuje, které protokoly budou použity. Profily tedy netvoří žádné nové funkce, ale určují, které protokoly se z celé specifikace Bluetooth využijí. Pro Bluetooth specifikaci 1.1 existuje 13 základních profilů:

- GAP (Generic Access Profile)
- SDAP (Service Discovery Application Profile)
- CTP (Cordless Telephony Profile)
- IP (Intercom Profile)

- SPP (Serial Port Profile)
- HP (Headset Profile)
- DNP (Dial-up Network Profile)
- FP (Fax Profile)
- LAP (LAN Access Profile)
- GOEP (Generic Object Exchange Profile)
- OPP (Object Push Profile)
- FPP (File Transfer Profile)
- SP (Synchronization Profile)

Existují i další, ale výše uvedené lze považovat za základní. Jednotlivé profily jsou nahrány již při výrobě nebo jsou nahrány pomocí driverů nebo firmwaru [2].

### 3. JAVA 2 MICRO EDITION

J2ME [1], [4] je jedna z edicí Javy od společnosti Sun Microsystems, která na rozdíl od J2SE (Java 2 Standard Edition) a J2EE (Java 2 Enterprise Edition), které jsou určeny především pro stolní počítače či servery, je určena především pro menší zařízení jako např. mobilní telefony, pagery, osobní digitální asistenti(PDA), navigační zařízení, přídatná zařízení a další malá zařízení.

Na vývoji J2ME se dnes podílejí i další firmy, jako jsou např. Palm, Nokia, Motorola a další. K vývoji se může připojit kdokoliv s připojením na internet. Tento projekt je znám pod jménem Java Community Process (JCP) [7].

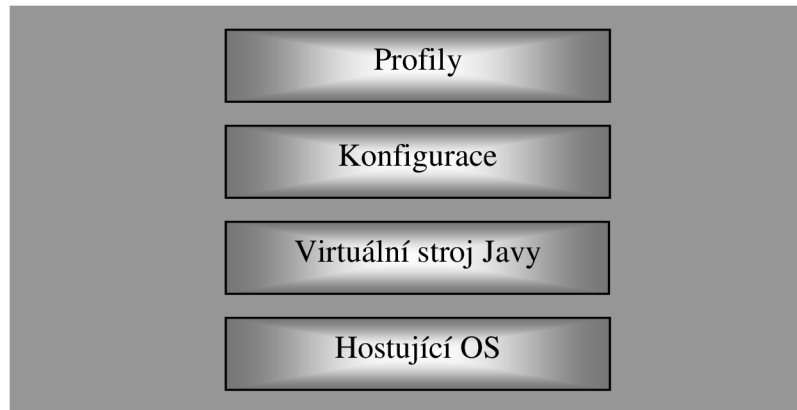
Výhodou Javy je přenositelnost kódu mezi zařízeními a objektově orientované programování. Musí se ale počítat s omezeními, protože jsou aplikace určeny pro malá zařízení. Malá zařízení mají omezené výpočetní vlastnosti, nízké rozlišení displejů, obtížné ovládání či malou operační paměť a další limity. Proto musela být J2ME přizpůsobena těmto vlastnostem a musela se omezit jen na některé balíčky ze standardní Javy. Navíc byly přidány nové balíčky, třídy a metody specifikované především pro malá zařízení.

#### 3.1 ARCHITEKTURA J2ME

J2ME definuje následující komponenty:

- Virtuální stroje Javy, kde se každý virtuální stroj Javy uplatňuje u jiného typu malého zařízení a s jinými nároky,
- API-skupina knihoven, které lze spustit na virtuálním stroji každého malého zařízení. Říká se jim konfigurace a profily,
- další nástroje pro vývoj a nastavení zařízení

API a virtuální stroj Javy tvoří pracovní prostředí J2ME. Na obr.2 je přehled pracovního prostředí. Centrem pracovního prostředí je virtuální stroj Javy, který běží na operačním systému daného zařízení. Nad ním je konfigurace, která se skládá z programovacích knihoven a na vrcholu stojí profily, které doplňují základní programovací knihovny o doplňkové programovací knihovny.



Obr.2 Architektura pracovního prostředí J2ME

## 3.2 KONFIGURACE

Malá zařízení (mobilní telefony, pagery, PDA a další) se liší svými funkcemi a vlastnostmi. Mají však některé společné rysy, jako podobné procesory nebo velikost paměti. Proto mohli autoři J2ME vytvořit konfigurace. Konfigurace se definuje jako horizontální členění produktů založené na dostupném množství paměti a výkonu procesoru jednotlivých zařízení. Jakmile má tyto informace k dispozici, konfigurace zjistí následující údaje:

- Podporované rysy programovacího jazyka Java,
- podporované rysy virtuálního stroje Javy,
- podporované základní javovské knihovny a API

V současné době existují dvě standardní konfigurace pro J2ME:CLDC (Connected Limited Device Configuration) a CDC (Connected Device Configuration) viz následující kapitoly.

### 3.2.1 CDC

Konfigurace CDC je určena pro výkonná zařízení, která bývají připojená k síti. Jsou to např. navigační systémy pro vozidla, přídatná zařízení, domácí spotřebiče a další. CDC obsahuje plnou verzi virtuálního stroje Javy podobnou J2SE.

Požadavky na zdroje pro CDC:

- Zařízení řídí 32-bitový procesor,
- paměť RAM a ROM má 2 MB a více pro Javu,
- zařízení vyžaduje plně funkční virtuální stroj Java 2 „Blue Book“,
- zařízení je nejčastěji připojeno bezdrátovým přerušovaným připojením s omezenou šířkou pásma, obvykle 9600 b/s nebo méně,

- zařízení mívá propracované uživatelské rozhraní, ale není podmínkou.

### **3.2.2 CLDC**

Tato konfigurace ukládá na zařízení mnohem menší požadavky než CDC. Používá se pro zabudovaná zařízení, dvousměrné pagery, PDA a další.

Požadavky pro J2ME CLDC:

- Paměť RAM a ROM může mít celkem 160 až 512 kB pro Javu,
- využívá jako zdroj napájení omezený zdroj napájení, např. baterie,
- zařízení je nejčastěji připojeno bezdrátovým přerušovaným připojením s omezenou šířkou pásma, obvykle 9600 bps nebo méně,
- zařízení mívá propracované uživatelské rozhraní, ale není podmínkou.

Z výše uvedeného je patrné, že hranice mezi těmito konfiguracemi není ostrá, hranice mezi těmito konfiguracemi se stále stírá, rozdíl udává především ve velikosti paměti přístroje, použitém napájení či nepřítomnosti uživatelského rozhraní.

## **3.3 VIRTUÁLNÍ STROJE**

Virtuální stroj je abstraktní počítač, který obsahuje runtime system (realizuje spojení s hardwarem) a interpretem (vykonává binární kód).

CLDC a CDC stanoví své vlastní skupiny podporovaných rysů virtuálního stroje Javy. Virtuální stroj CLDC je menší než virtuální stroj CDC, protože podporuje mnohem méně rysů. Virtuální stroj pro CLDC se nazývá KVM (Kilo Virtual Machina) a pro CDC je to CVM.

## **3.4 PROFILY**

Je to sada programových rozhraní (API) a tvoří nadstavbu konfigurace. Umožňuje přístup k vlastnostem specifickým pro dané zařízení. Viz obr.3.

### **3.4.1 MIDP**

Je navržen pro CLDC. Je určen především pro mobilní telefony nebo obousměrné pagery. Umožňuje nahrávat do koncového zařízení nové aplikace. Aplikacím běžícím pod MIDP se říká Midlety. MIDP existuje v několika verzích, od MIDP 1.0 přes MIDP 2.0 po nejnovější MIDP 2.1.

### 3.4.2 Profil PDA

Také založen na CLDC s API pro uživatelské rozhraní a API pro ukládání dat.

### 3.4.3 Základní profily

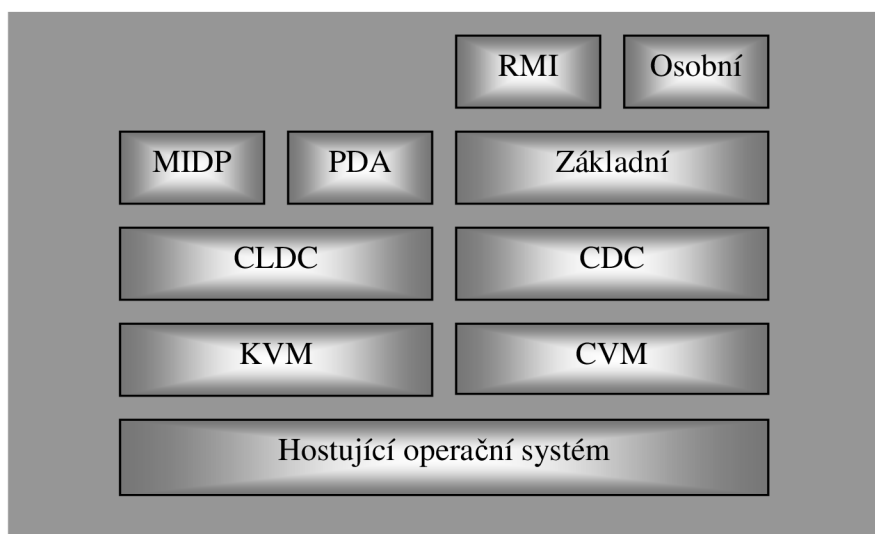
Základní profily jsou založeny na CDC, rozšiřují jeho programové vybavení, ale nedodávají API pro uživatelské rozhraní. Má sloužit pro další profily, např. Osobní profil, který rozšiřuje jeho možnosti o grafické uživatelské rozhraní (GUI).

### 3.4.4 RMI profil

RMI (Remote Method Invocation) [11] umožňuje z jednoho virtuálního stroje vyvolávat metody na jiném virtuálním stroji. Stará se o tři věci:

- Dokáže lokalizovat vzdálený objekt,
- Komunikuje se vzdálenými objekty,
- Dokáže přenést mezi objekty jiný objekt v podobě bajtového kódu.

RMI profil rozšiřuje Základní profil o RMI pro daná zařízení. Je určen pro CDC/Základní profil a ne pro CLDC/MIDP.



Obr.3 Prostředí J2ME

## 3.5 CLDC (CONNECTED LIMITED DEVICE CONFIGURATION)

CLDC definuje standardní platformu Javy s minimálními nároky na malá zařízení s omezenými zdroji. Rysy specifické pro trh s určitými produkty jako mobilní telefony nebo pagery však nejsou součástí CLDC, ale jsou definovány v profilech, které jsou CLDC nadřazeny. Konfigurace se zaměřuje hlavně na

podobné prvky různých zařízení jako je například množství paměti nebo podobně výkonný procesor. V CLDC neexistují žádné volitelné rysy. Základním cílem CLDC je zajištění mobility a interoperability mezi aplikacemi na různých zařízeních s omezenými zdroji.

### 3.5.1 Vlastnosti

Podle specifikace mají zařízení s CLDC tyto vlastnosti:

- 160 kB až 512 kB celkové paměti – Zařízení pracující na CLDC by mělo mít minimálně 128 kB stálé paměti pro virtuální stroj Javy a pro CLDC knihovny a minimálně 32 kB dočasné paměti pro práci virtuálního stroje, vše nezávislé na chodu ostatních aplikací,
- 16-bitový nebo 32-bitový procesor s minimální taktovací frekvencí 25 MHz. Jsou to typické procesory pro malá zařízení,
- připojitelnost k některému typu sítě – často se jedná o dvousměrné bezdrátové připojení s omezenou šířkou pásma,
- Nízká spotřeba energie – Zařízení s CLDC často používají jako zdroj energie baterii.

Tomuto popisu vyhovují mobilní telefony, pagery, čtečky čárových kódů, audio a video zařízení, navigační systémy a další. Postupně s přibývajícými zařízeními a jejich nároky se i specifikace CLDC bude měnit.

CLDC poskytuje zařízením tyto funkce:

- Soubor rysů Javy a virtuálního stroje Javy,
- soubor základních knihoven Javy,
- základní vstup/výstup,
- základní podporu sítí,
- zabezpečení.

CLDC však nezasahuje do běhu aplikací, do uživatelského rozhraní, ani do správy událostí nebo do interakce mezi uživatelem a aplikací. To má na starosti profil, jako je MIDP.

Oproti J2SE chybí virtuálnímu stroji Javy v J2ME tyto části:

- Podpora plovoucí řádové čárky,
- finalizace,
- možnosti zpracování chyb,



- JNI (Java Native Interface),
- uživatelem definované zavaděče tříd,
- Podpora reflexe,
- skupiny vláken nebo podprocesy typu démon,
- slabé odkazy.

Tyto rysy byly odstraněny kvůli jejich nadměrným požadavkům jak na paměť tak na procesor. Během programování bychom si měli na tyto věci dát pozor.

### **3.5.2 KVM – Kilo Virtual Machine**

KVM je kompletní javovské pracovní prostředí pro malá zařízení. Jedná se o opravdový virtuální stroj Javy, ale s malými odchylkami pro přizpůsobení se malým zařízením. Zpráva o J2ME, kterou můžete nalézt na [12], popisuje KVM takto:

- Vytvořené pro 16-bitové a 32-bitové CISC nebo RISC procesory a procesory s minimální rychlostí 25 MHz,
- nároky na statickou paměť od 50 do 80 kB,
- vysoce mobilní s ustálenými rozměry a přizpůsobivé,
- za daných podmínek co možná nejrychlejší a kompletní bez nutnosti obětovat jiné, výše zmíněné vlastnosti designu.

### **3.6 MIDP 1.0 (MOBIL INFORMATION DEVICE PROFIL)**

MIDP je vrcholem CLDC a definuje otevřené prostředí pro vývoj aplikací, kterým dala společnost Sun název mobilní informační zařízení (Mobile Information Device – MID). MIDP je J2ME profil určený pro bezdrátová zařízení jako je třeba mobilní telefon. I MIDP se stále vyvíjí dle potřeb společností a programátorské veřejnosti.

Vlastnosti MIDP:

- Displej – Minimální velikost obrazovky 96 x 54 pixelů a hloubka barev 1 bit
- Vstupní zařízení – Klávesnice nebo dotyková obrazovka
- Paměť – 32 kB nestálé paměti pro práci Javy, 128 kB stálé paměti a 8 kB paměti pro ukládání dat z aplikací
- Práce v síti – Přerušované obousměrné spojení, obvykle bezdrátové s omezenou šířkou pásma

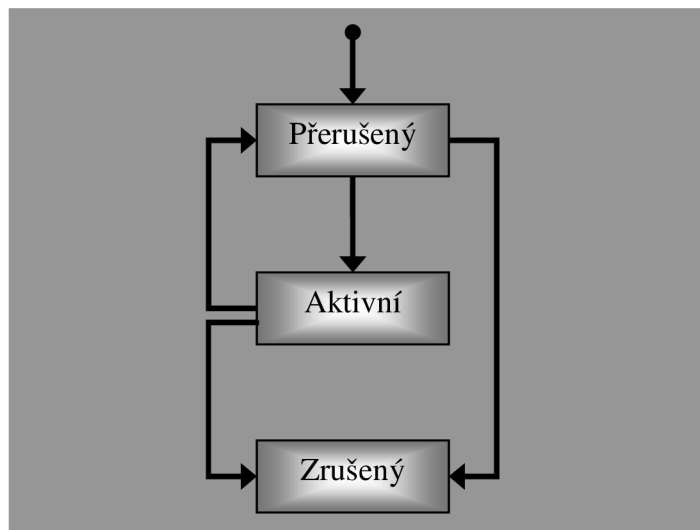
Neřešené oblasti v CLDC:

- Správa průběhu aplikací – MIDP obsahuje balíček *javax.microedition.midlet* pro správu průběhu aplikace
- Uživatelské rozhraní a události – MIDP poskytuje balíček *javax.microedition.lcdui*, které slouží k tvorbě uživatelského rozhraní
- Ukládání dat v zařízení – MIDP také podporuje balíček *javax.microedition.rms*, který zvládá databázový systém
- Připojitelnost k síti – MIDP podporuje *ContentConnection* a rozšiřuje ho

### 3.6.1 MIDlet

MIDlet je aplikace běžící na MIDP zařízení. MIDlety se tvoří jako jedna nebo několik tříd Javy, jejichž objekty jsou komprimovány do JAR souboru. MIDlet se může nacházet v těchto stavech (obr.4):

- Přerušovaný
- Aktivní
- Zrušený



Obr.4. Pravidla přechodu mezi jednotlivými stavy

Při prvním spuštění se MIDlet nachází ve stavu Přerušovaný, když je připraven, převede ho kontrolní program do stavu Aktivní. V tuto chvíli může uživatel s aplikací pracovat. Aplikaci přeruší znova buď sám program nebo MIDP. MIDlet lze také z jakéhokoli stavu převést do stavu Zrušený, opět to udělá buď program

nebo MIDP. Ve stavu Zrušený by MIDlet měl uvolnit systému MIDP všechny zdroje, které v tu chvíli používá.

### **3.7 MIDP 2.0**

Uživatelé chtějí do svých mobilních zařízení další aplikace dle svých potřeb a vkusu a nestačí jim pouze předdefinované aplikace výrobcem. Specifikace MIDP 1.0 tedy nestačila programátorům pro vývoj nových aplikací.

V listopadu 2002 byla dokončena specifikace MIDP 2.0 [4]. Tato specifikace je především určena pro mobilní telefony. Na tvorbě specifikace se podílelo i několik výrobců mobilních telefonů.

Specifikace MIDP 2.0 obsahuje téměř dvakrát tolik tříd co specifikace MIDP 1.0, tudíž se zvýšili i nároky na mobilní zařízení.

- 256kB RAM navíc nad požadavky CLDC
- 128kB pro běh JRE

#### **3.7.1 Novinky oproti MIDP 1.0**

Přibylo herní rozhraní *javax.microedition.lcdui.game*, protože J2ME je hojně využívána právě pro psaní her. Dále ještě např. možnost připojení přes zabezpečený HTTPS protokol. Mnoho dalších rozdílů je možné nalézt v [4].

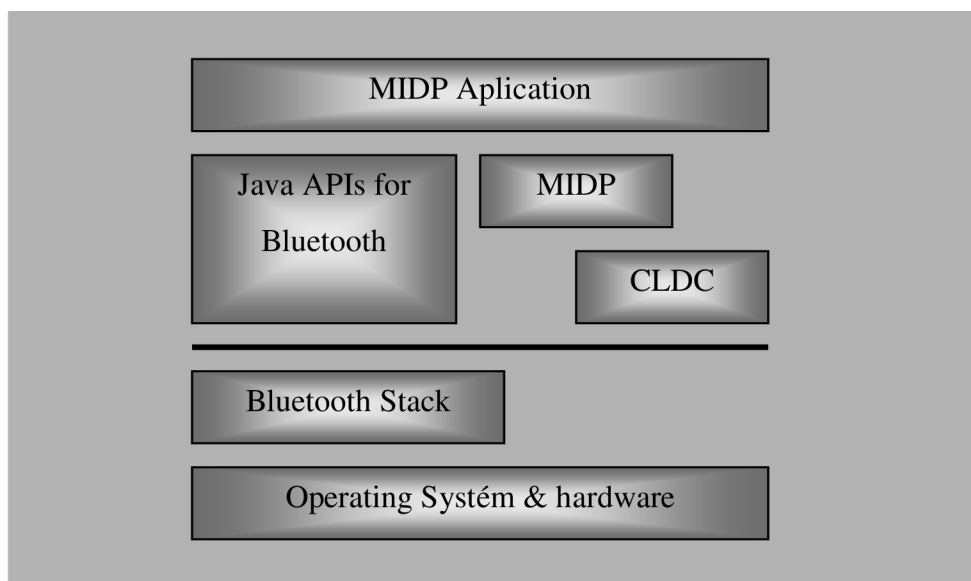
Pro nás je důležité, že specifikace MIDP 2.0 obsahuje potřebný balíček Bluetooth *javax.bluetooth*.

## 4. JAVA FOR BLUETOOTH

Aplikační rozhraní jazyka Java (APIs) [3] jsou definována a spravována skupinou Java Community Process (JCP). Informace o JCP lze nalézt na jejich stránkách [7]. Každé nově vzniklé rozhraní se nazývá Java Specification Request (JSR), což je tzv. požadavek na změnu, který se používá při každém požadavku nebo doplnění dané platformy. Tento požadavek schvaluje výborem JCP, který pracuje na dané specifikaci.

Třídy Java, které podporují technologii Bluetooth, se nazývají JABWT (Java APIs for Bluetooth Wireless Technology – aplikační rozhraní Javy pro Bluetooth). Java Specification Request pro toto aplikační rozhraní má označení JSR-82.

Z obr.5 je vidět, že JABWT je odděleno od MIDP a CLDC, tedy že k ovládání má přístup pouze JABWT, ale MIDP k nim může přistupovat. U JABWT se jedná o speciální Java knihovny, tudíž je nutné, aby tyto knihovny byly podporovány mobilním telefonem.



Obr.5 Schéma přístupu MIDP k Bluetooth

### 4.1 POUŽITÍ JABWT

U většiny Bluetooth zařízení spočívá běh tohoto rozhraní v následujících krocích:

- Inicializace Bluetooth stacku – nastavení Bluetooth parametrů zařízení, spuštění

- Hledání dalších Bluetooth zařízení v dosahu.
- Hledání služeb dostupných na těchto zařízeních.
- Navázání spojení se zařízením s vyhovujícími službami.
- Komunikace, přenos dat.
- Uzavření spojení.

## **4.2 BLUETOOTH PROTOCOL STACK**

Je to software nebo skupina mikroprogramů, která má přímý přístup k Bluetooth zařízení. Má kontrolu nad přístrojovým nastavením, komunikačními parametry a výkonem Bluetooth zařízení. Stack se skládá z vrstev [3], kde každá má svůj specifický úkol v celkové funkčnosti Bluetooth zařízení. Všechny vrstvy stacku nemusí být nezbytně nutně použity.

## **4.3 PROFILY J2ME A BLUETOOTH**

Je třeba od sebe rozlišit pojmy Bluetooth profily a J2ME profily. J2ME profily jsou sada Javovských tříd, které rozšiřují funkčnost konfigurace J2ME, jako například MIDP rozšiřuje CLDC konfiguraci. Na druhé straně Bluetooth profil může být implementován do jiného programovacího jazyka, protože se odkazuje na funkce Bluetooth zařízení.

## 5. VYTVOŘENÁ APLIKACE

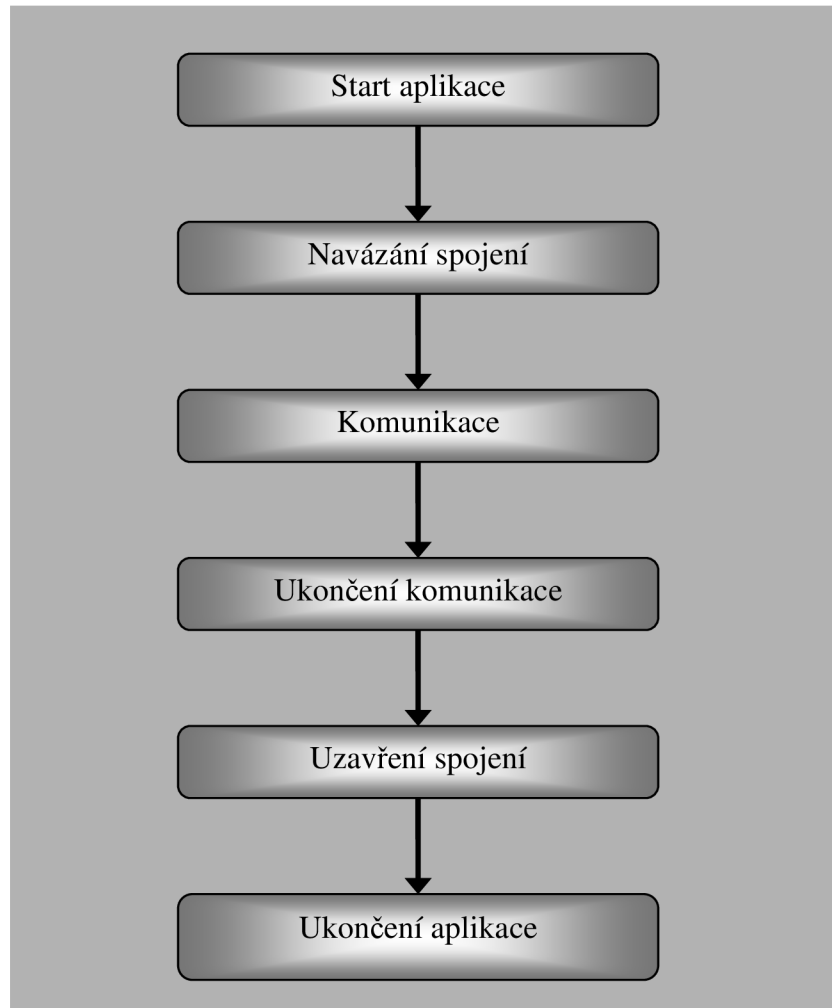
Následující kapitoly rozebírají naprogramovanou aplikaci, její chod, funkce a popisují její životní cyklus. Dle zadání bakalářské práce půjde o aplikaci, která umožní posílat zprávy přes Bluetooth mobilními zařízeními.

### 5.1 ZÁKLADNÍ VLASTNOSTI APLIKACE

Nejdůležitější funkcí této aplikace je vytvoření spojení, jeho udržení, následné posílání zpráv přes toto spojení a ukončení spojení. Měla by tedy vykonávat činnosti v následujícím pořadí:

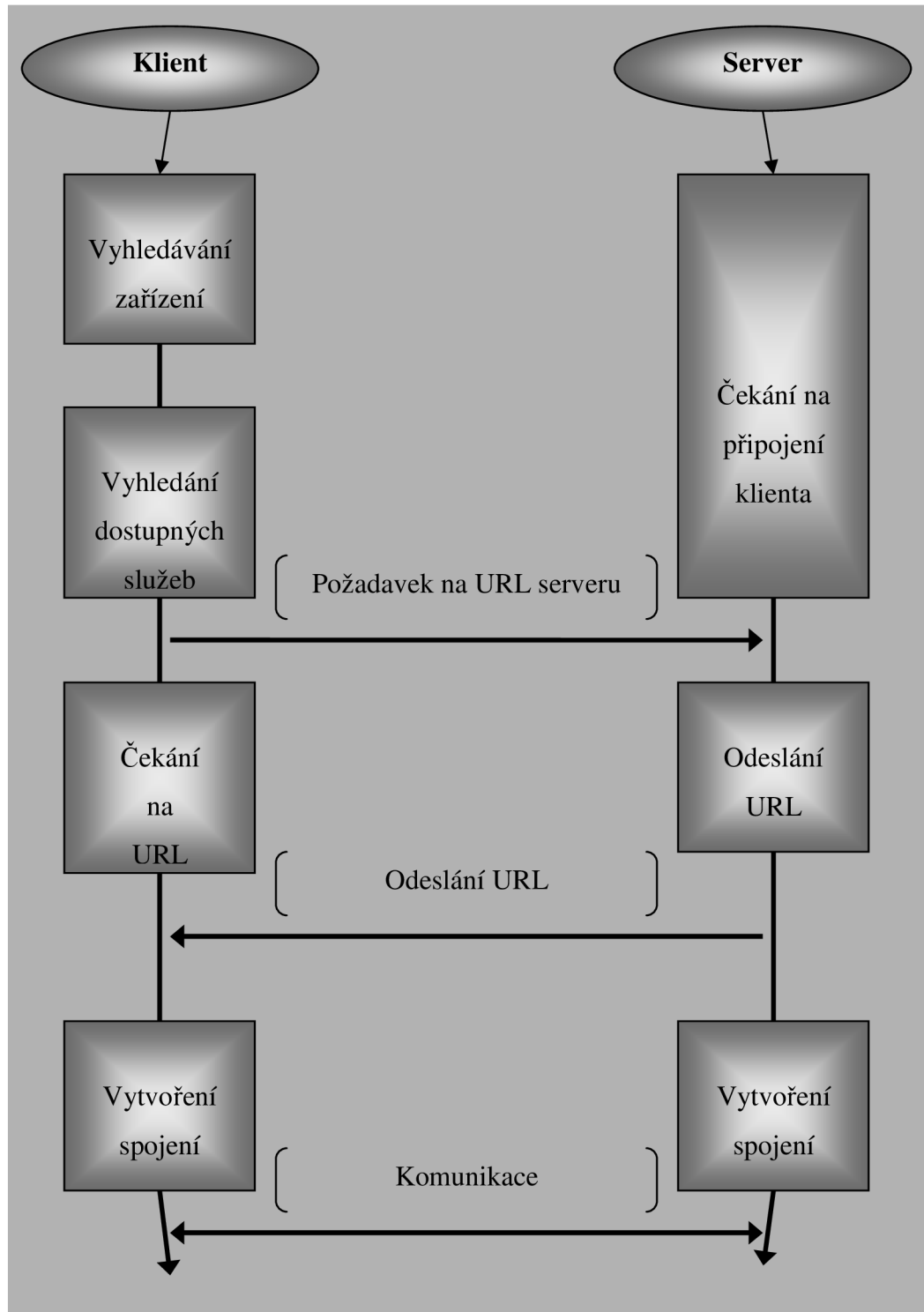
- Spuštění aplikace,
- navázání spojení s druhým zařízením,
- vlastní komunikace spočívající v přenosu textových zpráv přes Bluetooth,
- uzavření spojení
- ukončení aplikace.

Při spuštění aplikace se uživateli zobrazí uživatelské rozhraní, funkce aplikace atp. Uživatel si zvolí, co bude provádět s aplikací, rozhodne se, zda chce vytvořit server (*master*) nebo jestli se bude chtít připojit už k vytvořenému serveru jako klient (*slave*). Poté se klientské zařízení připojí k serveru. Následně proběhne komunikace mezi oběma zařízeními. Po vzájemné komunikaci se spojení ukončí. V případě odpojení klientského zařízení server stále běží a pikosít' je stále vytvořena. Proto zařízení na straně serveru buď setrvá ve svém stavu a vyčkává na připojení dalšího zařízení, nebo svoji činnost ukončí. Životní cyklus aplikace je znázorněn na Obr.6.



Obr.6 Životní cyklus aplikace

Při vlastním spojování je využita architektura klient-server (*master-slave*), jak již bylo naznačeno výše. Takže jedno zařízení musí vytvořit server a čekat na připojení dalších zařízení. Klientské zařízení musí nejprve vyhledat dostupná zařízení v dosahu a zjistit, zdali na nich je spuštěna požadovaná služba. Jestliže najde zařízení, kde je hledaná služba, vyšle klientské zařízení požadavek na získání URL a poté dojde k připojení na tuto URL a sestavení spojení (viz. Obr. 7)



Obr.7 Sestavení spojení

## 5.2 REALIZACE V J2ME

Zdrojový kód vytvořené aplikace je přiložen jako příloha na přiloženém CD, kde se nachází celý projekt aplikace pro prostředí NetBeans včetně již zkompilevané aplikace.



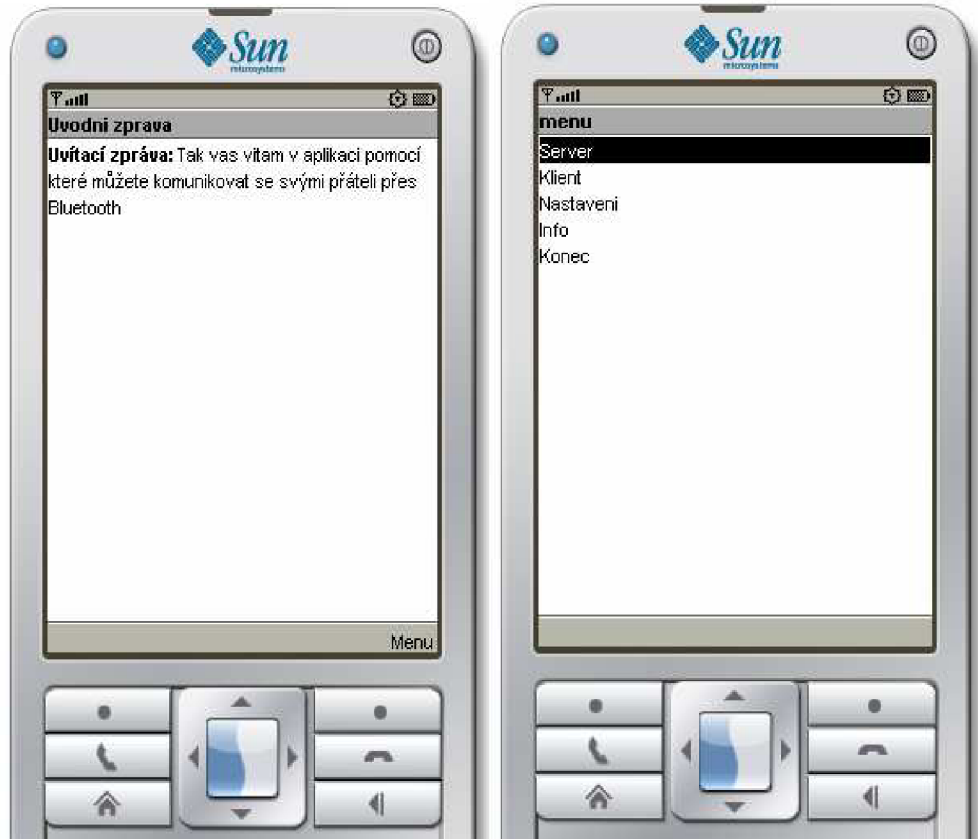
Při realizaci aplikace bylo použito hierarchie klient – server (*master – slave*) tak, že se uživatelé budou muset domluvit, který z nich vytvoří server a který z nich se připojí. Řešení bylo zvoleno tak, aby bylo zřejmé, co na které straně spojení probíhá. Aplikace je tvořena jedním balíčkem (package) *prvni*. V balíčku *prvni* najdeme 6 tříd, 2 rozhraní a 1 spustitelný midlet.

### 5.2.1 StartMidlet

StartMidlet je midlet, který je spuštěn na začátku běhu aplikace. Tento midlet ovládá celou aplikaci. Pomocí ostatních tříd řídí životní cyklus aplikace. Obsahuje výchozí bezparametrický konstruktor, který tvoří kostru midletu. Kostra midletu se skládá z:

- *protected void startApp()*
- *protected void pauseApp()*
- *protected void destroyApp(boolean unconditional)*

V daném okamžiku se midlet nachází v jednom ze stavů: Paused, Active nebo Destroyed. Při zavedení se midlet nachází ve stavu Paused, kdy dochází k inicializaci tříd a instancí. Poté je jeho stav změněn z Paused na Active a je volána metoda *startApp()*. Po zavolání metody *startApp()* se na displeji zobrazí obrazovka s úvodní zprávou pro uživatele a s jedním tlačítkem pro menu viz obr.8a.



Obr.8a a 8b Uvítací zpráva, menu

Po stisknutí tlačítka menu se dostaneme do menu aplikace, kde si uživatel může vybrat, co chce provést za akci. Menu je vidět na obr.8b.

V menu máme na výběr z pěti položek:

- Server
- Klient
- Nastavení
- Info
- Konec

#### 5.2.1.1 Server

Pod položkou Server se nachází spuštění serverové části aplikace. Po zvolení této položky se dostaneme do serverového formuláře, na kterém jsou položky potřebné pro posílání zpráv.

#### 5.2.1.2 Klient

Pod položkou Klient se nachází spuštění klientské části aplikace. Je podobná jako položka Server jen s malým rozdílem ve funkcích.

### 5.2.1.3 Nastavení

Pod položkou nastavení se mělo nacházet nastavení zvuku a vibrací. Položky pro výběr tu jsou, ale nejsou funkční. Tyto položky jsou zatím ve stádiu vývoje.

### 5.2.1.4 Info

Pod touto položkou můžeme najít nějaké informace o aplikaci, autorovi apod.

### 5.2.1.5 Konec

Výběr této položky menu ukončí korektně běh aplikace a zavře ji.

### 5.2.1.6 Další funkce

Dále StartMidlet obsahuje UUID (Universally Unique Identifier), což je číslo o maximální délce 32 hexadecimálních čísel, které jednoznačně označuje aplikaci, tak aby ji mohl mobilní telefon, na kterém je spuštěn Klient, nalézt.

Další důležitou částí je *LocalDevice.getLocalDevice().getDiscoverable()*; která je potřebná pro inicializaci Bluetooth modulu v mobilním telefonu. Na některých mobilních telefonech bez této části kódu aplikaci ani nelze spustit.

StartMidlet obsahuje rozhraní *commandAction*, které obsahuje nastavení pohybu v menu a výběr jednotlivých položek.

## 5.2.2 ServerSluzby

V této třídě se nacházejí parametry a nastavení služeb a spojení pro vytvoření serverové části aplikace.

Obsahuje vlákno, které vytvoří server, pokud se nevyskytne nějaká výjimka. Vlákno se vytvoří pomocí *Thread thread = new Thread(this)*; Vlákno se spustí pomocí *thread.start()*; a tělo vlákna je v *public void run()*. Konstruktor *connectionNotifier = (StreamConnectionNotifier) Connector.open(url)*; otevře spojení s URL, která obsahuje UUID. *StreamConnection connection = (StreamConnection) connectionNotifier.acceptAndOpen()*; vytvoří server a informuje příslušné rozhraní.

## 5.2.3 Server

Tato třída zobrazuje formulář se serverovými položkami: Počet odeslaných a přijatých zpráv, textové pole pro zprávy k odeslání, status spojení a textové pole pro přijaté zprávy. Textové pole pro zprávy k odeslání je omezeno 250 znaky a zadávat

můžeme jakékoli znaky. Pro přijaté zprávy je také použito textové pole. Není to ideální řešení, ale postačuje požadavkům. Textové pole pro přijaté zprávy je omezeno 100000 znaky a obsahovat může libovolné znaky. V případě, že by uživatelé komunikovali tak, že by textové pole přijatých zpráv zaplnili 100000 znaky, mohou je smazat. Jednotlivé zprávy jsou odděleny ---*Dalsi zprava:*, aby bylo poznat, kde začíná jaká zpráva. Každá nová zpráva se zobrazuje vždy na začátek textového pole.

Server obsahuje také rozhraní *commandAction*, které spravuje příkazy. Obsahuje dva příkazy, Odeslat a Konec. Příkaz Odeslat slouží k odeslání zprávy. Vezme data v textovém poli pro zprávy k odeslání, přidá k nim id zprávy, uloží do *Pole* z třídy *Vector* a odešle je metodě *frontaZprav* třídy *ServerSpojeni*. Příkaz Konec po vybrání ukončí běh aplikace.

Dále obsahuje 5 metod viz následující podkapitoly:

#### **5.2.3.1 PoleAaO**

Tato metoda slouží pro připojení druhého zařízení k serveru.

#### **5.2.3.2 poleO**

Tato metoda je volána po vytvoření spojení. Nastaví status na *Spojeno*.

#### **5.2.3.3 polePrMSG**

Tato metoda zobrazuje přijaté zprávy do textového pole s přijatými zprávami. Dále inkrementuje proměnnou *PocPrijatych* a nastavuje počet přijatých a odeslaných zpráv celkem v položce formuláře *Pocet zprav*.

#### **5.2.3.4 poleOdMSG**

Tato metoda inkrementuje proměnnou *PocOdeslanych* a nastavuje počet přijatých a odeslaných zpráv celkem v položce formuláře *Pocet zprav*.

#### **5.2.3.5 PoleZav**

Tato metoda se volá při uzavření spojení. Informuje tím uživatele pomocí nastavení položky formuláře *Status* hláškou *Spojeni uzavreno*.

#### 5.2.4 ServerSpojeni

Tato třída má za úkol obstarávat zpracování zpráv a jejich příjem a odeslání. Jak přijatých tak odeslaných. Obsahuje tyto metody:

##### 5.2.4.1 close

Tato metoda se volá při ukončení aplikace. Její funkcí je zrušit všechna navázaná spojení, jak příchozích tak odchozích.

##### 5.2.4.2 frontaZprav

Metoda nejdříve zkontroluje délku zprávy, zda-li není příliš dlouhá. Pokud ano vyvolá výjimku *IllegalArgumentException*. Po úspěšné kontrole délky zprávy přidáme do kolekce *Hashtabele Zprava* číslo zprávy a data zprávy, která nám přišla po stisknutí tlačítka *Odeslat*.

##### 5.2.4.3 run

Je to metoda, která tvoří tělo vlákna. Nejprve vytvoří odchozí a příchozí stream a potom vytvoří další vlákno. Toto vlákno slouží k odeslání dat.

Další část metody se stará o příjem zpráv ze streamu. Jedná se o přečtení dat ze streamu, porovnání jejich délky pro zjištění, zda-li není spojení ukončeno, a odeslání příchozích dat metodě *polePrMSG* třídy *Server* s parametry příchozích dat. V případě, že je spojení uzavřeno, se volá metoda *PoleZav* třídy *Server*.

##### 5.2.4.4 readFully

Tato metoda přečte data ze streamu a převede je na *integer count*. Velikost proměnné porovná, zda se nerovná -1. Pokud ano tak se vyvolá výjimka *IOException*.

##### 5.2.4.5 Odeslat

Tato metoda odesílá zprávy druhému mobilnímu telefonu. Nejdříve metoda zjistí, zda-li není zpráva příliš dlouhá. Dále nastaví velikost buferu pro odeslaná data. Zkopíruje data zprávy od indexu 0 do buferu od indexu 6, a to na tolik indexu, jak je dlouhá zpráva. Následně metoda zapíše data z buferu do odchozího streamu.

#### 5.2.4.6 Writer

Jedna se o třídu, která má za úkol posílat zprávy. Tvoří ji vlákno, v jehož těle se přistoupí ke kolekci *Hashtable Zprava*, ve které nalezneme jednotlivé elementy zprávy a přiřadíme jim *id*. Potom se odešle takto připravená zpráva metodě *Odeslat*, vymaže se daný element a odešlou se data metodě *poleOdMSG* třídy *Server*, která počítá odeslané zprávy.

#### 5.2.5 SpojeniServer

Jedná se o rozhraní, které slouží k přijímání a informování příslušných metod ve třídě *Server* a k posílání parametrů.

#### 5.2.6 HledaniZarizeni

Je to třída, která vyhledává další mobilní telefony, které mají spuštěnou tuto aplikaci.

Pomocí *discoveryAgent=LocalDevice.getLocalDevice().getDiscoveryAgent()* se najde na zařízení vyhledávací agent. Aplikace používá jako vyhledávacího agenta, agenta *GIAC*.

##### 5.2.6.1 init

Tato metoda slouží k ošetření případu, kdy aplikace již je ve stavu, kdy vyhledává další zařízení v dosahu.

##### 5.2.6.2 commandAction

Obsahuje nastavení a přiřazení funkcí k jednotlivým tlačítkům. Tlačítko *Hledej* slouží k hledání dalších zařízení v dosahu. Funkcí tlačítka je spustit hledání s vybraným vyhledávacím agentem, přidá na displej tlačítko *Stop* a informuje metodu *hledam*, že hledání právě probíhá.

Tlačítko *Stop* informuje metodu *hledam* o tom, že bylo hledání přerušeno a tím s hledání přeruší. Na displeji se zobrazí tlačítko *Hledej*.

Tlačítko *Konec* slouží k ukončení aplikace.

Tlačítko *Pripojit* slouží k připojení k vybranému nalezenému zařízení. Vybranému nalezenému zařízení jako položka seznamu. Na vybrané zařízení se aplikace připojí a spustí klientský formulář pomocí *midlet.NalezeneSluzby(OznZarizeni)*. Tzn. spuštění metody *NalezeneSluzby* třídy *StartMidlet* s parametrem *OznZarizeni*.

### 5.2.6.3 servicesDiscovered

Je to metoda, která se volá při dokončení hledání. Metoda získává BT adresu vzdáleného zařízení. Tato adresa je zobrazena v seznamu nalezených zařízení jako název zařízení.

### 5.2.6.4 serviceSearchCompleted

Tato metoda slouží k zobrazení stavu hledání, přesněji při ukončeném hledání. Její funkcí je změnění nadpisu a vyměnění tlačítka *Stop* za *Hledej*.

### 5.2.6.5 run

Jedná se o tělo vlákna, jehož funkcí je vyhledávání zařízení.

### 5.2.6.6 hledaniSluzby

Tuto metodu volá vlákno vyhledávání. Nejprve zjistí, zda-li bylo nějaké zařízení nalezeno. Dále si uloží do proměnných co na druhém zařízení hledat a spustí vyhledávacího agenta. Tím je spuštěno hledání.

## 5.2.7 Klient

Tato třída zobrazuje serverový formulář s položkami: Počet odeslaných a přijatých zpráv, textové pole pro zprávu k odeslání, stav spojení a textové pole přijatých zpráv. Nastavení položek je obdobné jako ve třídě *Server*.

Rozhraní *commandAction* slouží k nastavení ovládní aplikace. Obsahuje nastavení dvou tlačítek, *Odeslat* a *Konec*, stejně jako u třídy *Server*. Jejich nastavení a funkce je totožná jako u třídy *Server*.

### 5.2.7.1 vytvoreniSpojeni

Tato metoda slouží k připojení klientské části aplikace k serveru. Jestliže je nalezeno nějaké vzdálené zařízení připojí se k němu tak, že odešle třídě *KlientSpojeni*, která se stará o spojení se serverem, parametry vzdáleného spojení.

### 5.2.7.2 Další metody

Ostatní metody třídy *Klient* jsou obdobné jako metody ve třídě *Server*. Jedná se o metody: *poleO*, *PolePrMSG*, *PoleOdMSG* a *PoleZav*.

### 5.2.8 KlientSpojeni

Tato třída je podobná třídě *ServerSpojeni*. Je tu jediný rozdíl a to v tom, že při spuštění vlákna pro připojení, je potřeba znát URL adresu serveru.

### 5.2.9 SpojeníServer a SpojeniKlient

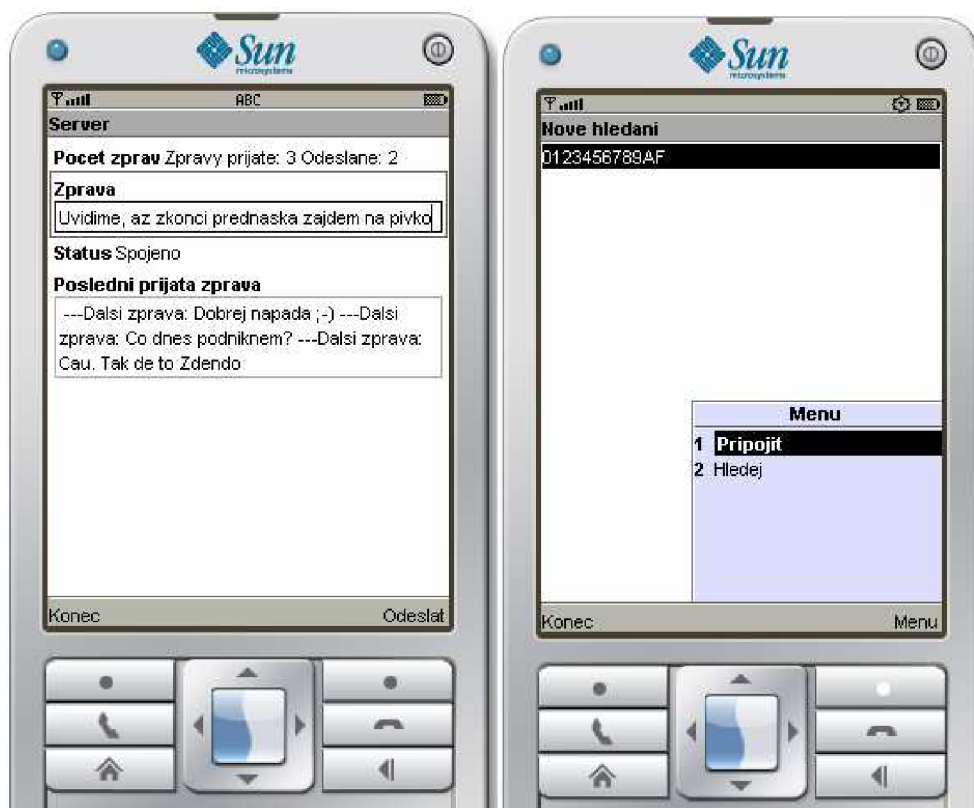
Jedná se o dvě rozhraní, které slouží k informování daných metod. Obě rozhraní obsahují *poleO*, *polePrMSG*, *poleOdMSG*, *PoleZav*. Tyto metody slouží k informování a posílání parametrů metodám ve třídách *Server* a *Klient*. Rozhraní *SpojeniServer* navíc ještě obsahuje metodu *PoleAaO*, která slouží k informování, že byl server vytvořen a čeká na připojení.

### 5.2.10 Výsledek

Obr.10a znázorňuje spuštěný serverový formulář. Obr.10b ukazuje jak vypadá vyhledávání zařízení. Na obr.11 je vidět spuštěný klientský formulář.

Aplikace je vyvíjena pouze pro dva mobilní telefony, tedy nelze vytvořit chat pro více jak dvě zařízení.

Zkompilovaný JAR soubor naleznete na přiloženém CD.



Obr.9a a 9b Spuštěný serverový formulář, Hledání zařízení





Obr.10. Spuštěný klientský formulář

### 5.3 TESTOVÁNÍ APLIKACE

Aplikaci jsem testoval na těchto mobilních telefonech: Sony Ericsson k750i, Nokia 6280, Nokia 5310, Nokia E50. Aplikace byla také testována na emulátoru mobilního telefonu, který je součástí NetBeans.

V emulátoru s aplikací nenastávali žádné problémy.

Aplikace na Sony Ericssonu k750i ve spojení se všemi ostatními telefony pracovala správně.

Také u Nokie 6280 se nevyskytly problémy při běhu aplikace.

U Nokie 5310 nastal v aplikaci jediný problém. Jde o to, že na tomto telefonu musí aplikace vytvořit server, nebyla totiž schopná najít žádné zařízení v dosahu. V čem tento problém spočívá se nepodařilo zjistit.

U Nokie E50 byl podobný problém akorát s tím rozdílem, že pokud aplikace na tomto telefonu vytvořila server, tak jiný telefon se spuštěnou klientskou částí aplikace nebyl schopen server nalézt. Ale pokud byl zvolen klient na Nokii E50, tak sice našel vytvořený server, ale po stisku tlačítka *Pripojit*, aplikace nahlásila neočekávanou výjimku a požadovala ukončení aplikace.

## 6. ZÁVĚR

Tématem bakalářské práce bylo prostudování problematiky programování J2ME aplikací a jejich využití v bezdrátových přenosech ve vybrané bezdrátové technologii. Byla zvolena bezdrátová technologie Bluetooth, protože je implementována na většině dnes dostupných mobilních telefonech.

Byla navržena aplikace umožňující přenos zpráv přes rozhraní Bluetooth. Aplikace byla následně realizována v programovacím jazyku J2ME. Aplikace se skládá z 6 tříd, 2 rozhraní a 1 midletu. Můžeme jí rozdělit na 2 části, klient a server. Obě části mají mnoho podobných částí, liší se pouze ve způsobu spojení. Server spojení vytváří a klient se k serveru připojuje.

Aplikace byla testována na několika mobilních telefonech a v emulátoru v počítači. V emulátoru, na kterém se celá aplikace ladila, nenastává žádný problém s používáním aplikace. Na některých telefonech ale aplikace nefungovala správně a vyskytly se neočekávané chyby. Aplikace se na různých mobilních telefonech zobrazovala odlišně. Jedná se o jiné rozmístění a přiřazení tlačítek ovládání na různých typech mobilních telefonů a jiné zobrazení a vkládání jednotlivých zpráv.

Aplikace by se dala dále ještě rozšířit o další funkce, jako například přenos souborů uložených v telefonu nebo přidání více přívětivého grafického rozhraní. Aplikaci by se mohla využít pro rozhovor v prostorech, kde nesmí být hluk nebo naopak, kde je veliký hluk.

## 7. SEZNAM POUŽITÉ LITERATURY

- [1] MAHMOUD, Qusay H. Naučte se Java 2 Micro Edition. Pavel Makovec. 1. vyd. Praha : GradaPublishing a.s., 2002. 264 s.
- [2] TRČÁLEK, Antonín. Bakalářská práce : Bezdrátový senzor pomocí rozhraní Bluetooth. 2005. 30 s.
- [3] HOPKINS, Bruce. ANTONY, Ranjith. Bluetooth for Java. New York : Apress, 2003. 352 s.
- [4] TOPLEY, Kim. J2ME v kostce Pohotová referenční příručka. Pavel Makovec. 1. vyd. Praha : GradaPublishing a.s., 2004. 536 s.
- [5] Vývoj aplikací . 2005 [cit. 2008-06-01]. Dostupný z WWW: <http://interval.cz>
- [6] MobilMania.cz . 2005 [cit. 2008-06-01]. Dostupný z WWW: [www.mobilmania.cz](http://www.mobilmania.cz)
- [7] JCP . 1995 [cit. 2008-06-01]. Dostupný z WWW: <http://jcp.org>
- [8] HANUS, Stanislav. Rádiové a mobilní komunikace, 2002. 85 s
- [9] Sun Microsystems . 1994 [cit. 2008-06-01]. Dostupný z WWW: <http://www.sun.com>
- [10] NetBeans . 2000 [cit. 2008-06-01]. Dostupný z WWW: <http://www.netbeans.org>
- [11] Remote Method Invocation . 2000 [cit. 2008-06-01]. Dostupný z WWW: <http://nb.vse.cz/~zelenyj/it380/eseje/xnovm44/rmi.htm#lab1>
- [12] Sun Microsystems, Inc.. J2ME Building Blocks for Mobile Devices . 2000 [cit. 2008-06-01]. Dostupný z WWW: <http://java.sun.com/products/cldc/wp/KVMwp.pdf>
- [13] Bluetooth Technology Overview, 2003. 23 s. Dostupný z WWW: <http://www.forum.nokia.com>

## 8. ZKRATKY

<b>ACL</b>	- Asynchronous Connestion-Less	- Asynchronní spojení
<b>API</b>	- Aplication Interface	- Aplikační rozhraní
<b>CDC</b>	- Connected Device Configuration	- Konfigurace pro větší zařízení
<b>CLDC</b>	- Connected Limited Device Configuration	- Konfigurace pro malá zařízení
<b>EDGE</b>	- Enhanced Data rates for GSM Evolution	- Mobilní paketový přenos dat
<b>GPRS</b>	- General Packed Radio Service	- Mobilní paketový přenos dat
<b>GPS</b>	- Global Position Systém	- Satelitní navigace
<b>HCI</b>	- Host Controller Interface	- Rozhraní mezi software a Hardware
<b>HSCSD</b>	- High Speed Circuit Switch Data	- Mobilní přenos dat
<b>IrDA</b>	- Infrared Data Association	- Přenos dat přes infra port
<b>J2EE</b>	- Java 2 Enterprise Edition	- Edice Javy
<b>J2ME</b>	- Java 2 Micro Edition	- Edice Javy
<b>J2SE</b>	- Java 2 Standart Edition	- Edice Javy
<b>JABWT</b>	- Java APIs for Bluetooth Wireless Technology	- Aplikační rozhraní Javy pro Bluetooth
<b>JAR</b>	- Java Archive	- Java archív
<b>JCP</b>	- Java Community Process	- Vývojový proces Javy
<b>JSR</b>	- Java Specification Request	- Žádost o specifikaci Javy
<b>KVM</b>	- Kilo Virtual Machine	- Virtuální stroj Javy pro CLDC
<b>L2CAP</b>	- Logical Link Control and Adaption Protocol	- Bluetooth protokol
<b>LMP</b>	- Link Manager protocol	- Bluetooth protokol
<b>MIDP</b>	- Mobile Information Device Profile	- Profil Javy
<b>OS</b>	- Operation Systém	- Operační systém
<b>PDA</b>	- Personal Digital Asistant	- Osobní digitální asistent
<b>RAM</b>	- Random-Access Memory	- RAM paměť
<b>RFCOMM</b>	- Radio Frequency Communication	- Protokol emulující sériový port RS232
<b>RMI</b>	- Remote Method Invocation	- Java profil
<b>ROM</b>	- Read-Only Memory	- ROM paměť
<b>SCO</b>	- Synchronous Connection-Oriented	- Synchronní spojení
<b>SDP</b>	- Service Discovery Protocol	- Bluetooth protokol
<b>SMS</b>	- Short Message Service	- Krátká textová zpráva
<b>URL</b>	- Uniform Resource Locator	- Jednotný lokátor zdrojů
<b>UUID</b>	- Universally Unique Identifier	- Univerzální unikátní identifikátor

## 9. SEZNAM OBRÁZKŮ

Obr.1. Zásobník Bluetooth protokolu .....	14
Obr.2. Architektura pracovního prostředí J2ME .....	21
Obr.3. Prostředí J2ME .....	23
Obr.4. Pravidla přechodu mezi jednotlivými stavy.....	26
Obr.5. Schéma přístupu MIDP k Bluetooth.....	28
Obr.6. Životní cyklus aplikace.....	31
Obr.7. Sestavení spojení .....	32
Obr.8a. Uvítací zpráva .....	34
Obr.8b. menu .....	34
Obr.9a. Spuštěný serverový formulář .....	40
Obr.9b. Hledání zařízení.....	40
Obr.10. Spuštěný klientský formulář.....	41

## 10. SEZNAM TABULEK

Tab.1 Výkonové třídy Bluetooth .....	15
--------------------------------------	----

## 11. SEZNAM PŘÍLOH

### 11.1 OBSAH CD

Složka BJ Chat :

- build – Složka obsahuje soubory vytvořené při kompilaci aplikace
- dist – Složka obsahuje zkompilovaný JAR soubor aplikace
- nbproject – Složka obsahuje soubory pro NetBeans prostředí
- src – Složka obsahuje zdrojový kód aplikace