

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

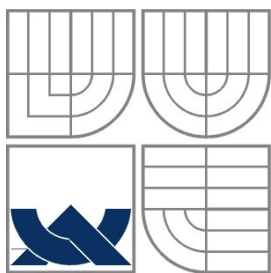
JEDNODUCHÝ IS PRO PDA ZAŘÍZENÍ  
PRO EVIDENCI HOVORŮ

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

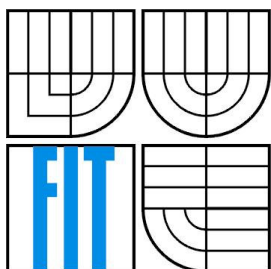
AUTOR PRÁCE  
AUTHOR

JAN DVOŘÁK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# JEDNODUCHÝ IS PRO PDA ZAŘÍZENÍ PRO EVIDENCI HOVORŮ

SIMPLE PDA IS FOR CALL EVIDENCE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN DVOŘÁK

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. FRANTIŠEK ŠČUGLÍK

BRNO 2007

## Zadání bakalářské práce

Řešitel: **Dvořák Jan**

Obor: Informační technologie

Téma: **Jednoduchý IS pro PDA zařízení pro evidenci telefonních hovorů**

Kategorie: Počítačové sítě

### Pokyny:

1. Seznamte se s možnostmi tvorby aplikací pro zařízení PDA
2. Zvolte jedno z vývojových prostředí toto prostředí si prostudujte
3. Navrhněte jednoduchý IS pro evidenci telefonních hovorů s možností zobrazování statistik a přehledů. Předpokládejte komunikaci s MT prostřednictvím Infra nebo BT. Uvažujte možnost kalkulace cen hovorů na základě vyplněné tabulky cen hovorů na jednotlivé mobilní operátory, přehledy pro více mobilních čísel
4. Navržený systém implementujte a otestujte jeho funkčnost

### Literatura:

- Schiller, J.: Mobile Communications (2nd edition), Addison-Wesley 2003, ISBN 0-321-12381-6
- dle doporučení vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Bez požadavků.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Ščuglík František, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2007

Datum odevzdání: 19. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav informačních systémů  
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář  
vedoucí ústavu

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Jan Dvořák**  
Id studenta: 84414  
Bytem: Luční 1044/18, 591 01 Žďár nad Sázavou  
Narozen: 12. 09. 1984, Nové Město na Moravě  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1**  
**Specifikace školního díla**

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

Název VŠKP: Jednoduchý IS pro PDA zařízení pro evidenci telefonních  
hovorů

Vedoucí/školicel VŠKP: Ščuglík František, Ing.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě                      počet exemplářů: 1

elektronické formě                počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

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.

Brně dne: .....

.....

Nabyvatel

.....

Autor

## **Abstrakt**

Tato bakalářská práce se zabývá programováním aplikací pro PDA zařízení, primárně na platformách firmy Microsoft. Dále pak bezdrátovou technologií Bluetooth a její využití v PDA zařízeních a mobilních telefonech. Práce zahrnuje informační systém kde jsou zpracovávány hovory, které byly do PDA načtené pomocí Bluetooth z mobilního telefonu.

## **Klíčová slova**

PDA, .NET Framework, Bluetooth, Microsoft Visual Studio 2008, Windows Mobile, Windows Mobile 6 SDK

## **Abstract**

This bachelor thesis is about programming applications for PDA devices, primary on the Microsoft platforms. Next it deals with wireless technology Bluetooth and its use in the PDA devices and mobile phones. This project includes information system where mobile phone calls, which were loaded to PDA by Bluetooth, are processed.

## **Keywords**

PDA, .NET Framework, Bluetooth, Microsoft Visual Studio 2008, Windows Mobile, Windows Mobile 6 SDK

## **Citace**

Jan Dvořák: Jednoduchý IS pro PDA pro evidenci telefonních hovorů. Brno, 2008, bakalářská práce, FIT VUT v Brně.

# Jednoduchý IS pro PDA pro evidenci hovorů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Františka Ščuglíka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jan Dvořák  
14. května 2008

## Poděkování

Děkuji vedoucímu bakalářské práce Ing. Františku Ščuglíkovi za vstřícnost.

© Jan Dvořák, 2008.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..*

# Obsah

<b>Obsah .....</b>	<b>1</b>
<b>1 Úvod.....</b>	<b>3</b>
<b>2 PDA zařízení .....</b>	<b>4</b>
2.1 Operační systémy .....	4
2.1.1 Windows Mobile.....	4
2.1.2 Palm OS.....	5
2.1.3 Srovnání a výběr .....	5
2.2 Vývoj pro PDA .....	6
2.2.1 Microsoft Visual Studio .....	6
2.2.2 .NET Framework .....	6
2.2.3 Emulace .....	7
<b>3 Bluetooth .....</b>	<b>8</b>
3.1 Historie .....	8
3.2 Vlastnosti a technické parametry .....	8
3.3 Topologie sítě.....	9
3.4 Komunikace .....	10
3.4.1 Vytvoření spojení.....	10
3.4.2 Bluetooth Protocol Stack.....	11
3.5 Komunikační profily .....	12
3.6 Ostatní technologie a srovnání.....	14
<b>4 Návrh aplikace .....</b>	<b>15</b>
4.1 Spojení s mobilním telefonem .....	15
4.2 Tabulka hovorů .....	15
4.3 Statistiky .....	16
4.4 Zadání cen.....	16
<b>5 Implementace .....</b>	<b>17</b>
5.1 Jazyk C# .....	17
5.2 Bluetooth a C# .....	17
5.2.1 Přístup k zařízení.....	17
5.2.2 Přístupové knihovny.....	18
5.2.3 Knihovna InTheHand.Net.Personal.dll .....	18
5.3 Vývojové prostředí .....	18



5.4	Uložení dat.....	19
5.4.1	Uložení cen.....	19
5.4.2	Uložení hovorů.....	20
5.5	Hlavní okno .....	20
5.6	Ostatní prvky.....	21
5.7	Bluetooth klient.....	22
5.7.1	Inicializace a vyhledání dostupných zařízení.....	23
5.7.2	Spojení.....	23
5.7.3	Přenos souboru s hovory.....	23
<b>6</b>	<b>Závěr .....</b>	<b>25</b>
	<b>Literatura.....</b>	<b>26</b>
	<b>Seznam příloh .....</b>	<b>26</b>

# 1 Úvod

Trendem poslední doby v oblasti informační techniky je neustálá miniaturizace a mobilita za všech okolností. Nejenom z těchto důvodů vznikla zařízení o velikosti většího mobilního telefonu se schopnostmi podobnými osobním počítačům, jsou to PDA – malé kapesní počítače. Mnohé výhody, jako je například potřeba uživatelů mít svůj přístroj neustále u sebe, malé rozměry, stále klesající cena, větší množství funkcí a další, činí tato zařízení čím dál tím populárnější. Proto jsem se rozhodl, že se na tato zařízení podívám z tzv. „druhé strany“ a to ne z pohledu uživatele, ale z pohledu vývojáře.

Téměř každý člověk dnes vlastní mobilní telefon a bez něho si dnes již velká většina z nás nedovede život představit. Bez ohledu na to, kolik mají různých funkcí, primární stále zůstává telefonování. Většina mobilních přístrojů má svůj systém na ukládání hovorů, avšak tyto jsou vesměs velmi nepřehledné a často neposkytují informace, které by uživatel potřeboval.

Cílem této práce bude tedy vytvořit systém pro evidenci telefonních hovorů pro PDA zařízení. Pro spojení těchto dvou zařízení využiji bezdrátovou technologii Bluetooth, která bývá nedílnou součástí většiny moderních PDA i mobilních telefonů.

Moje práce je rozdělena na šest kapitol následujícím způsobem:

- **Úvod** – Zasazení do celkového kontextu a formulace cíle.
- **PDA zařízení** – Zde budou poskytnuty informace o PDA zařízeních a o vývoji aplikací.
- **Bluetooth** – V této kapitole podrobně seznámím s technologií Bluetooth.
- **Návrh aplikace** – Tato kapitola se zabývá návrhem řešení, stanovením co a jak má fungovat.
- **Implementace** – Rozeberu klíčové prvky v implementaci.
- **Závěr** – Zhodnocení výsledků a návrhy k případnému vylepšení nebo rozšíření aplikace.

## 2 PDA zařízení

Personal digital assistant či palmtop je malý kapesní počítač, který bývá ovládaný obvykle dotykovým displejem a perem (které se nazývá stylus). Původně měly PDA za cíl především pomoci s organizováním času a kontaktů. Současné PDA jsou velmi výkonné a zvládají i přehrávání videa a velké množství dalších aplikací. Základní funkcí je synchronizace, umožňující výměnu dat mezi PDA a osobním počítačem.

PDA obvykle obsahuje podobné komponenty, jako osobní počítač, jen s poněkud omezenými atributy. Procesor dosahuje frekvence okolo 500 MHz, paměť RAM kolem 64 MB. Tu doplňuje paměť typu Flash ROM, která nepotřebuje k uchování dat přísun energie, proto se využívá k ukládání operačního systému a jiných dat. Dále obsahuje sloty pro paměťové karty a jiná zařízení, případně slot pro SIM kartu, pokud daný přístroj podporuje GSM.



Obrázek 1 - Příklad PDA zařízení

### 2.1 Operační systémy

Mezi nejčastěji používané operační systémy na PDA patří Windows Mobile a PalmOS.

#### 2.1.1 Windows Mobile

Windows Mobile je kompaktní operační systém založený na Microsoft Win32 API určený pro PocketPC (PDA s Windows Mobile). Obsahuje některé základní aplikace, na které jsme zvyklí

z klasických Windows, ovšem v poněkud „jednodušších“ verzích. Mezi nejdůležitější patří například Internet Explorer, Office Mobile nebo Windows Media Player. Samozřejmostí je dotykový displej, použití paměťových karet a podpora protokolů pro bezdrátovou komunikaci.

První verze Windows Mobile byla vydána v roce 2000, pak následovaly další. V roce 2003 to byl Windows Mobile 2003, v roce 2005 Windows Mobile 5.0 a nejnovější oficiálně vydanou verzí je Windows Mobile 6.0.

Poslední zmíněný existuje ve třech verzích a to: „Windows Mobile 6 Standard“ pro smartphony, „Windows Mobile 6 Professional“ pro PDA s integrovaným telefonem (Pocket PC Phone Edition) a „Windows Mobile 6 Classic“ pro PDA bez telefonního modulu.

Z nejnovějších vlastností tohoto systému lze zmínit např.: podpora USB 2.0, podpora rozlišení 800x480 (WVGA), podpora technologií Direct3D, DirectDraw a DirectShow pro vývojáře, schopnost automatických aktualizací systému, podpora VoIP, podpora AJAX a JavaScript v Internet Explorer Mobile, šifrování dat na výměnné paměťové kartě, a mnohé další.

## **2.1.2 Palm OS**

Palm OS je operační systém s grafickým rozhraním a poměrně snadným ovládáním určený pro PDA a komunikátory. Nabízí dotykový displej, podporu multimédií, práci s paměťovou kartou a konektivitu pro IrDA, Bluetooth a Wifi. Jeho výhodou jsou nízké nároky na výkon, paměť a rychlost, jelikož v jednom okamžiku může běžet pouze jedna aplikace. Při přepínání mezi aplikacemi si aplikace uloží poslední stav, takže uživatel nepozná, že se aplikace spouští znovu. Od verze 5 umožňuje PalmOS spustit jednu až dvě aplikace jako rezidentní, což umožňuje například přehrávání mp3 na pozadí.

PalmOS je nejčastěji používán na zařízeních společnosti Palm. Dříve existovalo více výrobců PDA s tímto systémem, postupně však zbyl jediný. Nejnovější verze Palm OS nese označení 5.0

## **2.1.3 Srovnání a výběr**

Největší výhodou systému Windows Mobile je jeho podoba se stolními Windows. Uživatel tohoto OS nebude mít po zakoupení PDA problém s ovládáním, systém je podobně intuitivní. Kapesní počítač je možné také velmi jednoduše propojit s PC, pomocí synchronizačního programu ActiveSync. Poté se mohou z počítače na PDA instalovat programy či procházet soubory na paměťové kartě v kapesním počítači atd. Sandná je i práce se soubory vytvořenými například v kancelářském balíku Office, jehož odlehčenou verzi, jak již bylo zmíněno výše, systém obsahuje.

PalmOS těží ze své minulosti, kdy byl dominantním operačním systémem na kapesních počítačích. Díky tomu existuje pro tento OS mnoho aplikací. Ty jsou ve srovnání s programy pro Windows Mobile leckdy datově mnohem méně náročnější, což už však v dnešní době levných

paměťových karet není až taková výhoda. Programy a celkově ovládání Palmu bývá obecně rychlejší, než v případě PocketPC.

Úvaha o aplikaci nezávislé na platformě je možná, ale řešení by bylo velmi obtížné, proto se budu soustředit pouze na vývoj pro jeden operační systém.

To, jaký operační systém zvolím, je klíčové, protože to zcela ovlivní vývojové prostředí a některé postupy při tvorbě programu. Nakonec jsem vybral Windows mobile z následujících důvodů. Operační systémy firmy Microsoft jsou v současnosti nejrozšířenějšími. Aplikaci psanou pro tuto platformu bude moci používat nejvíce uživatelů. Microsoft také nabízí vhodné nástroje k tvorbě pro PDA, především platformu Microsoft .NET Framework spolu s Microsoft Visual Studio 2008.

## 2.2 Vývoj pro PDA

### 2.2.1 Microsoft Visual Studio

Jako vývojové prostředí jsem zvolil Visual Studio .NET, což je plně integrované prostředí, které je navrženo pro maximální usnadnění procesu psaní, ladění a překladu kódu. V praxi to znamená, že Visual Studio .NET je velmi pohodlné MDI (práce s více okny) rozhraní, v němž můžete libovolně pracovat s kódem. Již je zvykem, že editor kódu má technologii IntelliSense, která automaticky nabízí seznam tříd, metod i argumentů a všechna dostupná přetížení a to je skvělá pomoc při psaní kódu.

Nejnovější verzí je Visual Studio 2008, ve kterém jsem také mojí aplikaci v programovacím jazyku C# psal. S instalací prostředí se také nainstaluje nové operační prostředí .NET Framework 3.5.

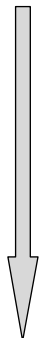
### 2.2.2 .NET Framework

.NET Framework je vlastně balík knihoven, jehož jednotné prostředí nabízí své objektové třídy vyšším programovacím jazykům, aby je využily k práci při řešení konkrétních úloh nezávisle na použitém programovacím jazyce. Platforma .NET nepředepisuje použití žádného programovacího jazyka. Všechny kompatibilní jazyky jsou překládány do společného mezijazyka označovaného jako **Common Intermediate Language (CIL)**. Ten je nezávislý na hardwarové platformě.

#### 2.2.2.1 Součásti .NET Frameworku

Schéma součástí .NET Frameworku je pochopitelné z následující tabulky.

VB.NET	C#.NET	J#.NET	další...
Common Language Specification (CLS)			
Common Type System (CTS)			
Framework Class Library (FCL)			
ADO.NET	ASP.NET	Windows Forms	další (CIL)
Common Language Runtime (CLR)			
<i>Common Language Infrastructure (CLI)</i>			
Operační systém			



Tabulka 1 – Součásti .NET Frameworku

- **Common Language Infrastructure (CLI)** - je otevřená specifikace, která popisuje vlastnosti proveditelného kódu a prostředí pro jeho běh (runtime environment), které tvoří jádro Microsoft .NET Framework. Specifikace definuje prostředí, které umožňuje používání více vysokoúrovňových programovacích jazyků na různých počítačových platformách, aniž by bylo nutné přepisovat jejich překladače s ohledem na zvláštnosti konkrétní architektury.
- **Common Language Runtime (CLR)** - je implementace specifikace CLI
- **Common Language Specification (CLS)** - sada základních pravidel, která by měl splňovat každý jazyk vyhovující specifikaci CLI. Cílem je dosažení vzájemné spolupráce s ostatními jazyky splňující specifikaci CLI.
- **The Common Type System (CTS)** - sada datových typů a operací, které se používají v mnoha programovacích jazycích.
- **Framework Class Library (FCL)** - standartní knihovna v .NET Frameworku

### 2.2.3 Emulace

Z důvodu usnadnění testování aplikací pro PDA, je vhodné pro tento účel použít emulátor zařízení, pro které náš program píšeme. Jak jsem již zmínil výše, já si vybral PDA zařízení s operačním systémem Windows Mobile 6, proto jsem si nainstaloval balík pro tvorbu aplikací Windows Mobile 6 SDK (Software Development Kit), spolu s emulátory pro zařízení Pocket PC a Smartphones různých rozlišení a tvarů displeje.

Pro spouštění emulátorů je ve Visual Studiu komponenta Device Emulator Manager, který umožňuje zapínat jednotlivá zařízení, virtuálně připojit k počítači (cradle), ukládat stav zařízení, restartovat a vypínat.

## 3 Bluetooth

Technologie Bluetooth je standard umožňující bezdrátovou komunikaci na krátké vzdálenosti, jehož původním smyslem bylo odprostit elektronická zařízení od propojovacích kabelů. Mnohé výhody oproti kabeláži a nebo i oproti infračervenému spojení, způsobily, že se technologie Bluetooth velmi rychle rozšířila a dnes je nedílnou součástí lépe vybavených mobilních telefonů, PDA, a mnohých dalších zařízení.

Bluetooth má dnes mnoho různých využití. Od datové komunikace mezi dvěma zařízeními (ať už se jedná o dvě zařízení stejného typu, např. dvě PDA nebo různá zařízení, např. počítač – mobilní telefon), přes přenos hlasu, připojení různých periferií k PC, až po např. vytvoření Ad-hoc lokální sítě mezi počítači.



Obrázek 2 - Příklad zařízení, která mohou komunikovat pomocí Bluetooth

### 3.1 Historie

Slovo „Bluetooth“ bylo odvozeno z anglického jména dánského krále Haralda Modrozuba (Harald Bluetooth), který vládl v 10. Století a diplomatickou cestou sjednotil Dánsko a část Norska pod svoji nadvládu.

Jako první se v roce 1994 společnost Ericsson začala zabývat navrhováním technologie, která by umožnila bezdrátové propojení mobilních zařízení a jejich příslušenství. Hlavními požadavky byla nízká cena komponent, malý příkon a rozměry zařízení. Zanedlouho se ke společnosti Ericsson připojuje Intel, IBM, Nokia a Toshiba a v roce 1998 spolu vytváří skupinu zabývající se výzkumem

bezdrátové technologie Bluetooth (Bluetooth Special Interest Group - SIG), která má za úkol podrobně specifikovat standard Bluetooth. V červenci 1999 Bluetooth SIG publikuje Bluetooth specifikaci verze 1.0.

Prozatím poslední verze je 2.1, však dnes nejrozšířenější zůstává Bluetooth 2.0.

## 3.2 Vlastnosti a technické parametry

Bezdrátová technologie Bluetooth je specifikována standardem IEEE 802.15.1. Stejně jako většina jiných specifikací i Bluetooth zahrnuje pravidla pro implementaci rozhraní i pro návrh periférií. Bluetooth je hlavní představitel sítí s malým dosahem do 10 m, tzv. sítí osobních (PAN, Personal Area Network). Vyznačuje se specifickými vlastnostmi, které popíšu v následujících řádcích.

- nízkovýkonová rádiová komunikace na krátkou vzdálenost několika metrů integrovaná do nejrůznějších zařízení komunikační i spotřební elektroniky.
- schopnost vyhledání komunikujících uzlů v okolí (discovery)
- jednoduchost navazování spojení (párování)
- pro spojení není potřeba přímý dohled mezi zařízeními
- podpora nejen datových, ale i hlasových přenosů (např. profil Headset při připojení bezdrátové handsfree soupravy k přístroji)

Další parametry:

- standardně funguje na frekvenci 2,4 GHz. K přenosu používá metodu SSFH (Spread Spectrum Frequency Hopping), kdy během jedné sekundy provede 1600 přeladění mezi 79 frekvencemi v rozestupu 1 MHz. Tímto mechanismem se předchází rušení na stejné frekvenci.
- výměna dat maximální rychlostí v řádu od stovek kbit/s do 3 Mbit/s (nejvyšší verze 2.1 s využitím EDR - Enhanced Data Rate)

Bluetooth se podle výkonu dělí do 3 tříd a to:

Třída	Maximální povolený výkon	Nominální výstupní výkon	Minimální výstupní výkon	Přibližný dosah
1	100 mW (20 dBm)	není stanoveno	1 mW (0 dBm)	≈ 100 m
2	2,5 mW (4 dBm)	1 mW (0 dBm)	0,25 mW (-6 dBm)	≈ 10 m
3	1 mW (0 dBm)	není stanoveno	není stanoveno	≈ 1 m

Tabulka 2 – Bluetooth technické parametry

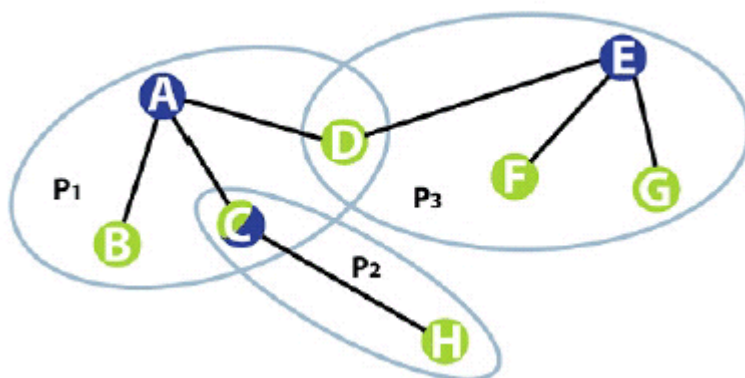


## 3.3 Topologie sítě

Každé Bluetooth zařízení je identifikováno pomocí své adresy (Bluetooth Device Adres). Systém Bluetooth umožňuje spojení dvou zařízení mezi sebou (*point – to – point*) a nebo jedno zařízení s více (*point – to – multipoint*).

Zařízení jsou seskupeny ve skupinách, nazývajících se *Piconety*. V každém Piconetu je jedna stanice zvolena jako *master* a ta může obsloužit až sedm zařízení typu *slave*. I zařízení v jednotlivých Piconetech mezi sebou můžou komunikovat.

Na následujícím obrázku je příklad topologie, kde P1, P2 a P3 jsou jednotlivé Piconety a A, C a E jsou typu *master* a B, D, F, G a H jsou typu *slave*. Za zmínku stojí, že zařízení C vystupuje v jednom Piconetu jako *slave* (P1) a ve druhém zároveň jako *master* (P2). Toto topologické uspořádání, které se nazývá *scatternet*, umožňuje propojení více sítí piconet.



Obrázek 2 – Příklad topologie sítě Bluetooth

## 3.4 Komunikace

Bluetooth umožňuje využívat dva typy komunikačních kanálů, a to:

- Asynchronní (ACL = Asynchronous Connectionless)  
Tento typ přenosu je využíván především pro přenos dat, protože dosahuje vyšších přenosových rychlostí než SCO.
- Synchronní (SCO = Synchronous Connection Oriented)  
Ten je využíván především pro přenos videa a zvuku, díky nízké odezvě oproti ACL.

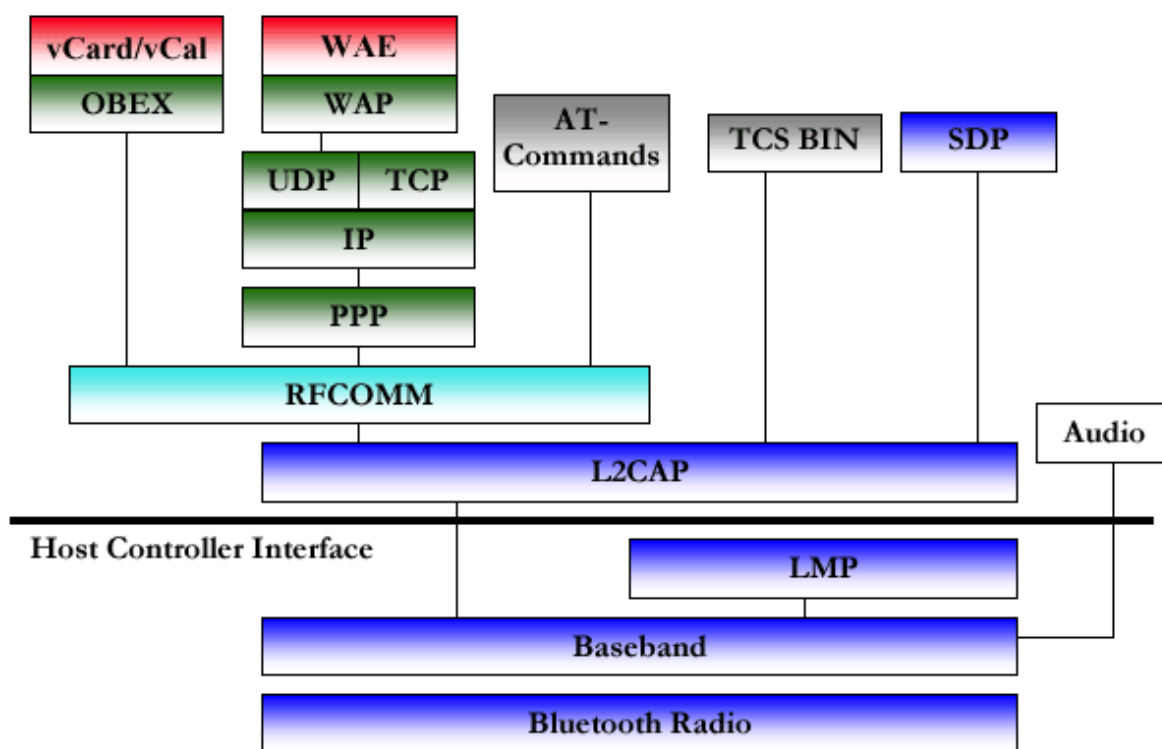
### 3.4.1 Vytvoření spojení

Zařízení, které není připojeno k Piconetu, je v pohotovostním režimu a v intervalu 1,28 vteřin čeká na zprávu. Pokud se zařízení chce připojit k jinému, pak vyšle 16 různých zpráv na 16

frekvencích. Pokud slave neodpovídá, pak master pošle zprávu na jiných 16 frekvencích. Pokud master nezná adresu zařízení slave, potom musí zprávu předcházet „pátrací“ zpráva, která vyžaduje speciální odpověď od zařízení slave. Pokud slave odpoví na zprávu od mastera, pak může master začít vysílat data.

### 3.4.2 Bluetooth Protocol Stack

Základem specifikace Bluetooth je tzv. „Bluetooth protocol stack“, který definuje, jak celá technologie funguje. Rozděluje Bluetooth na jednotlivé vrstvy a to způsobem znázorněným na následujícím obrázku a popsáním níže.



Obrázek 3 – Bluetooth Protocol Stack

#### 3.4.2.1 Nižší vrstvy

- **Bluetooth Radio** – Je to nejnižší vrstva, která se stará o modulaci a demulaci signálu a popisuje fyzické požadavky na Bluetooth vysílač a přijímač konkrétního zařízení.
- **Baseband** - Tato vrstva se stará o formátování dat do podoby vhodné pro přenos vzduchem a synchronizaci spojení. Spravují oba dva typy kanálů definovaných ve standardu Bluetooth – SCO a ACL. Komunikaci řídí pomocí algoritmu pseudonáhodné přeskokové sekvence.

- **Link Manager Protocol (LMP)** – Vrstva navazuje a udržuje spojení mezi zařízeními. Také je zodpovědná za řízení a sestavení komunikace, potažmo dohodnutí délky paketů. Její další činností je řízení napájecích módů a tudíž i spotřeba. Další prací této vrstvy je generování, výměna a řízení kanálů, také výměna šifrovacích klíčů pro autentifikaci a šifrování.
- **Host Controller Interface (HCI)** – Tato vrstva se stará o spolupráci vyšších vrstev s vrstvami nižšími. Vytváří pomyslnou čáru mezi hardwarem a softwarem. V některých případech není HCI vyžadován.

#### 3.4.2.2 Vyšší vrstvy

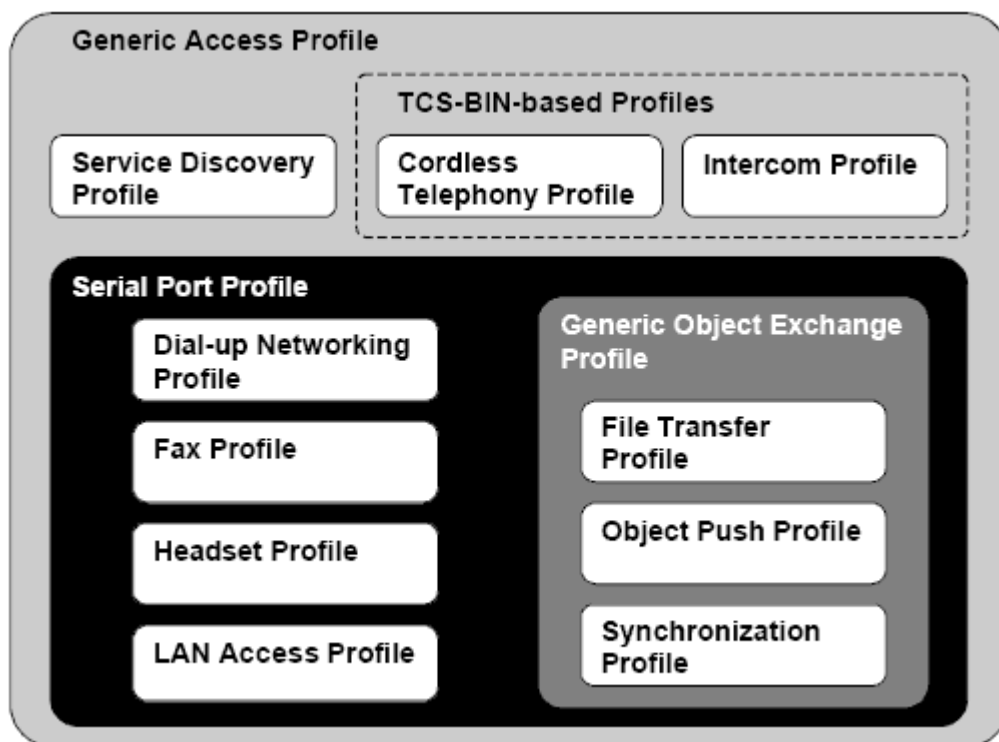
- **Logical Link Control and Application Protocol (L2CAP)** – Tato vrstva má na starosti zapouzdření paketů do formátu vhodného pro nižší vrstvy, multiplexování spojení tak, aby mohlo být využíváno více aplikacemi, atd.
- **Radio frequency communications port (RFCOMM)** - Umožňuje emulovat propojení sériovým kabelem, přičemž má všechna specifika standardu RS-232. Díky tomu mohou přes Bluetooth komunikovat aplikace navržené pro komunikaci sériovým portem.
- **Service discovery Protocol (SDP)** – Tento protokol definuje, jakým způsobem Bluetooth klient hledá služby Bluetooth serveru.
- **Telephony control protocol-binary (TCS BIN)** – Zajišťuje řízení linky pro přenos hlasu a dat mezi Bluetooth jednotkami.
- **Audio** – Standard, který definuje služby pro přenos zvuku.

## 3.5 Komunikační profily

Bluetooth profily zajišťují mezi zařízeními vzájemnou slučitelnost na nejvyšší programové úrovni a také určují vzájemný způsob komunikace. To znamená, že profily jsou zde od toho, aby byla eliminována případná nekompatibilita vyvíjených programů u Bluetooth zařízení různých výrobců. Každý profil zastupuje jinou aplikaci, jiný úkol resp. možné použití technologie Bluetooth.

Specifikace profilu musí zahrnovat závislosti na jiných profilech, navrhované formáty uživatelského rozhraní a části Bluetooth protocol stacku využívaného tímto profilem.

V následující části ukážu systém komunikačních profilů na zjednodušeném obrázku a nejdůležitější z nich níže popíšu.



Obrázek 4 – Bluetooth komunikační profily

- **Generic Acces Profile (GAP)** je hlavním, base profilem. Vycházejí z něj všechny další profily, čímž je zaručena všeobecná kompatibilita. GAP zajišťuje mimo jiné například vyhledání či navázání spojení dvou Bluetooth zařízení. Zajišťuje i základní zabezpečení. Tento profil je povinný pro všechna zařízení.
- **Service Discovery Application Profile (SDAP)** slouží k vyhledání služeb na vzdáleném zařízení pomocí SDP (viz. Bluetooth Stack Protocol)
- **Serial Port Profile (SPP)** umožňuje emulaci RS-232 sériového rozhraní na Bluetooth zařízení.
- **Generic Object Exchange Profile (GOEP)** slouží jako základ k výměně datových objektů a definuje role klienta a serveru.
- **Dial-up Networking Profile** určuje způsob, kterým má zařízení využívat modem pro připojení k telefonní síti.
- **Headset Profile** popisuje způsob, kterým Bluetooth headset komunikuje s telefonním přístrojem nebo stolním počítačem.

Kompletní popis všech profilů najdeme ve specifikaci.

## 3.6 Ostatní technologie a srovnání

Pro komunikaci PDA s mobilním telefonem se dají použít i jiné způsoby než pomocí **Bluetooth**. Existují zde ještě dvě další významné bezdrátové technologie a to **IrDA** a **IEEE 802.11b**.

IrDA se používá k propojení dvou zařízení (např. PC – mobilní telefon, PDA – mobilní telefon, TV – ovladač, apod.). IrDA umožňuje pouze spojení na krátkou vzdálenost a přímou viditelnost. Přenosové rychlosti jsou u tohoto typu spojení nízké.

IEEE 802.11b se nejčastěji používá k připojení počítačů, resp. notebooků do lokální sítě. Má poměrně veliký dosah (řádově desítky až stovky metrů), vysoké přenosové rychlosti a nevyžaduje přímou viditelnost mezi zařízeními.

Bluetooth je dnes pro bezdrátovou komunikaci mezi malými přenosnými zařízeními nejrozšířenější. Má sice relativně malý dosah signálu, řádově desítky metrů, avšak pro spojení PDA a mobilního telefonu více než dostačující. Dále má nízký vysílací výkon, potažmo nízké energetické nároky, a proto se do těchto přenosných zařízení velmi hodí.

Následuje srovnání těchto 3 technologií v přehledné tabulce.

Typ	IrDA	Bluetooth	IEEE 802.11b
Přenos	Optické paprsky	Rádiové vlny	Rádiové vlny
Kmitočet	850 nm	2,4 GHz	2,4 GHz
Vysílací výkon	40-500 mW/Sr	10-100 mW	100 mW
Max. fyz. Rychlost	9600 bps – 16 Mbps	1 Mbps	11 Mbps
Dosah	1 m (směrově)	10-100 m	100 m
Počet účastníků v síti	2	8	32
Adresování	32-bit ID	48-bit MAC	48-bit MAC
Relativní spotřeba el. en.	Minimální	Nízká	Vysoká
Relativní cena	Nízká	Střední	Vyšší

Tabulka 3 – Srovnání bezdrátových technologií

## 4 Návrh aplikace

Mým úkolem je vytvořit jednoduchý program pro PDA v podobě informačního systému, který bude Bluetooth klienta. Ten se připojí k mobilnímu telefonu a načte z něj data, resp. telefonní hovory, v určitém formátu, které byly na daném zařízení provedeny. Poté moje aplikace data analyzuje a uloží v upraveném formátu do tabulky, kterou na vyžádání zobrazí na PDA. Systém bude hovory následně zpracovávat a na základě zadaných cen pro zahraniční nebo domácí hovory bude vypočítávat náklady. Dále bude možné zobrazovat jejich statistiky jako celková cena a celková doba trvání domácích, zahraničních a všech hovorů. Také budou vypočítávána cenová a časová maxima.

### 4.1 Spojení s mobilním telefonem

Aplikace bude obsahovat Bluetooth klienta, jehož ovládací okno bude možno spustit z hlavního okna programu. Jednou z funkcí klienta bude vyhledání všech dostupných zařízení v okolí a zobrazení pro uživatele. Ten si pak vybere, ke kterému zařízení se chce připojit a poté bude možno požádat klienta o stažení požadovaných dat z mobilního telefonu.

### 4.2 Tabulka hovorů

Základem programu pro evidenci hovorů bude hlavní okno, ve kterém bude zobrazena tabulka načtených hovorů. U každého hovoru bude zobrazeno v jednotlivých sloupcích:

- číslo, na které uživatel volal
- datum hovoru
- čas hovoru
- délka hovoru
- cena hovoru

Toto hlavní okno bude zároveň sloužit jako výchozí bod, z kterého bude se bude uživatel dostávat k dalším funkcím programu. Dvě tlačítka na spodní straně PDA budu sloužit jako přístupové body pro menu a pro načtení hovorů.

V menu budou volby pro zadání cen hovorů, výpis statistik, nápověda, informace a samozřejmě možnost ukončení aplikace.

Druhé tlačítko bude sloužit jednak pro přímé načítání hovorů do tabulky a také pro spuštění formuláře klienta. Aplikace bude také schopná si pro testovací účely vytvořit vlastní soubor s hovory,

ze kterého také může načíst data do tabulky. Dále se zde budou moci uložit hovory do souboru, který si uživatel bude moci prohlížet.

## 4.3 Statistiky

Formulář se statistikami bude dostupný z hlavního okna aplikace. Okno se statistikami bude mít pro uživatele informativní charakter. A vše bude samozřejmě vypočítáváno na základě zadaných cen hovorů a načtených hovorech v tabulce.

Statistiky budou sestávat ze tří částí a to:

- Délky – Zde budou zobrazeny tři informace – celková délka všech domácích hovorů, celková délka všech zahraničních hovorů a celková délka všech hovorů.
- Ceny – Zde budou zobrazeny také tři informace – celková cena všech domácích hovorů, celková cena všech zahraničních hovorů a celková cena všech hovorů.
- Maxima – V této části se uživatel dozví, jaký byl nejdelší a nejdražší hovor.

## 4.4 Zadání cen

Pro výpočet a zobrazení veškerých cen je nutné nejdříve zadat cenu za minutu hovoru a to jak domácího, tak i zahraničního. Pro tento účel vytvořím formulář, kde uživatel tyto ceny zadá. Pokud se tak nestane, statistiky se vypíší bez cen.

Domácí hovory odliším od zahraničních pomocí české předvolby „+420“, ta jednoznačně určuje domácí hovor.

Uvažoval jsem také o rozlišení cen hovorů pro jednotlivé operátory podle čísel, ale dnešní stav, kdy může uživatel přejít se svým číslem od jednoho operátora k jinému, to bohužel nedovoluje. A příslušnost k určitému operátorovi už není možné podle prvních tří číslic určit.

## 5 Implementace

Pro psaní aplikací pro PDA s operačním systémem Windows Mobile jsem zvolil vývojové prostředí Visual Studio a Windows Mobile SDK. Spolu s ním jazyk C# a jeho běhové prostředí .NET Framework. Pro Bluetooth komunikaci se v tomto jazyce dají použít mnohé knihovny, z nichž některé jsou zdarma.

Základem pro psaní mého programu byla správná instalace všech potřebných součástí vývojového prostředí, kterou popíši v příloženém manuálu.

Samotný vývoj aplikace probíhal v Operačním systému Windows Vista.

### 5.1 Jazyk C#

Jazyk C# je objektivě orientovaný programovací jazyk vyvinutý zároveň s platformou .NET Framework. Je to poměrně nový jazyk, založený na jazycích C++ a Java.

C# má široké využití, nejen v tvorbě softwaru pro mobilní zařízení, ale i při psaní databázových programů, webových aplikací a stránek, webových služeb nebo formulářových aplikací ve Windows. Jevil se proto jako nejvhodnější prostředek pro psaní programu pro Windows Mobile ve Visual Studiu na platformě .NET.

### 5.2 Bluetooth a C#

K tomu, aby jazyk C# mohl pracovat s technologií Bluetooth, se dají použít různé způsoby. Já jsem zvolil způsob, který se mi zdál být pro přístup k Bluetooth nejlepší.

#### 5.2.1 Přístup k zařízení

Každý osobní počítač může být vybaven různými druhy Bluetooth přijímačů, resp. vysílačů. Každé toto zařízení potřebuje mimo ovladačů i takzvaný „Bluetooth Stack“. To je software, který ovládá modul zařízení a přístup k němu. Mezi nejrozšířenější Bluetooth stacky pro Microsoft Windows patří zejména:

- *Microsoft Windows stack*, který bývá součástí dnešních Windows Vista nebo XP.
- *Widcomm* bylo vůbec prvním Bluetooth rozhraním pro operační systémy Windows. Dnes s ním pracuje mnoho přístrojů od známých výrobců.
- *Toshiba stack* je dalším rozšířením pro systémy Windows.



Problémem je, že ne každá knihovna, kterou můžu pro řízení Bluetooth použít, umí ovládat zařízení. Proto je nutné zvolit programový přístup podle použitého Bluetooth stacku.

## 5.2.2 Přístupové knihovny

Z internetu je možné stáhnout vícero knihoven pro přístup k zařízení, ne každá z nich však bude fungovat s daným Bluetooth stackem. Proto jsem musel vybírat také z toho hlediska, aby nalezená knihovna fungovala s daným stackem fungovala, potažmo aby umožnila přístup k zařízení.

Jako nejvhodnější pro účely mé práce se ukázala knihovna *InTheHand.Net.Personal*. Nevýhodou této knihovny je, že přistupuje pouze k Microsoft Windows stacku. Naopak velkou výhodou je její volná dostupnost. Knihovnu je možné ji stáhnout ze stránek výrobce In The Hand Ltd, [4], kde je k dispozici spolu s manuálem a různými příklady zcela zdarma.

Alternativně lze také pro Bluetooth zařízení, která nepoužívají Microsoft Windows stack, stack změnit. Jde však pouze o neoficiální návod, který nemusí být funkční, proto jsem ho v řešení neuvažoval.

## 5.2.3 Knihovna InTheHand.Net.Personal.dll

Na tuto knihovnu musí být v mém projektu přidána reference. Aby bylo možné používat objekty jako jsou `BluetoothService`, `BluetoothAddress`, `BluetoothClient`, případně některé další, musí se uvést názvy jednotlivých namespaces a to:

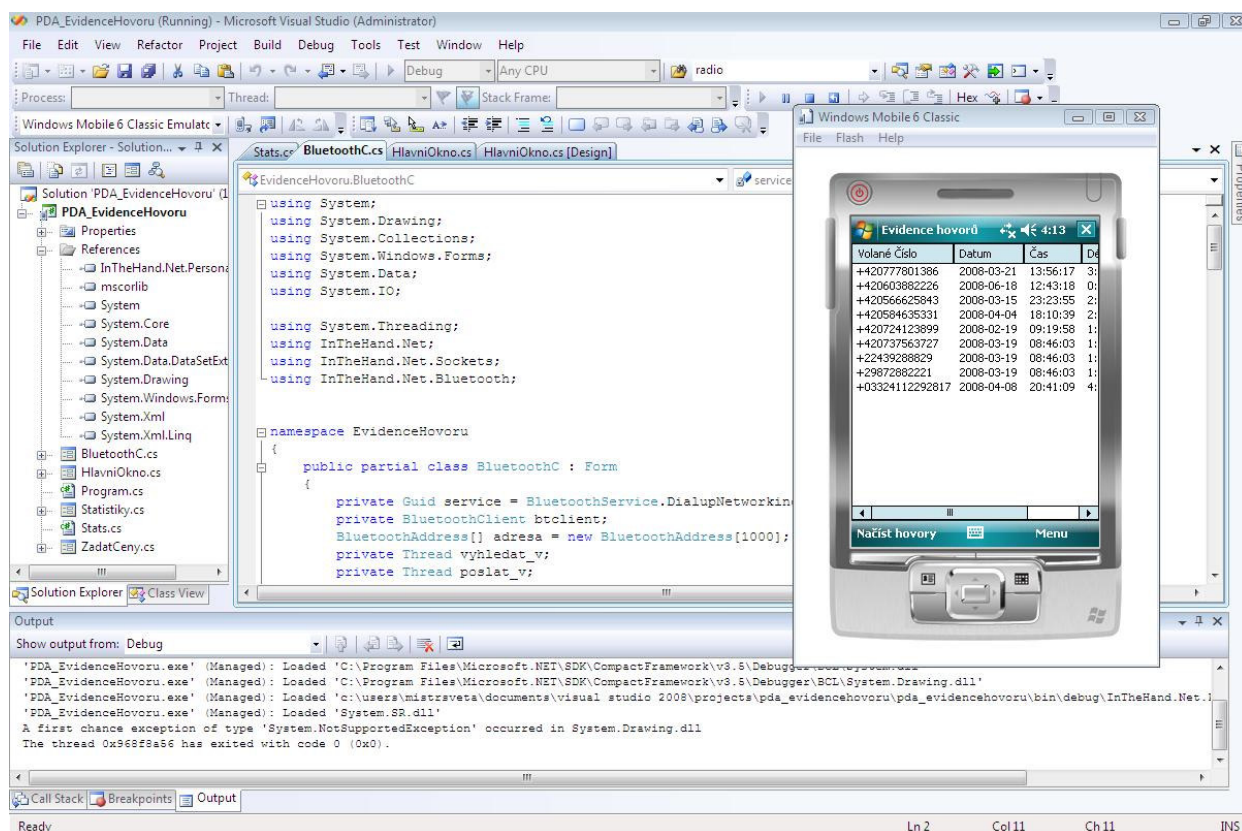
- `InTheHand.Net.Sockets`
- `InTheHand.Net.Bluetooth`

Podrobný popis jednotlivých namespaces se dá nalézt na stránkách výrobce [3]

## 5.3 Vývojové prostředí

Po spuštění Visual Studia 2008 založíme nový projekt, dále vyberem jazyk C# a typ projektu *Smart Device Project* a zvolíme verzi .NET Frameworku. Mnou vybraná verze byla *.NET Framework 3.5*, která má však jen malá rozšíření oproti 3.0, která nejsou pro mou práci podstatná. Proto by programování i samotné fungování projektu mělo probíhat stejně, jako s použitím verze 3.0. Také je třeba zadat jméno projektu, v tomto případě jsem zadal *PDA\_EvidenceHovoru* a cestu. Nakonec jsem ještě zvolil šablonu *Device Application* a jako cílovou platformu *Windows Mobile 6 Professional SDK*. Tento balík je třeba mít nainstalovaný na svém počítači. Obsahuje veškeré nástroje, které umožňují tvorbu aplikací pro operační systém Windows Mobile 6.

Po zadání všech těchto náležitostí je vygenerován základ pro náš nový projekt, do kterého už můžeme začít psát náš kód případně editovat vzhled formulářových oken pomocí editoru.



Obrázek 5 – vývojové prostředí Visual Studia 2008 a emulátor Pocket PC

## 5.4 Uložení dat

Pro popis kódu aplikace je nejdříve vhodné, vysvětlit si způsob uložení dat v PDA zařízení.

V mé aplikaci se se pracuje s dvěma druhy dat a to s *cenami hovorů*, které zadává uživatel a *záznamy hovorů* jako takovými, které jsou do PDA umístěovány buď pomocí Bluetooth nebo ručně a poté ukládány.

### 5.4.1 Uložení cen

Ceny hovorů jsou rozděleny na zahraniční a domácí a obě dvě hodnoty musí být zachovány i po restartu zařízení, proto jsem se rozhodl ukládat tato data do souboru v textovém formátu a to následovně v souboru s názvem *ceny.txt*.

Na prvním řádku souboru je uložena cena za minutu volání po České republice a na druhém cena za minutu volání do zahraničí. Cena může mít až 8 cifer.

## 5.4.2 Uložení hovorů

Jednotlivé záznamy s hovory musí mít jednotný tvar, který je třeba mít dohodnutý se stranou serveru. Proto všechny záznamy hovorů mají následující tvar:

```
telefonní_číslo@datum@čas@délka@
```

například takto:

```
" +420566625843@2008-03-15@11:23:55@2:18@ "
```

Jednotlivé záznamy hovorů se nacházejí v souboru *hovory.txt*. Záznamy hovorů jsou v tomto souboru uloženy na jednotlivých řádcích, tzn. na každém řádku jeden záznam. Tento soubor obdržíme od serveru, tedy mobilního telefonu.

Uživatel může chtít hovory uchovat v přístroji pro případné prohlížení a vyhodnocování, proto je zde možnost uložení do souboru *Vypis\_hovoru.txt* v přehledném formátování.

## 5.5 Hlavní okno aplikace

Základním bodem pro můj program je hlavní okno aplikace implementováno třídou *HlavniOkno.cs*, která je spouštěná první třídou programu, jmenovitě *Program.cs*. Po vytvoření hlavního okna jsem postupně implementoval další třídy, které zastávají zbylé funkce.

Třída *HlavniOkno.cs* má na starosti zobrazovat tabulku s načtenými hovory a slouží také jako přístupný bod pro ostatní funkce programu. Při spuštění programu se načte metodou *Application.Run()*.

Na začátku jsou definovány potřebné systémové namespace pomocí klíčového slova *using*.

První metodou v této třídě je *NacteniZeSouboru()*. Ta se pokusí otevřít soubor *ceny.txt* a přečíst z něj ceny. V případě, že tento soubor neexistuje a ceny ještě nebyly zadány, je uživatel tázán, zda jej chce vytvořit a ceny zadat a nebo pokračovat dál, ovšem bez možnosti výpočtu cen. Poté se začne se čtením souboru *hovory.txt* (tento soubor se do PDA dostane buď pomocí Bluetooth nebo jej vytvoříme sami, k tomuto se dostanu v následujících kapitolách). Pokud je už soubor s hovory načtený v telefonu, pak se z něho přečtou jednotlivé řádky, z kterých se pomocí regulárních výrazů vyčlení jednotlivé informace jako jsou: *číslo hovoru*, *datum*, *čas*, *délka* a následně i *cena*.

Při průchodu souborem jsou zároveň zjištěny záznamy, které obsahují zahraniční hovor a které domácí, podle toho jsou dále vypočítávány další hodnoty. Těmito vypočítanými hodnotami jsou: *cena všech domácích volání*, *cena všech zahraničních volání*, *cena celkem*, pak *trvání hovorů* ve stejném

rozdělení a nakonec *nejdelší a nejdražší hovory*. Nakonec jsou všechny tyto hodnoty zobrazeny v prvku *ListView* na hlavním okně.

Za zmínku stojí ještě metoda *Uloz()*, která na požádání prochází celou zobrazenou tabulku v *ListView* a jednotlivé řádky (resp.) záznamy formátuje, očísluje a ukládá do souboru *Vypis\_hovoru.txt*.

Dále jsou v této třídě definována dvě hlavní tlačítka a to *Hovory* a *Menu*. Stisknutím tlačítka ***Hovory*** se dostaneme do submenu, ve kterém je nabídka:

- ***Spojit s telefonem*** – Otevře formulář *BluetoothC.cs* pro připojení k telefonu.
- ***Načíst ze souboru*** – Načte hovory ze souboru existujícího na tomto PDA zařízení.
- ***Uložit hovory*** – Uloží hovory do *Vypis\_hovoru.txt*.
- ***Vyčistit tabulku***

Druhým tlačítkem je ***Menu*** a pod ním:

- ***Statistiky*** – Otevře formulář se statistikami *Statistiky.cs*.
- ***Zadat ceny*** – Načte Otevře formulář pro zadávání cen *ZadatCeny.cs*.
- ***Nápověda*** – Zobrazí dialogové okno se stručnou nápovědou.
- ***O programu*** – Zobrazí dialogové okno se stručným info o programu
- ***Konec***

## 5.6 Ostatní prvky

V programu existují mimo hlavního okna i další třídy.

Třída *Statistiky.cs*, jak již bylo zmíněno výše slouží pro zobrazení statistik pro klienta. Tato třída úzce spolupracuje s třídou *stat.cs*, ve které jsou definované instance proměnné a metoda pro předávání statistik. Samotná třída *Statistiky.cs* obsahuje pouze definici tlačítek *Přepočítat* a *Zpět* a funkci pro převod přijímaných čísel na požadované jednotky s případným zaokrouhlením a jejich zobrazení ve výsledném formátu.

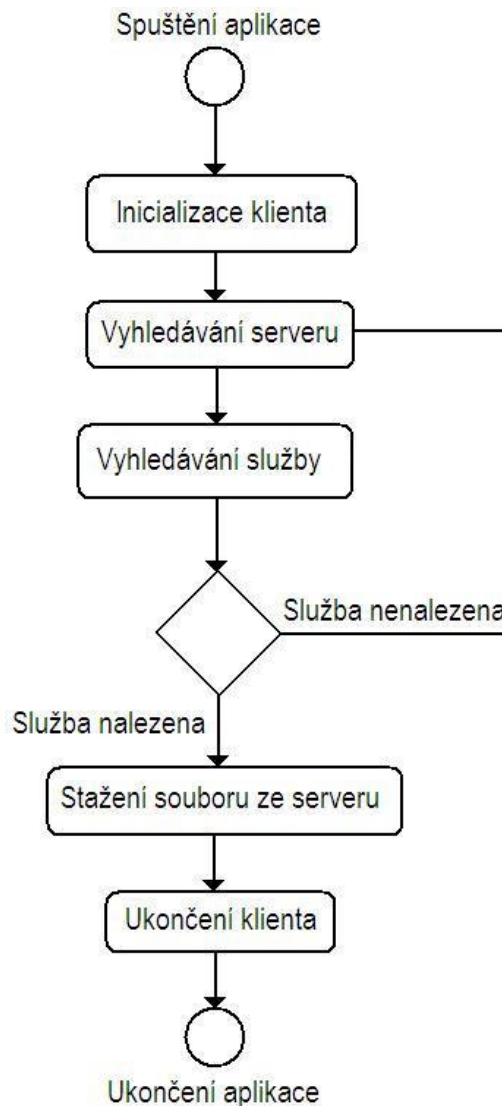
Další důležitou třídou je *ZadatCeny.cs*. Která od uživatele vyžádá ceny hovorů a s nimi pak dále pracuje. Tato třída obsahuje drobné metody jako jsou například *ZobrazitCenyDialog()*, která se stará o vypsání cen v kontrolním dialog boxu, *ValidateNumber()*, která ověřuje, jestli bylo zadáno číslo a nebo *ZapisCenyDoSouboru()*. Ta ukládá ceny do souboru v požadovaném formátu, přičemž případnou prázdnou hodnotu při zadávání cen nahrazuje nulou.

## 5.7 Bluetooth klient

Obečně se dá k problému přistoupit tak, že beru bluetooth stejně jako síťovou kartu a z toho pohledu, že když mám bluetooth tak mám síťové rozhraní, když mám síťové rozhraní tak mám možnost použít sokety, a to už je krůček k tomu si napsat si klient/server aplikaci vystavěnou na klasickém TCP. Situace v jazyce C# ovšem byla ovšem trochu odlišná a bylo třeba najít knihovnu, která přístup a práci s Bluetooth umožnila. V jazyce C# umožňuje využívat technologii Bluetooth výše popsaná knihovna *InTheHand.Net.Personal.dll*.

Moje aplikace je vytvořena podle modelu *klient-server*. Já však tvořím pouze klienta, který se k serveru připojuje, stahuje data a pracuje s nimi.

Celý cyklus komunikace se serverem je znázorněn na následujícím obrázku.



Obrázek 6 – cyklus komunikace klienta

Můj Bluetooth klient je implementován v třídě *BluetoothC.cs*.

### 5.7.1 Inicializace a vyhledání dostupných zařízení

První co je potřeba provést je inicializace klienta. Je potřeba zkontrolovat, zda zařízení, na kterém běží aplikace, má zapnutou jednotku Bluetooth. Jinak připojování skončí chybou, proto je nutné toto zkontrolovat v bloku *try-catch*.

Po startu hledá klient všechna dostupná Bluetooth zařízení. Přičemž se počítá s tím, že na mobilním telefonu server již běží a že je zapnuto Bluetooth. Není-li server spuštěn, spojení nemůže být navázáno a žádné zařízení se nevyhledá. Po dokončení vyhledávání je zobrazen seznam všech dostupných zařízení, z kterých si klient vybere partnera pro navázání spojení.

K tomu se používá metoda *Vyhledat()*. V té si na začátku přepnu *BluetoothRadio* do módu *Discoverable* tímto způsobem `BluetoothRadio.PrimaryRadio.Mode = RadioMode.Discoverable`, samozřejmě v bloku *try-catch* pro případ, že by Bluetooth zařízení nebylo aktivní.

Pak vytvořím objekt typu *BluetoothClient*, z knihovny *InTheHand.Net.Personal.dll* (*.Net.Sockets*). V této metodě dále využíváme objekt typu *BluetoothDeviceInfo[]*, který nám vytvoří prostor pro uložení informací o zařízení v podobě *ArrayListu*. Metoda *DiscoverService()* pak nalezne potřebné informace o zařízení jako jsou jméno a adresa, a zobrazí je v seznamu umístěném uprostřed okna klienta.

### 5.7.2 Spojení

Klient nejprve vyhledá všechna zařízení v dosahu a pak se u nich snaží vyhledat požadovanou službu (tzn. zda je spuštěný daný server). Pokud se mu podaří nalézt zařízení s běžícím serverem, pokusí se od něj získat jeho URL adresu, na kterou se vzápětí připojí a tím dojde k sestavení spojení se serverem. Pokud službu nenachází, je vrácen zpět ke hledání serveru.

K navázání spojení s vybraným serverem slouží tlačítko *Spojit*. Po stisknutí se zjišťuje, zda bylo vybráno vhodné zařízení ze seznamu. Poté se definuje end point objektem *BluetoothEndPoint* a připojí se k serveru metodou, obecně: *klient.Connect(server)*.

### 5.7.3 Přenos souboru s hovory

Klíčovým bodem je přenos souboru, ve kterém jsou v daném formátu záznamy hovorů. Počítám s tím, že server už vytvořil soubor *hovory.txt* s daty a je připraven k odesílání.

K samotnému přenosu využívám protokol *OBEX (Object EXchange)*, (designem a funkcemi podobný protokolu HTTP), což je protokol, vytvořený k přenosu binárních dat technologií IrDA nebo Bluetooth.

*OdeslatSoubor()* je metoda třídy *BluetoothC.cs*, která se stará o výše uvedené. Zde vytvořím nový objekt *ObexListener(ObexTransport.Buletooth)*, na který pak budu posílat soubor. Nastartuji ho metodou *Start()*. Poté je zde pomocí třídy *ObexListenerContext* a metody *GetContext()* čekám na příchozí žádost o zaslání souboru a třídou *ObexListenerRequest* a její metodou *Request()*, která reprezentuje klientovu žádost o sobor. Poté je na klientovi vytvořen soubor (v tomto případě *hovory.txt*) a do něj je metodou *Write()* zapisován výsledný objekt typu *ObexListener*.

Pokud nedošlo k vyjímce, je nakonec komunikace ukončena metodou *Stop()*.

## 6 Závěr

Jako téma bakalářské práce jsem si vybral „Jednoduchý IS pro PDA pro evidenci hovorů“, jelikož oblast mobilních technologií je mi blízká. Avšak vývoj pro PDA na platformě .NET byl pro mě naprostou novinkou, proto jsem musel vše studovat od začátku. Po proniknutí hlouběji do problematiky jsem ovšem objevil netušené možnosti.

Cílem této bakalářské práce bylo vytvořit program pro PDA v podobě jednoduchého informačního systému, který spravuje hovory z mobilních zařízení a k přenosu dat využívá bezdrátovou technologii Bluetooth

Uživateli by měl program sloužit pro vytvoření většího přehledu o jeho uskutečněných hovorech. Vypočítávat ceny hovorů, jejich délky a další hodnoty. Výsledky si pak uživatel pro lepší prostudování může uložit do souboru.

Pro vytváření mé aplikace jsem zvolil platformu .NET, která poskytuje skutečně rozsáhlé možnosti při vývoji aplikací a to nejen pro PDA. V rámci této platformy jsem musel vybrat i jazyk. Z důvodu předchozích zkušeností s jazyky se základem v C, jsem zvolil objektově orientovaný programovací jazyk C#, s kterým jsem se musel od základů seznámit. Díky předchozím zkušenostem jsem jako vývojové prostředí vybral Visual Studio 2008, jenž je podobné verzi 2005, kterou jsem znal z dřívějšíka.

Při psaní této práce jsem se musel vypořádat s mnoha problémy. Mezi největší z nich patřil způsob přístupu k Bluetooth v jazyce C#. K tomuto tématu jsem musel na internetu dlouho hledat možná řešení. Nakonec se objevilo jedno, které se později ukázalo jako nejlepší, a to použití výše zmíněné knihovny InTheHand.Net.Personal.dll. To má však za následek nevýhodu celé aplikace, která je díky tomu schopná spolupráce pouze s ovladači Microsoft Windows Stack. Tomu by se dalo zamezit dvěma způsoby, a to nalezením jiné knihovny, která by fungovala na více ovladačích pro Bluetooth nebo použitím jiného programovacího jazyka.

Na práci bych určitě chtěl v budoucnu pracovat dál. Vidím mnohé možnosti rozšíření pro mou aplikaci. Mohl bych napsat i klientskou část pro mobil, například v jazyce Java a vytvořit tak komplexní systém evidence a správy hovorů. Také bych mohl rozšířit možnosti komunikace přes Bluetooth, než jen posílání souborů. Další možností budoucího vývoje by mohlo být také zvýšení bezpečnosti pomocí autentizace uživatele nebo šifrování přenosu dat s hovory.



# Literatura

- [1] Puš, P. Poznáváme C# a Microsoft .NET. Dokument dostupný na URL:  
<http://www.zive.cz/default.aspx?section=44&author=405&fname=Petr+Pu%u0161>
- [2] Wikipedia The Free Encyklopedia, <http://en.wikipedia.org/>, květen 2008.
- [3] Microsoft Developer Network Library, <http://msdn2.microsoft.com/en-us/library>, květen 2008.
- [4] Webová stránka firmy In The Hand.Ltd., <http://inthehand.com/>, květen 2008.
- [5] Mobile Developer Wiki, <http://channel9.msdn.com/>, květen 2008.
- [6] Pocket PC Developer Network, <http://www.pocketpcdn.com/>, květen 2008.
- [7] The Code Project, <http://www.codeproject.com/>, květen 2008
- [8] Příklady zdrojových kódů v různých jazycích, <http://www.java2s.com/>, květen 2008.

## A. Obsah přiloženého CD

- soubor *readme.txt* – textový soubor obsahující informace o struktuře CD
- soubor *manual.pdf* – soubor s návodem k instalaci a použití
- soubor *xdvora77\_bp.pdf* – technická zpráva
- adresář *PDA\_EvidenceHovoru* - adresář obsahující zdrojové kódy aplikace