# VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra elektrotechniky a informatiky Obor Aplikovaná informatika

# Aplikace pro silniční a drážní váhu

bakalářská práce

Autor: Vít Vacata

Vedoucí práce: Mgr. Antonín Přibyl

Jihlava 2014



### ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce: Studijní program: Obor: Název práce: Cíl práce: Vít Vacata

Elektrotechnika a informatika Aplikovaná informatika

Aplikace pro silniční a drážní váhu

Student vytvoří windows desktopovou aplikaci pro silniční a drážní váhu s ukládáním dat do databáze. Program umožní evidenci hmotnosti projíždějících vozidel/vlaků včetně tisku vážních lístků. Student nejprve provede analýzu neveřejného komunikačního protokolu na sběrnici RS232, po níž jsou data předávána do PC. Program bude umožňovat i manuální vkládání dat v případě, že se nepodaří komunikaci dekódovat nebo program bude pracovat v offline režimu. Program bude vytvořen na základě stávajícího, již nevyhovujícího programu podle požadavků zadavatele. Aplikace bude dále umožnovat export dat a návrh tiskových sestav dle požadavků zadavatele.

Mgr. Antonín Příbyl vedoucí bakalářské práce

Ing. Bc. Michal Vopálenský, Ph.D. vedoucí katedry Katedra elektrotechniky a informatiky

# Abstrakt

Tato práce se zabývá vytvořením aplikace pro komunikaci po sériové lince RS-232 se silniční váhou pro zadavatele Agropodnik Jihlava, a.s. Tato aplikace má nahradit stávající zastaralé řešení. V práci je popsána stávající situace, komunikace po sériové lince pomocí jazyka C#, analýza dat pro navázání komunikace se silniční váhou a návrh a tvorba nové aplikace.

# Klíčová slova

C#, Sériová linka RS-232, LINQ to SQL, CSV, ARES

## Abstract

This thesis aims on creating an application for communication through serial port RS-232 with road weighbridges for the contracting authority Agropodnik Jihlava, a. s. This application is going to replace current obsolete solution. In this work current situation, communication through serial port with C#, analysis of data for establishing communication with road weighbridge and design and creation of new application are described.

# Keywords

C#, Serial port RS-232, LINQ to SQL, CSV, ARES

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též "**AZ**").

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ.

Byl/a jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje AZ, zejména § 60 (školní dílo).

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **souhlasím** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutí licence.

V Jihlavě dne

.....

Podpis

# Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu práce Mgr. Antonín Přibyl za poskytnutí zajímavého tématu a možnost ho vytvářet pod jeho vedením. Dále bych chtěl poděkovat kamarádům a rodině, kteří mě podporovali v mé práci.

# Obsah

| 1 | Úv   | od   | 8    |
|---|------|--|------|
| 2 | Teo  | pretický rozbor                                      | 9    |
|   | 2.1  | Silniční váha  | 9    |
|   | 2.2  | Mostové váhy   | 9    |
|   | 2.3  | Mobilní váhy   | 9    |
|   | 2.4  | Tenzometrický snímač                                 | 9    |
|   | 2.5  | Indikátor váhy PR 1612                               | . 10 |
|   | 2.6  | RS-232   | . 10 |
|   | 2.7  | Komunikace RS-232                                    | . 10 |
|   | 2.7  | .1 Zapojení RS-232                                   | . 12 |
|   | 2.8  | Virtuální simulace RS-232                            | . 14 |
|   | 2.9  | C# a .NET Framework                                  | . 15 |
|   | 2.9  | .1 Historie C#                                       | . 15 |
|   | 2.9  | .2 Rysy jazyka C#                                    | . 15 |
|   | 2.9  | .3 .NET Framework                                    | . 15 |
|   | 2.10 | Databáze   | . 16 |
|   | 2.11 | LINQ   | . 16 |
|   | 2.1  | 1.1 LINQ to SQL                                      | . 17 |
|   | 2.1  | 1.2 Výhody LINQ oproti SQL z pohledu vývoje aplikace | . 17 |
|   | 2.12 | SerialPort Terminal                                  | . 18 |
|   | 2.13 | Analýza čtení dat ze sériového portu v jazyku C#     | . 18 |
|   | 2.14 | Analýza kódu   | . 20 |
|   | 2.15 | Analýza stávajícího řešení                           | . 22 |
|   | 2.1  | 5.1 Funkční nedostatky                               | . 22 |
|   | 2.1  | 5.2 Design a uživatelské rozhraní                    | . 23 |

| 3  | Re                       | alizace programu24  | 4 |  |  |
|----|--------------------------|---|---|--|--|
|    | 3.1                      | Databáze24  | 4 |  |  |
|    | 3.2                      | Komunikace po sériové lince2                              | 6 |  |  |
|    | 3.3                      | Načítání dat pomocí služby ARES                           | 0 |  |  |
|    | 3.4                      | Export do CSV souboru                                     | 1 |  |  |
|    | 3.5                      | Tisk vybraného vážního lístku                             | 5 |  |  |
|    | 3.6                      | Uživatelské rozhraní                                      | 7 |  |  |
|    | 3.6                      | .1 Popis silniční váhy                                    | 8 |  |  |
|    | 3.6                      | .2 Adresář firem - popis                                  | 7 |  |  |
|    | 3.6                      | .3 Železniční váha4                                       | 8 |  |  |
| 4  | Zá                       | věr 4   | 9 |  |  |
| Se | eznam                    | použité literatury  | 0 |  |  |
| Se | eznam                    | obrázků   | 2 |  |  |
| Se | Seznam použitých zkratek |   |   |  |  |
| Pì | filohy.                  |   | 4 |  |  |
| 1  | Obsah přiloženého CD     |   |   |  |  |
| 2  | Už                       | Uživatelský manuál se nachází na přiloženém CD54          |   |  |  |
| 3  | Uk                       | Ukázka základního kódu v C# pro vytvoření Sériového portu |   |  |  |

# 1 Úvod

Společnost Agropodnik Jihlava, a. s. využívá dvě mostové váhy a jednu železniční váhu k vážení nákladů. Protože veškerá produkce této společnosti je k zákazníkům dopravována buď pomocí nákladních automobilů, nebo pomocí vlaků, je problematika vážení nákladu a uložení těchto hodnot do databáze zásadní, vždy je nutné rychle vytisknout dokumenty k zásilce a zaevidovat změny zásob.

Vážní most je tvořen železobetonovou modulární konstrukcí o délce 10m. Konstrukce mostu, která je navržena dle DIN 8119, je konstruována na vysoké zatížení. Snímání zatížení mostu je prováděno tenzometrickými snímači zatížení. [1] Tenzometrický snímač je připojen na dekodér, který převádí výstupní napětí z tohoto snímače do numericky vyjádřené váhy. Tento dekodér umí odesílat přes sběrnici RS-232 získaná data k dalšímu zpracování na dalších zařízení např. PC.

V současné době společnost využívá softwarové řešení od společnosti Lesyco, a. s., které je ale zastaralé a nevyhovující potřebám firmy. Mým úkolem je napsat aplikaci, která má současný software aktualizovat pro novější operační systémy, odstranit závady z nefunkční aplikace a v aktuálním programu a vylepšit a zpřehlednit uživatelské prostředí. Tento program budu vyvíjet v prostředí .NET Framework 4.5 a bude psaný v jazyce C#.

## 2 Teoretický rozbor

### 2.1 Silniční váha

Existuje několik typů silničních vah, například mostové váhy a mobilní váhy. V následujících odstavcích jsou blíže rozebrány tyto dva typy vah.

### 2.2 Mostové váhy

Mostové váhy jsou určené pro vážení vozidel a nákladu. Základem je ocelový nebo železobetonový vážní most. Mostové váhy se mohou instalovat ve dvou provedeních První provedení je zapuštěné do úrovně vozovky. Druhé provedení je nadúrovňovém - nájezdovém. V Agropodniku Jihlava, a. s. je využívána tato váha v zapuštěném provedení. [2]

#### 2.3 Mobilní váhy

Vývoj tenzometrických snímačů umožnil vyrobit toto tenké a lehké vážící zařízení. Vážící plošinky umožní rychlou kontrolu zatížení silničních vozidel. Tyto váhy dovolují vážení staticky (po nápravách) ale i za pohybu (přejezdem přes plošinky). [3]

### 2.4 Tenzometrický snímač

Než se tenzometry rozšířili v průmyslových aplikacích jako metoda vážení, byly velmi rozšířeny váhy mechanické. Mechanické váhy váží vše od tabletek až po kamiony a poskytují nám přesné a spolehlivé výsledky, ale pod podmínkou, že jsou řádně cejchovány a udržovány. V roce 1843 anglický fyzik Wheatstone vynalezl můstkové zapojení, které dokázalo změřit elektrický odpor. Tento Wheatstoneův můstek byl a stále je ideální pro měření změn odporu, jež nastávají v tenzometrech. [4]

Tenzometrický snímače síly jsou převodníky převádějící sílu, na měřitelný elektrický výstup. Existuje mnoho různých snímačů síly, ale tenzometrické snímače jsou stále nejrozšířenější. [4]

#### 2.5 Indikátor váhy PR 1612

Indikátor váhy PR 1612 je ukazatelem hmotnosti řízený mikropočítačem a je navržený hlavně pro průmyslové váhy. Tenzometr posílá hodnoty napětí do PR 1602, kde jsou pomocí mikropočítače zpracovány na BCD kód, který je pak zobrazen na displeji už jako aktuální váha. PR 1612 má i RS-232 výstup který může být zapojen do dalších zařízení, například PC. Tím se docílí, že se pošle aktuální váhu do aplikace na PC a může se s ní dále pracovat, například ukládat váhu do databáze atd. [5]

#### 2.6 RS-232

RS-232 je starší rozhraní pro přenos informací do vzdálenosti 20 m mezi dvěma zařízeními. Jestliže chceme větší odolnost proti rušení, musíme přenášet informaci větším napětím, než je standardních 5 V. Přenos dat probíhá asynchronně, pomocí pevně nastavené přenosové rychlosti známe také jako baud rate a synchronizujeme sestupnou hranou startovacího impulzu. [7]

Sériové rozhraní RS-232 v dnešní výbavě osobních počítačů těžko nalezneme. Pokud bychom chtěli používat sériový port, ale náš počítač už toto rozhraní nemá, můžeme použít spolehlivé USB-RS232 převodníky s integrovaným obvodem FT232 značky FTDI. Nevýhodou RS-232 rozhraní je nízká přenosová rychlost a topologie pouze typu bod-bod (účastníci komunikace mohou být pouze dva). Proto bylo vytlačeno novějšími, rychlejšími a s výhodnější stromovou topologií, jako je USB či FireWire (IEEE1394). Pro jednoduchost je RS-232 stále ve velké oblibě. Největší uplatnění se nachází v komunikaci mezi mikroprocesory nebo PLC automaty. [6]

#### 2.7 Komunikace RS-232

RS-232 používá dvě napěťové úrovně. Logickou 1 a 0. Log. 1 indikujeme zápornou úroveň, zatímco logickou 0 je přenášena kladná úroveň. [7] Pro komunikaci dvou zařízení přes RS-232 je potřeba nastavit pět různých parametrů přenosu.

Prvním z parametrů, které je u RS-232 potřeba nastavit je rychlost přenosu, tzv. baud rate. Baud je jednotka, kterou používáme pro měření rychlosti přenosu dat. *"Baud rate udává počet změn signálu za sekundu"*. [7]. Pokud nebude nastaven správný baud rate

dochází ke zkreslení přenášených hodnot. Druhým z parametrů, které musíme nastavit, je způsob zabezpečení dat. Nejjednodušším způsobem kontroly správnosti přenášených dat bez nároků na výpočetní výkon je parita. Ve vysílacím zařízení se sečte počet jedničkových bitů a doplní se paritním bitem tak, aby byla zachována předem dohodnutá podmínka sudého, nebo lichého počtu jedničkových bitů.

**Sudá parita** – Počet jedničkových bitů + paritní bit = Sudé číslo

Lichá parita – Počet jedničkových bitů + paritní bit = Liché číslo

**Space parity** – Tzv. nulová parita – paritní bit je vždy v log. 0, používáme například při komunikaci s 7bitového zařízení s 8bitovým, kdy paritní bit nahradíme log. 0 poslední bit v byte, tím zůstává zachovaná kompatibilita s 8bitovým přenosem.

**Mark parity** - Paritní bit se nastaví napevno na log. 1, při kompenzaci 7bitového provozu je potřeba jej na přijímací straně nulovat, jinak není zajištěna kompatibilita s ASCII.[9]

Třetím z parametrů je délka slova (data bits). Každý znak může být dlouhý 5 bitu (Baudotcode), 6 bitu (používáno pouze vzácně), 7 bitu (plná ASCII tabulka), 8 bitu (pro většinu dat v nových aplikacích, pro binární data) nebo 9 bitu (pouze vzácně). 5 nebo 7 bitu se používalo dříve u starších zařízení.[10] Pro potřeby mé aplikace budeme používat délku slova 7 bitu.

Čtvrtým parametrem je stop bit, kterým definujeme ukončení rámce. "Zároveň zajišťuje určitou prodlevu pro přijímač. Právě v době příjmu STOP bitu většina zařízení zpracovává přijatý BYTE".[11]

Posledním parametrem, který je potřeba nastavit je handshake, zajišťující potvrzení příjmu a zahájení přenosu na úrovni hardwarového nebo softwarového rozhraní.

Hardwarový handshaking zajišťuje přenos od vysílače k přijímači, čímž informuje, že vysílač má připravena platná data k odeslání. Zajišťuje také přenos informace od přijímače k vysílači o tom, že přijímač může začít data zpracovávat.

Softwarový handshaking probíhá na úrovni komunikačních protokolů (ZMODEM, KERMIT...) za pomocí běžného datového kanálu přijímač vysílači sdělí, zda je schopen data přijímat a zpracovávat data. Pro SW handshaking se používají znaky v ASCII tabulce XON/XOF. Je-li však potřeba v toku dat znaky XON/XOF vyslat, je nutné tak učinit speciální sekvencí znaků, která přenos dat značně zpomalí. [11]

### 2.7.1 Zapojení RS-232



Obrázek 1: Zapojeni RS-232[8]

| Zkratka | Název               | Funkce   |
|---------|---------------------|--|
| TD      | Transmit Data       | Serial Data Výstup (TXD)                               |
| RD      | Receive Data        | Serial Data Vstup (RXD)                                |
|         |                     | "Povolení k vysílání"; Logickou jedničkou na tomto     |
| CTS     | Clear to Send       | vstupu protistrana signalizuje, že DTE může vysílat    |
|         |                     | data   |
|         |                     | Logickou jedničkou na tomto vstupu protistrana         |
| DCD     | Data Carrier Datast | signalizuje, že detekovala na vedení nosný signál a    |
| DCD     | Data CamerDetect    | může komunikovat (DCE je např. modem na telefonní      |
|         |                     | lince)   |
|         |                     | Logickou jedničkou na tomto vstupu protistrana         |
| DSR     | Data Set Ready      | signalizuje, že je připravena (což neznamená že DTE    |
|         |                     | může okamžitě zaslat data, viz CTS)                    |
|         |                     | Logickou jedničkou na tomto výstupu DTE signalizuje    |
|         | Data Terminal       | protistraně svoji připravenost. Protistrana (např.     |
| DTR     | Ready               | modem) se tím aktivuje nebo zase deaktivuje. Modem     |
|         |                     | obvykle odpovídá nastavením DSR na logickou            |
|         |                     | jedničku.  |
|         |                     | "Požadavek na vysílání"; Logická jednička na tomto     |
|         |                     | výstupu signalizuje, že DTE chce vysílat data. Některé |
| RTS     | Request To Send     | převodníky RS232/RS485 tento signál používají pro      |
|         |                     | přepínání směru linky, což však vyžaduje, aby          |
|         |                     | software tento signál správně obsluhoval.              |
|         |                     | Logická jednička signalizuje do DTE příchozí hovor,    |
| RI      | Ring Indicator      | tedy že někdo požaduje datové spojení ("ring" je       |
|         |                     | anglicky "zvonit"; zvl. u telefonního modemu).         |

### Tabulka 1:BeyondLogic, zkraceno autorem.[12]

### 2.8 Virtuální simulace RS-232

Pro účely testování při absenci možnosti připojení pomocí RS-232 se využívají virtuální simulace sériového portu. Existuje mnoho různých programů na virtualizaci sériového portu RS-232, například Virtualserial port od firmy Eltima, Virtualserial port od firmy HWgroup a Virtualserial kit od firmy FabulaTech, který využívám.

Tento program umožňuje vytvoření dvojice seriál portů propojených virtuálním nullmodem kabel. Komunikace v tomto softwaru je prováděna shodně jako při fyzickém propojení mezi dvěma sériovými porty. Všechna data zapsaná na prvním COM1 portu budou okamžitě čtená na portu COM2 a obráceně. V programu jde vytvořit libovolné množství společných portů. [13]



Obrázek 2: Virtuální sériový port [14]

### 2.9 C# a .NET Framework

C# je programovací jazyk od firmy Microsoft a je navržený pro vytváření různorodých aplikací, které běží na rozhraní .NET Framework. "Jazyk C# je jednoduchý, výkonný, typově bezpečný a objektově orientovaný". Spousta inovací v jazyce C# umožňuje rapidně rychlý vývoj aplikací. Knihovna tříd .NET Framework poskytuje přístup k mnohá službám operačního systému windows a dalším užitečným třídám, které umožňují vývoj výrazně zjednodušit a urychlit.[15]

#### 2.9.1 Historie C#

C# 1.0 - První verze byla vydaná v roce 2002 s .NET Frameworkem 1.0 obsahovala pouze základní podporu objektového programování, které vycházela z jiných jazyků například C++ nebo java. [16]

C# 2.0 – Vyšla v roce 2005. "Mezi nové vlastnosti patří nativní podpora generik, částečné a statické třídy, iterátory, anonymní metody pro pohodlnější užívání delegátů (odkazů na metody), nullovatelné hodnotové typy a operátor koalescence". [16]

C# 3.0 - Koncem roku 2007 byl vydán společně s .NET Frameworkem 3.5 a Visual Studiem 2008. Obsahoval poměrně revoluční změny. [16]

C# 4.0 - Duben 2010. Nová verze se zaměřuje hlavně na dynamické aspekty programování a frameworky. [16]

C# 5.0 - Verze 5.0 byla uvedena v srpnu 2012 společně s .NET Framework 4.5 a vývojovým prostředím Visual Studio 2012.[16]

#### 2.9.2 Rysy jazyka C#

Jazyk C# obsahuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní. Podporuje členská data a metody, přidává vlastnosti a události. Správa paměti je automatická, o dealokaci objektů atd. se stará *garbagecollector*, podporuje zpracování chyb pomocí výjimek. Je case-sensitive (rozlišuje velká a malá písmena). [17]

#### 2.9.3 .NET Framework

".NET Framework je technologie která podporuje budování a spouštění next-gen aplikacích a XML webových služeb." [18]

".Net je integrovanou součástí ve spoustě aplikací běžících pod systémem Windows a poskytuje funkcionalitu pro ty to aplikace a jejich spuštění. Pro vývojáře poskytuje komplexní a konzistentní programovací model pro vytváření aplikací" [19]

#### 2.9.3.1 .NET Framework se skládá ze čtyř komponent

Jazyk, Microsoft Visual Studio, Virtuální stroj a Knihovny.

#### 2.10 Databáze

Databáze je nástroj pro shromažďování a uspořádání informací. Do databází lze ukládat informace o osobách, produktech, objednávkách nebo čemkoli jiném. [20] Skládá se z tabulek.

Relační databázový model má jednoduchou strukturu, data jsou organizována v tabulkách, které se skládají z řádků a sloupců. V těchto tabulkách jsou prováděny všechny databázové operace. [21] Relační databázi využívám i ve své aplikaci



Obrázek 3: Relační model databáze[22]

## 2.11 LINQ

"LINQ (anglicky LanguageIntegratedQuery) přináší nový způsob pro dotazování nad jakýmikoliv daty, usnadňuje jejich tvorbu, třídění, jejich propojování i vyhledávání v nich".[23]

### 2.11.1 LINQ to SQL

Umožní dotazování nad databázemi využívající rozhraní MS SQL. Příkazy LINQu se mapují na odpovídající příkazy SQL. Protože jsou data v těchto databázích uložena jako relační, musí být nasazen tzv. mapper těchto dat, který je převede na objektová data, která používá LINQ. Výhodou LINQ je objektový pohled na data. [23]

"Dotazy se ukládají do proměnné typu var. Klíčové slovo var umožňuje přenechat výběr datového typu na kompileru (C# ho za nás sám přiřadí při překladu). "[24]

### 2.11.2 Výhody LINQ oproti SQL z pohledu vývoje aplikace

V LINQ se nepoužívají string řetězce pro vložení SQL příkazu, ale využíváme C# zápis, díky kterému je psaní kódu výrazně jednoduší. Výhodou LINQ je také jednodušší vyhýbání se syntaktickým chybám díky neustálým kontrolám kompilátoru a nástroje Intellisense, který funguje také jako nápověda, kdy nabízí různé možnosti dokončení kódu.

SQL zápis je náchylný na syntaktické chyby, které nezvýrazní Intellisense, může být proto problém případné chyby následně odhalit. Pokud bychom například v následující ukázce kódu omylem napsali místo SELECT jenom SELET, tak nás na to vývojové prostředí neupozorní a kód půjde zkompilovat, ale když spustíme v programu část kódu s tímto chybným zápisem SQL dotazu, neošetřený program spadne.

Ukázka SQL zápisu

string prikaz = "SELECT \* FROM auta WHERE id = 1";

Vybere všechny data z tabulky auta na řádku, který má id 1.

LINQ zápis oproti SQL má několik nástrojů, díky kterým se lze podobným chybám vyhnout. Chyby v syntaktickém zápisu jsou ihned vypisovány ve vývojovém prostředí, což je výhodou oproti zápisu SQL ve formě řetězce. Nástroj Intellisense také pomáhá nabízením dostupných funkcí a proměnných, kterými lze dokončit kód. Další výhodou je reakce na změnu databáze, kdy nám program nahlásí chybu, pokud se databáze změní. Nevýhodou je pouze zanedbatelné snížení výkonu.

Ukázka LINQ zápisu

```
var dotaz = (from a in auta
    where a.id = 1
    select a).SingleOrDefault();
```

Vybere všechny data z tabulky auta na řádku, který má id 1.

### 2.12 SerialPort Terminal

| 税 SerialPort Terminal  |                             |
|--|-----------------------------|
| Application Started at 10/19/2009 4:22:36 PM                       |                             |
|  |                             |
| Send Data: Send  | Clear                       |
| COM Serial Port Settings   | Data Mode                   |
| COM Port: Baud Rate: Parity: Data Bits: Stop Bits:                 | O Text                      |
| COM1 ▼ 2400 ▼ None ▼ 8 ▼ One ▼                                     | ⊘ Hex                       |
| Line Signals<br>DTR RTS CTS DSR CD Clear on Oper<br>Clear with DTI | Open Port<br>R <u>About</u> |

Obrázek 4: SerialPort Terminal prostředí aplikace

Tato aplikace [25] slouží k testování sériového portu RS-232, jedná se o licenci opensource, takže kód je volně dostupný a otevřený k úpravám. Slouží také jako pomocný výukový kód pro naprogramování aplikací, které využívají sériový port RS-232. Jak je vidět na obrázku, aplikace má všechna potřebná nastavení pro konfiguraci sériového portu RS-232. Umí přijímat i odesílat data v modu jak string, tak i text a nebo kód v šestnáctkové soustavě (hex).

### 2.13 Analýza čtení dat ze sériového portu v jazyku C#

V příloze této práce naleznete celý kód. Zde jsem rozebral pouze jeho hlavní části.

```
using System.IO.Ports;
```

Abychom mohli pracovat s porty v C# musíme použít knihovnu System.IO.Ports

```
private SerialPort port = new SerialPort("COM1", 9600, Parity.None, 8,
StopBits.One);
```

Jednoduché vytvoření portu s názvem SerialPort a jeho základním nastavením od názvu až po stop bit, nebo pro pokročilejší aplikaci využívající windowsform použijeme ovládací prvky jako comboBox atd.

Kód může vypadat například takto:

```
port.DataReceived +=
new SerialDataReceivedEventHandler(port_DataReceived);
```

Tato metoda je zavolána pokud čekají nějaká data v portbufferu

Pokud je vše nastaveno správně tak se port otevře jinak je vyvolaná výjimka, a zavolá se messageBox.show který zobrazí informaci o chybě

```
{
    Console.WriteLine(port.ReadExisting());
}
```

Tato metoda slouží na výpis aktuálních dat z bufferu portu. Nemusíme vypisovat data jenom do konzole, ale můžeme využít windowsform a vypisovat například do textBoxu, listBoxu, nebo richBoxu.

## 2.14 Analýza kódu

Při analýze kódu jsem nejprve připojil notebook k indikátoru váhy PR 1612 přes sériový port RS-232. Poté jsem spustil aplikaci SerialPort Terminal, která vyžadovala konfiguraci. Pro konfiguraci komunikace RS-232 bylo zapotřebí zjistit baud rate, délku slova, počet stop bitu, paritu a handshake. Nejdříve byla konfigurace přejata z nastavení portu ve Windows 98, ale po aplikaci tohoto nastavení jsem otevřel port a v okně se začaly objevovat nesmyslné znaky, proto jsem přepnul data mode v programu z text na hex a hodnoty indikátoru váhy PR 1612 již vypadaly smysluplně. Dále jsem provedl sérii měření od nuly až do jedenácti tun. Všechny naměřené hodnoty jsem uložil a seřadil podle váhy a začal hledat algoritmus kódu. Tento kód se ale bohužel nepodobal kódu popsanému v datasheetu PR 1612 [6], mj. se měnila jeho délka.

Struktura datového přenosového řetězce z PR 1612

| STX N | MODE | STATUS | WEIGHT | ] | ETX             |
|-------|------|--------|--------|---|-----------------|
| 1 1   | 1    | + 1    | 6      | 1 | počet znaků [6] |

Ve stávajícím programu vypadá tento kód podobně, jen je z hex převeden na ASCII.

| 048128 0012048128 001204<br>Ruční vstup 🔽 Filtr firem 🗙 |      |   |  |  |  |
|---|------|---|--|--|--|
| PR 1612   |      |   |  |  |  |
| 000 0 <b>00 000</b>                                     |      |   |  |  |  |
|   |      |   |  |  |  |
|   |      |   |  |  |  |
|   | Levā | Р |  |  |  |

Obrázek 5: Komunikace v původní aplikaci

Při simulaci na virtuálním počítači a pomocí programu SerialPort Terminal jsem se snažil nasimulovat data tak, jak je popsáno v dokumentaci. Simulace se podařila a stávající aplikace začala zobrazovat smysluplné údaje, jako kdyby byla připojená fyzicky k váze. Proto jsem myslel, že už jsem našel řešení a napsal jsem funkce, které dokážou upravit tento kód a zobrazit pouze váhu. Ačkoliv v simulacích tento program fungoval správně, po připojení k váze program začal ukazovat nesmyslné znaky. Při své další návštěvě v Agropodniku Jihlava, a. s., jsem se znovu snažil přijít na způsob šifrování dat. Po připojení k váze jsem spustil SerialPort Terminal a začal jsem znovu zachytávat komunikaci, ale tentokrát jsem zkoušel různě měnit nastavení baud rate, stop bit, paritu a délku slova. V nastavení datového módu na hex jsem vyzkoušel spousty kombinací, ale stále jsem se nedostával požadovaná data, tak jsem nakonec vrátil výstup zpátky na text a opět začal měnit nastavení. Jediné nastavení, které umožňuje správnou komunikaci mezi váhou a PC, je baudrate 4800, parita na none, délka slova 7 a stop bit one. Při využití popsaného nastavení se konečně zobrazí požadovaný datový string tak, jak je popsán v datasheetu PR 1612.

| SerialPort Terminal  |  |
|--|--|
| $\begin{array}{ c c c c c c c c c c c c c c c c c c c$   | 00120 4128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128<br>04128 |
| Send Data: Send  | <u>C</u> lear  |
| COM Serial Port Settings   | Data <u>M</u> ode  |
| COM Port:Baud Rate:Parity:Data Bits:Stop Bits:COM34800None7One                                 | ● Text ○ Hex   |
| Line Signals       DTR     RTS     CTS     DSR     CD       Clear on Open       Clear with DTR | <u>C</u> lose Port<br><u>About</u>   |

**Obrázek 6: Komunikace v SerialPort Terminal** 

Díky rozšifrování komunikace indikátoru váhy PR 1612 jsem mohl napsat funkce pracující s přijatými daty.

### 2.15 Analýza stávajícího řešení

V současné době používají k zpracování dat ze silniční a drážní váhy program Lesyco. Tento program je ale zastaralý a nevyhovující ke každodenní práci. Největším problémem je nefunkčnost na novějších operačních systémech Windows XP, Windows Vista, Windows 7 a Windows 8, proto je pro používání programu na novějších počítačích nutné používat virtualizované prostředí Windows 98.

#### 2.15.1 Funkční nedostatky

Program obsahuje spoustu funkčních nedostatků nebo funkcí které uživatelé vůbec nevyužívají. Jeden z hlavních funkčních nedostatků je možnost přepisovat kdekoliv data přímo v tabulce bez jakékoliv programové kontroly. Mohu tedy vkládat data ve špatném formátu, např. do buňky s IČ společnosti mohu vložit text a program nám to bez problému umožní. Další závažná chyba je možnost měnit identifikační číslo vážního lístku. Toto číslo by mělo sloužit k jednoznačné identifikaci dokumentu, a proto by v žádném případě neměla být umožněna jeho změna či přepsání. Program má také jasně definovaný postup akcí při vytváření nové vážního lístku. Pokud uživatel tento proces nedodrží, program se začne chovat nekonzistentně, je nestabilní a například zobrazuje nesprávně popsaná varovná hlášení, která pro uživatele nemají žádnou informační hodnotu.

Firma také potřebuje každý měsíc vytvářet sestavy vážných záznamů. Ve stávajícím programu je však export dat nefunkční. Obsluha programu tedy musí vytvářet sestavy ručně v tabulkovém kalkulátoru, což je velmi časově náročné a může to vést k chybám při přepisování dat. Obsluha programu si také stěžuje na nemožnost přidat nový záznam do firemního adresáře, je-li rozpracováno přidávání nového vážního lístku. Tato nefunkčnost se projeví při zakládání nového vážního lísku pro firmu, která dosud není v adresáři, a je tedy nutné jí nejprve přidat do adresáře, a teprve poté se zobrazí ve výběru firem u vážního lístku. Proto obsluha musí ukončit rozpracovaný vážní lístek, přidat firmu do adresáře a poté opět založit nový vážný lístek. Jelikož program nepodporuje nové operační systémy, je nutné využívat zastaralou jehličkovou tiskárnu a není možné připojení nových laserových tiskáren, protože ovladače nových tiskáren nepodporují zpětnou kompatibilitu s Windows 98.

### 2.15.2 Design a uživatelské rozhraní

Stáří programu je vidět hned na první pohled, díky zastaralému designu z Windows 98. Design je navržen pro nízké rozlišení monitoru. Proto všechny důležité ovládací prvky a formuláře jsou nepřehledně vtěsnány do toho to malého prostoru. Například prostor pro zobrazování uložených záznamu je velice malý, nepřehledný a špatně čitelný. Z tohoto důvodu je nutné používat posuvník, abychom mohli zobrazit všechny hodnoty ve vybraném řádku. Přehlednost se nezlepší ani při maximalizaci okna, jak bychom očekávali, protože rozložení prvků zůstane na svých místech a na maximalizaci okna nijak nereaguje.

Při mém prvním spuštění a testování aplikace, jsem se nemohl zorientovat v prostředí programu. Ovládací prvky jsou umístěny nepřehledně a popsány tak že nezkušený uživatel se v programu lehce ztratí. Bez instrukcí zkušené obsluhy jsem ani nedokázal vytvořit testovací vážní lístek, a provést běžné úkony, které by měl program umožňovat.

| PC_vaha-AGP - VMware Player (Non-commercial use only) |                                 |                                 |             | ×   |
|---|---------------------------------|---------------------------------|-------------|---|
| Player 👻 🔛 🖛 🔚  |                                 |                                 |             |   |
| 🎢 LESYCO - vážírna silniční váhy                      |                                 |                                 |             | _ <u>-</u> |
| Vážení Adresář Ceníky Želva Sestavy Nastav            | vení Kredity Kon.Příjemce Dokla | ady Export/import Fakturace     |             |   |
| ⊲⇒ Příjezd ⊲⇒ Odjezd ŽelVa ⊠ ⊲ D                      | 🖂 🗠 🖻 🖹 Tisk VL 🗖 🖡             | Buční vstup 🔲 Filtr firem 🛛 🗙 🗌 | -           |   |
| Vážní záznam  | NÍ                              | R 1612                          |             |   |
| Doklad 09179 VL taktura                               | a 17.5.2014                     | មិន មិន មិន មិន មិន             |             |   |
| Faktura   | ^ I                             | E E E                           |             |   |
| SPZ BRU   | по                              |                                 |             |   |
| Čst NET   | TO 15:19:54                     | L Levá P                        |             |   |
| Druh Cena   | DPH S                           | EQ WG1 WG2 DATE2                | <b>_</b>    |   |
| Místo Dopri   |                                 |                                 |             |   |
| určení  |                                 |                                 |             |   |
| Doprava Celke   | em                              |                                 |             |   |
| Řidič Druh 💽 OK                                       | příjezd Zruš příjezd 🚽          |                                 | <b>▼</b> 1  |   |
|   |                                 |                                 |             |   |
| Doklad SPZ Číslo-IČO DRUH T                           | ARA NETT BRUTO Cena DPH         | Faktura Místo určer Řidič [     | ). <u> </u> |   |
| 93861   |                                 |                                 | _           |   |
| 93862   |                                 |                                 |             |   |
| 93864   |                                 |                                 |             |   |
| × 69179   |                                 |                                 | -           |   |
|   |                                 | <u>1</u>                        | ſ           |   |
|   |                                 |                                 |             |   |
|   |                                 |                                 |             |   |
|   |                                 |                                 |             |   |
| 🙀 Start 💾 🕼 🙍 🧑 Vazenissp                             |                                 |                                 |             | <br>C2 😥 😳 16:33  |
| 🚱 🧿 嶺 PC_vaha-AGP - V                                 |                                 |                                 |             | CS 🔺 🔐 📴 16:33<br>17.5.2014   |

Obrázek 7: Stávající program s maximalizovaným oknem

## 3 Realizace programu

#### 3.1 Databáze

V programu využívám Microsoft SQL server a databázi typu MDF, se kterou mám největší zkušenosti. Vytvořil jsem jednoduchou databázi, která je uložena v souboru AGP\_Databaze.mdf. Tato databáze je nezbytná pro fungování celé mé aplikace, neboť do ní ukládám všechny údaje, například vážní záznamy ze silniční a železniční váhy a také adresář firem, pro které firma váží.



Obrázek 8: Databázový model

Databáze obsahuje tři tabulky. Tabulku dbDoklad, dbZelva a dbFirma. V tabulce dbDoklad ukládám všechna data o vážních lístcích ze silniční váhy. V tabulce dbZelva jsou uložena data o lístcích z železniční váhy a v tabulce dbFirma jsou uloženy záznamy o firmách v adresáři.

| Název           | Datový typ   | Popis                                   | Název       | Datový typ   | Popis                                   |
|-----------------|--------------|---|-------------|--------------|---|
| id              | int          | číslo lístku –<br>primární klíč         | id          | int          | číslo lístku –<br>primární klíč         |
| cisloVagon<br>u | nvarchar(50) | spz                                     | cisloVagonu | nvarchar(50) | Číslo vagónu                            |
| spzPrives       | nvarchar(50) | spz přívěsu                             | idIco       | int          | cizí klíč z<br>tabulky<br>dbFirma k ico |
| idlco           | int          | cizí klíč z<br>tabulky<br>dbFirma k ico | druh        | nvarchar(50) | druh zboží co<br>se váží                |
| druh            | nvarchar(50) | druh zboží co<br>se váží                | tara        | int          | prazdný<br>kamion                       |
| tara            | int          | prazdný<br>kamion                       | bruto       | int          | naložený<br>kamion                      |
| bruto           | int          | naložený<br>kamion                      | netto       | int          | čistá hmotnost                          |
| netto           | int          | čistá hmotnost                          | mistoUrceni | nvarchar(50) | místo určení                            |
| mistoUrcen<br>i | nvarchar(50) | místo určení                            | Poznámka    | text         | jméno řidiče                            |
| ridic           | nvarchar(50) | jméno řidiče                            |             |              |   |
| datum           | DateTime     | datum uložení<br>dokladu                |             |              |   |
| poznamka        | text         | poznámka<br>obsluhy                     |             |              |   |

#### Tabulka 2: dbDoklad a dbZelva

|            |              | -                           |
|------------|--------------|-----------------------------|
| Název      | Datový typ   | Popis                       |
| id         | int          | ičo firmy – primární klíč   |
| nazev      | nvarchar(50) | název firmy                 |
| ulice      | nvarchar(50) | název firmy kde firma sídlí |
| cisloDomu  | int          | číslo domu firmy            |
| město      | nvarchar(50) | město ve které firma sídlí  |
| psc        | int          | psč kde firma sídlí         |
| dic        | nvarchar(50) | dič firmy                   |
| insolvence | bool         | firma v insolvenci ano/ne   |
| platceDph  | bool         | firma platí dph ano/ne      |

Vztah mezi tabulky dbDoklad a dbFirma je 1:N to znamená, že jedna firma může mít víc dokladů ale jeden doklad pouze jednu firmu. Tento vztah 1:N platí i u dbZelva a dbFirma.

Ke komunikaci s databází využívám LINQ to SQL. Visual studio automaticky vytvoří přístup k databázi. Pak stačí už jen volat předefinované funkce LINQ to SQL. Pro připojení k databázi využívám tento connectionstring:

```
"DataSource=(LocalDB)\\v11.0;AttachDbFilename=|DataDirectory|\\AGP_Dat
abaze.mdf;IntegratedSecurity=True;PersistSecurityInfo=False".
Pro zobrazení aktuálních dat z databáze dbDoklad využijeme funkci
```

```
public static Table<dbDoklad>GetDokladDB()
{
    DataClassesAGPDataContextdb = newDataClassesAGPDataContext();
    returndb.GetTable<dbDoklad>();
```

}

Tato funkce vytvoří proměnou db, zavoláním konstruktoru z třídy DataClassesAGPDataContext. Tato třída je automaticky generována podle tabulek, které vložíme do data modelu LINQ to SQL. Poté můžeme přistupovat k funkcím třídy DataClassesAGPDataContext.

Funkce z třídy DataClassesAGPDataContextGetTable<dbDoklad>() vrací obsah z tabulky dbDoklad. Tato funkce je stejná jako SQL dotaz "Select \* From dbDoklad". Změnou názvu uvnitř < > měníme tabulku, kterou chceme zobrazit, například GetTable<dbFirma>. Tato funkce vrátí data, která zobrazíme v položce DataGridView.

### 3.2 Komunikace po sériové lince

Silniční váha, ze které potřebujeme získat údaje o váze vozidla, komunikuje po sériové lince. Program by tuto komunikaci měl umožňovat. Pro komunikaci využíváme třídu SerialPort, která obsahuje všechny metody nutné pro správnou komunikaci. Před samotným zahájením komunikace je nutné nastavit vlastnosti sériového portu. K tomuto nastavení je v programu určen formulář Form\_Nastaveni.

| Nastaveni Serial Port1       Nastaveni Serial Port2         COM Port:       Data Bits:         Image: Stop Bits:       Image: Stop Bits:         Image: Parita:       Handshake:         Image: Parita:       Handshake:         Image: Parita:       Handshake:         Image: Parita:       Mandshake:         Image: Parita:       Mandshake: | 🚽 Nastavení            |                        | <b>—</b> ——————————————————————————————————— |
|--|------------------------|------------------------|--|
| COM Port:       Data Bits:         COM Port:       Data Bits:         Baud Rate:       Stop Bits:         Parita:       Handshake:         Parita:       Handshake:         OK   | Nastaveni Serial Port1 | Nastaveni Serial Port2 | Rozě (řený režim Admin                       |
| Baud Rate:       Stop Bits:         T       T       T         Parita:       Handshake:       Parita:       Handshake:         T       T       T       OK       Stomo   | COM Port: Data Bits:   | COM Port: Data Bits:   |  |
| Parita: Handshake: Parita: Handshake:  | Baud Rate: Stop Bits:  | Baud Rate: Stop Bits:  |  |
|  | Parita: Handshake      | Parita: Handshake:     | OK Stomo                                     |

Obrázek 9: Formulář nastavení

Formulář nastavení umožňuje snadné nastavení obou sériových portů. Jestliže nastavení neproběhne pomocí našeho formuláře, jsou pro nastavení použity defaultní hodnoty. Po nastavení jsou hodnoty uloženy do konfiguračního souboru, z kterého se načítají při dalším spuštění. Program si tedy pamatuje nastavení i při dalším spuštění.

| Název     | Тур                       | Defaultní Hodnota | Popis               |
|-----------|---------------------------|-------------------|---------------------|
| PortName  | string                    | COM1              | název portu         |
| BaudRate  | int                       | 9600              | rychlost            |
| DataBits  | int                       | 7                 | počet datových bitu |
| Parity    | System.IO.Ports.Parity    | None              | příznak parity      |
| StopBits  | System.IO.Ports.StopBits  | One               | stop bit            |
| HandShake | Systém.IO.Ports.HandShake | false             | potvrzení přijmu    |

#### Tabulka 4: Tabulka nastavení komunikace

K vytváření konfiguračního souboru a nastavení vlastností portu používám třídu Settings.

```
private Settings nastaveni = Settings.Default; //přiřádím do proměnné
nastavení defaultní hodnoty z settings
private void UlozeniNastaveniPort1()
{
    nastaveni.PortName1 = cmbNazevPort1.Text;
    nastaveni.BaudRate1 = int.Parse(cmbBaudRate1.Text);
    nastaveni.Parity1 = (Parity)Enum.Parse(typeof(Parity),
    cmbParita1.Text);
    nastaveni.DataBits1 = int.Parse(cmbDataBits1.Text);
    nastaveni.StopBits1 = (StopBits)Enum.Parse(typeof(StopBits),
    cmbStopBits1.Text);
    nastaveni.HandShake1 = (Handshake)Enum.Parse(typeof(Handshake),
    cmbHandShake1.Text);
}
```

Zde na funkci UlozeniNastaveniPort1() můžeme vidět, jak probíhá ukládání nastavených hodnot uživatelem.

Po nastavení nezbytných parametrů se přiřadí hodnoty z proměnných nastavení do proměnné serialPort která reprezentuje třídu SerialPort, poté se otevře komunikace. Pokud se otevření komunikace podaří, program naslouchá nastavené sériové lince.

Přijdou-li data na nastavený sériový port je vyvolána událost SerialDataReceivedEventHandler, která zavolá funkci CteniDataPort1(), jenž zajišťuje zpracování přijatých dat.

```
private void CteniDataPort1(objectsender, SerialDataReceivedEventArgs
e)
{
      SerialPortsp = (SerialPort)sender;
      if (!sp.IsOpen)
      {
            TxtBoxTestPort("", false);
            return;
      }
      string data = sp.ReadExisting();
      if (data.Length<= 11)</pre>
      {
            try
            {
                   TxtBoxUkazatel(Prevod.UpravaDatVaha(data));
                  TxtBoxTestPort(data, true);
            }
            catch
            { }
      }
 }
```

Do proměnné sp se uloží sériový port, ze kterého byla událost vyvolána. Pokud port není otevřen, funkce končí. Jestliže je port otevřen data přečteme, a pokud mají požadovanou délku, jako je popsáno v datasheetu PR1612, tak se pošlou ke zpracování.

Zpracování dat zajišťuje třída Prevod a její metoda UpravaDatVaha.

```
public static string UpravaDatVaha(string data)
{
      string dataNew = "";
      try
      {
            dataNew = data.Substring(4, 5);
            int i = 0;
            while (dataNew.Substring(0, 1) == "0")
            {
                  if (i < 4)
                    {
                        dataNew = dataNew.Remove(0, 1);
                        i++;
                    }
                  else
                  {
                        break;
                  }
            }
      }
      catch
       {
      }
      returndataNew;
}
```

Příchozí data jsou v tomto formátu o této délce.

[]12800020[].

Pomocí funkce Substring s parametry (4,5) vyextrahujeme řetězec, který má index 4 a je dlouhý 5 znaků.

00020.

Po vyextrahování váhy, v cyklu odřízneme přebytečné nuly a funkce vrátí váhu v požadovaném formátu.

20.

Váha v požadovaném formátu je předaná funkciTxtBoxUkazatel. Tato funkce pouze zobrazí váhu uživateli pomocí metody invoke. Metoda invoke zajistí zobrazení dat z jiného vlákna do aktuálního formuláře.

```
private void TxtBoxUkazatel(string data)
{
    txtUkazatelVahyDoklad.Invoke(newEventHandler(delegate
    {
        txtUkazatelVahyDoklad.Text = data;
    }));
}
```

### 3.3 Načítání dat pomocí služby ARES

Administrativní registr ekonomických subjektů (ARES) je informační systém, umožňující vyhledávaní ekonomických subjektů registrovaných v České republice a zobrazení údajů o nich. [26]

V programu používáme ARES k načítání údajů o firmě pomocí identifikačního čísla (IČO). Pro načítání dat z registru ARES jsem mohl vybírat ze tří možností realizace. První možnost byla stažení příslušného záznamu podle zadaného identifikačního čísla (IČO) ve formátu XML, který bychom museli procházet a vyhledávat požadované informace. Toto řešení mi ale nepřišlo dost efektivní, proto jsem se rozhodl vybrat jiné řešení. Druhá možnost byla instalace nuget balíčku AresWebService [27] který zajišťuje komunikaci s registrem. Jelikož jsem s balíčky nuget doposud nepracoval, a nenalezl jsem vhodný příklad jeho fungování, rozhodl jsem se pro třetí možnost. Třetí možnost je využívání webové služby. [28] která umožní stáhnutí potřebných tříd pro přístup do registru ARES a vyhledávání v něm. Tuto možnost jsem zvolil, protože jsem našel vhodný ukázkový příklad pro práci s webovou službou ARES. [29]

Protože vyhledávání v registru ARES může trvat i několik vteřin podle toho, jak jsou zatíženy servery, spouštíme vyhledávání v samostatném vlákně. To nám zajistí, že nedojde k zastavení programu během hledání v registru, a obsluha může pracovat bez přerušení. Uživatel je informován o průběhu vyhledávání pomocí ukazatele průběhu (progress bar). Po ukončení vyhledávání je uživatel informován v dialogovém okně o jeho dokončení a přesměrován do předvyplněného formuláře. Na požadavek zadavatele také zjišťujeme, zda je firma v insolvenčním řízení a jestli je plátcem DPH.

### 3.4 Export do CSV souboru

Další požadavek byl od zadavatele, aby program uměl export z databáze do tabulkového editoru. Pro export jsem měl na výběr ze tří možností zápisu. První možnost byla přidat odkaz na Microsoft Excel 12.0 ObjectLibrary a pak využívat její metody pro zápis do formátu xls. Bohužel tato možnost je dostupná pouze v případě, že na počítači, kde je program vyvíjen, byl naistalován balíček Microsoft Office. Druhá možnost byla použít aplikaci třetích stran pro export do souboru formátu xls. Tento formát je správně načítán pouze v Microsoft Excel. Ostatní tabulkové editory neumožnují bezchybný import. Jelikož zadavatel využívá OpenOffice, rozhodl jsem se pro třetí možnost, což je zápis do souboru csv, který bezchybně otevře jak Microsoft Excel tak i OpenOffice. Tento soubor má jednoduchý styl zápisu, skládá se z dat, které jsou odděleny středníkem.

| NEVYBRÁNO_TEST_19.05.2014_19.05.2014.csv – Poznámkový blok   |                       | x             |
|--|-----------------------|---------------|
| Soubor Úpravy Formát Zobrazení Nápověda  |                       |               |
| Čislo dokladu;Datum;Materiál;Místo určení;Váha;SPZ;Řidič;POZI<br>5;19.5.2014 1:53:55;TEST;TEST;900;SPZ TEST;TEST;POZNAMKA TES<br>6;19.5.2014 1:54:31;TEST1;TEST1;18000;TEST1;TEST1;POZNAMKA TI | námka;<br>T;<br>EST1; | *             |
| 4  |                       |               |
| Řádek  | : 4, Sloupec J        | 1 <sub></sub> |

Obrázek 10: Ukázka zápisu CSV v poznámkovém bloku

|            |       | ) - (2 -  | ∓          |                      |                |                         |                            |           |         |             | AHOJ_13.05 | .2014_14    | 4.05.2014 | 4.csv - M | icrosoft E        | xcel    |        |
|------------|-------|-----------|------------|----------------------|----------------|-------------------------|----------------------------|-----------|---------|-------------|------------|-------------|-----------|-----------|-------------------|---------|--------|
| Sou        | ıbor  | Domů      | V          | 'ložení F            | Rozložení str  | ánky Vzorce             | e Da                       | ita Re    | vize    | Zobrazení   | Load Tes   | t Tý        | 'n        |           |                   |         |        |
| *          | A     |           | *          |                      |                | 0                       | <b>Připojer</b><br>Vlastno | ní<br>sti |         | Y           | 🐼 Vymaza   | it<br>znovu | ÷         | → → →     | 3                 |         | Ê      |
| Z ap<br>Ac | cess  | Z<br>webu | ۲<br>textu | Z jiných<br>zdrojů ∗ | Existujíci í   | Aktualizovat<br>vše 🔹 📟 | Upravit                    | odkazy    | A Serad | it Filtr    | ゾ Upřesn   | it          | sloupci   | duplicity | overeni<br>∕dat ≠ | Sloucit | analý: |
|            |       | Načís     | t exter    | ní data              |                | Připo                   | jení                       |           |         | Seřadit a f | iltrovat   |             |           | Da        | itové nást        | roje    |        |
|            |       | J4        |            | <b>-</b> (0          | f <sub>x</sub> |                         |                            |           |         |             |            |             |           |           |                   |         |        |
|            |       | А         |            | В                    | С              | D                       | E                          | F         | G       |             | н          | L.          |           | J         | К                 |         | L      |
| 1          | Čislo | dokladı   | J Dat      | um                   | Materiál       | Místo určení            | Váha                       | SPZ       | Řidič   | Poznámi     | (a         |             |           |           |                   |         |        |
| 2          |       |           | 5 19.      | 5.2014 1:53          | 3 TEST         | TEST                    | 900                        | SPZ TEST  | TEST    | POZNAN      | IKA TEST   |             |           |           |                   |         |        |
| 3          |       |           | 6 19.      | 5.2014 1:54          | TEST1          | TEST1                   | 18000                      | TEST1     | TEST1   | POZNAN      | IKA TEST1  |             |           |           |                   |         |        |
| 4          |       |           |            |                      |                |                         |                            |           |         |             |            |             |           |           |                   |         |        |
| 5          |       |           |            |                      |                |                         |                            |           |         |             |            |             |           |           |                   |         |        |
| 6          |       |           |            |                      |                |                         |                            |           |         |             |            |             |           |           |                   |         |        |
| 7          |       |           |            |                      |                |                         |                            |           |         |             |            |             |           |           |                   |         |        |

Obrázek 11: Ukázka dat v CSV formátu Microsoft Excel

|     | 24    | ) • (° • | . ≛     |            |               |              |                            | -         |            | -   |             | AHOJ_13.05 | .2014_1     | 4.05.201 | l4.csv - M  | icrosoft E | xcel    |        |
|-----|-------|----------|---------|------------|---------------|--------------|----------------------------|-----------|------------|-----|-------------|------------|-------------|----------|-------------|------------|---------|--------|
| Sou | ıbor  | Domů     | ٧       | ložení l   | Rozložení sti | ránky Vzorce | e Da                       | ata R     | evize      | Z   | obrazení    | Load Tes   | t Tj        | ým       |             |            |         |        |
| 采   | A     | <b>*</b> | *       | Tileúska   |               |              | <b>Připojer</b><br>Vlastno | ní<br>sti | <b>2</b> ↓ |     | T           | Ҡ Vymaza   | it<br>znovu |          |             |            |         |        |
| Ac  | cess  | webu     | textu   | zdrojů *   | připojení     | vše 👻 🥯      | Upravit                    | odkazy    | Ã↓ Se      | rau | t Filt      | ゾ Upřesn   | it          | sloup    | ů duplicity | dat *      | Sloucit | analý: |
|     |       | Načís    | t exter | ní data    |               | Připo        | jení                       |           |            |     | Seřadit a f | iltrovat   |             |          | Da          | tové nást  | roje    |        |
|     |       | J4       |         | <b>-</b>   | $f_{x}$       |              |                            |           |            |     |             |            |             |          |             |            |         |        |
|     |       | Α        |         | В          | С             | D            | Е                          | F         | 0          | 6   |             | н          | 1           |          | J           | К          |         | L      |
| 1   | Čislo | dokladı  | u Dat   | um         | Materiál      | Místo určení | Váha                       | SPZ       | Řid        | ič  | Poznámk     | a          |             |          |             |            |         |        |
| 2   |       |          | 5 19.   | 5.2014 1:5 | 3 TEST        | TEST         | 900                        | SPZ TES   | ST TES     | т   | POZNAM      | IKA TEST   |             |          |             |            |         |        |
| 3   |       |          | 6 19.   | 5.2014 1:5 | 4 TEST1       | TEST1        | 18000                      | TEST1     | TES        | Τ1  | POZNAM      | IKA TEST1  |             |          |             |            |         |        |
| 4   |       |          |         |            |               |              |                            |           |            |     |             |            |             |          |             |            |         |        |
| 5   |       |          |         |            |               |              |                            |           |            |     |             |            |             |          |             |            |         |        |
| 6   |       |          |         |            |               |              |                            |           |            |     |             |            |             |          |             |            |         |        |
| 7   |       |          |         |            |               |              |                            |           |            |     |             |            |             |          |             |            |         |        |

Obrázek 12: Ukázka dat CSV formátu Open Office

Jak můžeme vidět na obrázcích 10, 11 a 12, po načtení jak v programu Microsoft Excel, tak i v Open Office, je zobrazení dat úplně stejné. Podmínka pro správné načtení je nutnost v importu dat nastavit kódování textu na UTF.

Aby uživatel mohl exportovat data do souboru ve formátu CSV, musí v programu spustit export dat. Zobrazí se nový formulář, který je určen pro nastavení exportu.



Obrázek 13: Ukázka formuláře pro export dat

Formulář pro export do souboru CSV formátu, umožňuje snadný výběr dat a následně export podle zadaných kritérii. Filtrujeme data podle identifikačního čísla (IČO) a názvu firmy. Stačí nastavit jenom jednu položku, buď identifikační číslo (IČO) nebo název. Tyto dva comboboxy jsou na sebe závislé. To znamená, jakmile se změní jeden, druhý na to ihned reaguje a nastaví se podle toho, který byl změněn. Další kritérium, jež je potřeba nastavit, je druh. Jestliže chceme vyexportovat specifický druh od této firmy, tak v comboboxu druh zvolíme požadovanou položku, jinak nastavíme hodnotu na položku nevybráno a tím docílíme, že se vyexportují všechny druhy pro vybranou firmu. Poslední položkou je výběr od kdy do kdy. Po kliknutí na tlačítko export se vygeneruje název souboru Druh\_NazevFirmy\_Odkdy\_Dokdy.csv a program zobrazí dialogové okno na uložení souboru tak, jak jsme zvyklí z Windows aplikací, například Microsoft Word, Excel, Poznámkový blok, Malování atd. Jestliže nenastane žádná chyba, vyskočí dialogové okno, ve kterém jsme informováni o úspěšném exportu. Tento export funguje na stejném principu i u železniční váhy.

Pro výběr dat podle zadaných kritérií používám metodu ExportDBtoCsvDoklad v třídě Databaze.

```
static public AGP Vaha.dbDoklad[] ExportDBtoCsvDoklad(intico, string
druh, DateTimedatumOd, DateTimedatumDo)
{
      DataClassesAGPDataContext db = newDataClassesAGPDataContext();
      if (druh != "NEVYBRÁNO")
       {
            vardataExport = (from d indb.dbDoklad
                              whered.druh == druh && d.idIco == ico &&
                              d.datum.Value.Date >= datumOd &&
                              d.datum.Value.Date <= datumDo</pre>
                              select d).ToArray();
            returndataExport;
      }
      else
      {
            vardataExport = (from d indb.dbDoklad
                              where d.idIco == ico &&
                              d.datum.Value.Date >= datumOd &&
                              d.datum.Value.Date <= datumDo</pre>
                              select d).ToArray();
            returndataExport;
      }
}
```

Zde je ukázka funkce pro výběr dat. Nejdříve zkontrolujeme druh, jestli je nastaven na specifický druh nebo na položku nevybráno. Jestliže je podmínka splněna zavoláme dotaz hledání v databázi, kde vybíráme podle druh, IČ, datum od a datum do. Jinak se zavolá druhý dotaz, ve kterém chybí vyhledávání podle druhu. Nalezená data, která jsou získána z funkce, odesíláme na export. Export zajišťuje třída ExportToCSV a její metoda ExportCSV, které jsou předány data z LINQ dotazu.

```
public void ExportCSV(dbDoklad[] dotaz, stringFilename)
      try
      {
            using (StreamWriter sw = newStreamWriter(Filename, false)
            {
                  var lineNadpis =
                  string.Format("Čislodokladu;Datum;Materiál;Místourčen
                  i;Váha;SPZ;Řidič;Poznámka;");
                  sw.WriteLine(lineNadpis);
                  foreach (var d in dotaz)
                        var line = string.Format(d.id.ToString() + ";"
                        + d.datum.ToString() + ";" + d.druh.ToString()
                        + ";" + d.mistoUrceni.ToString() + ";" +
                        d.netto.ToString() + ";" + d.spz.ToString() +
                        ";" + d.ridic.ToString() + ";" +
                        d.poznamka.ToString() + ";");
                        sw.WriteLine(line);
                  }
            sw.Flush();
            }
     }
     catch (Exception ex)
     {
     throw (ex);
     }
}
```

Tato funkce zapisuje data pomocí funkce třídy StreamWriter. Nejdříve zapíšeme do souboru nadpis, ve kterém je uložen popis dat, které ukládáme. Nato se spouští cyklus foreach ve kterém funkce Writeline vypíše do souboru naše nalezená data a oddělí je středníkem. Tato funkce na konci vyvolá odřádkování. Cyklus se zase od znova spouští, dokud se nezapíšou všechna vyfiltrovaná data.

### 3.5 Tisk vybraného vážního lístku

Pro tisk požadovaného vážního lístku využívám třídu PrintDocument, třídu Tisk a metodu DataNaTiskDoklad ve třídě Databaze. Po stisknutí tlačítka tisk se spustí funkce, která zobrazí dialogové okno o možnostech nastavení tiskárny, počet kopii atd. Jestliže tento formulář uživatel potvrdí tlačítkem ok, spustí se událost PrintPageEventHandler, která zavolá funkci printDoc\_PrintPage. V této metodě je zjištěn vybraný řádek v tabulce a ten je předám funkci TiskDataDoklad ve třídě Tisk. TiskDataDoklad zavolá funkci DataNaTiskDoklad z třídy Databaze. U železniční váhy je to stejné, akorát se volají jinak pojmenované funkce, ve kterých jsou lehké změny, například RZ auta je nahrazena číslem vagónu atd.

```
static public dbDokladDataNaTiskDoklad(int id)
{
    DataClassesAGPDataContextdb = newDataClassesAGPDataContext();
    var dataTisk = (from d in db.dbDoklad
        where d.id == id
        select d).SingleOrDefault();
    return dataTisk;
}
```

Vyhledávání dat pro železniční váhu je stejné jako u této funkce, jenom nevyhledáváme v db.dbDoklad ale vdb.dbZelva.

Protože kód pro přípravu dokumentu na tisk je velmi dlouhý, jsou zde uvedeny pouze nejdůležitější části funkce TiskDataDoklad

```
public void TiskDataDoklad (PrintPageEventArgs e, int i)
{
     Graphics q = e.Graphics;
     dbDoklad d = Databaze.DataNaTiskDoklad(i);
     dataTisk[1] = "VÁŽNÍ A DODACÍ LIST č." + d.id +
     Environment.NewLine;
     dataTisk[2] = "Vozidlo SPZ: " + d.spz + " Vlek SPZ: " +
     d.spzPrives +
     Environment.NewLine +"Ze dne : " + d.datum;
     Fontfont = newFont("Courier New", 12);
     FontfontNadpis = newFont("Courier New", 20);
     int x1 = 20;
     int y1 = 80;
     //Doklad 1.
     RectangleFobdelnikVaha = newRectangleF(x1 + 480, y1 + 140, 200,
     20);
     RectangleFobdelnikFirma = newRectangleF(x1, y1 + 180, 780, 65);
     g.DrawRectangle(Pens.Black, Rectangle.Round(obdelnikVaha));
     g.DrawRectangle(Pens.Black, Rectangle.Round(obdelnikFirma));
     g.DrawString(dataTisk[0], font, Brushes.Black, x1, y1); //Agro
     podnik Jihlava
     g.DrawString(dataTisk[1], fontNadpis, Brushes.Black, x1, y1 +
     50); // Nadpis
     g.DrawString(dataTisk[2], font, Brushes.Black, x1, y1 + 90); //
     SPZ a čas
```

}

Funkce TiskDataDokladvybírá text a aktuální data z databáze která se mají vypsat na stránku. Nastavuje se zde taky v jakém písmu (font), velikosti a barvy. Poslední věc, která se zde řeší, je pozice vykreslování řádku na stránku a jestli se vykreslují nějaké

obrazce. V této funkci se vykreslují obdélníky kolem váhy netto a u odběratele pro lepší čitelnost.

| Dodavatel : Agropodnik, akciová společnost Jih<br>DIC : 315-46904689 Provozo | nlava Dobronín 315 588 13 Polná<br>ovna - Dobronín |
|--|--|
| VÁŽNÍ A DODACÍ LIST č.5  |  |
| Vozidlo SPZ: TEST SPZ Vlek SPZ: TEST SPZ<br>Ze dne : 19.5.2014 4:37:13       |  |
| Tarra = 12344 Kg Brutto = 30338 Kg   | Netto = 17994 Kg                                   |
| Odběratel:<br>ICO : 2 TEST ULICE 1 MĚSTO 111 11<br>DIC : DIC 1111            |  |
| Druh : TEST DRUH<br>Množství : 17,994 t<br>Místo určení : TEST MÍSTO URČENÍ  | Řidič : TEST ŘIDIČ                                 |
| Předal :   | Převzal :  |

#### Obrázek 14: Ukázka tisku vážních lístků

### 3.6 Uživatelské rozhraní

Uživatelské rozhraní vychází z konceptu původní aplikace a řeší její nedostatky. Je navrženo s ohledem na co nejjednodušší obsluhu programu. Aplikace se spouští v okně ale umí reagovat i na maximalizaci formuláře roztáhnutím prvků.

| iční váha Adre  | esář firem Ž   | elezniční váha  |   |  |   |   |   |  |  |   |  |
|---|--|---|---|--|---|---|---|--|--|---|--|
| Nový dok  | klad   |   | Tisk Exc  | æl                                     |   |   |   |  |  | СОМ   |  |
| Vážní Záznam  |  |   |   |  | Važení  | Zadat vanu ru   | ucne  |  |  |   |  |
| Doklad  | 1  |   |   |  | Tara  | 2   |   |  | 6  |   | <b>^</b>   |
| IČO / Název   | 27082440   | - Alza.cz   | a.s.  | <b>•</b>                               | Dente   |   |   |  |  | )550  | 5  |
| SPZ   | Z PEH-0001   | SPZ Př  | ivēs PEH-1111   |  | Bruto   | 1   |   |  | _  |   |  |
| Místo určeni  | f Pelhrimov  |   |   |  | Netto   | 3   |   |  | ● Le   | vá váha Pravá   | váha 💿   |
| Druh  | Ropa   |   |   |  |   |   |   |  |  |   | -  |
| -   |  |   | _   |  |   |   |   |  |  | Zvazit  |  |
| Ridič<br>Poznámka                                       | <ul> <li>Novák</li> <li>Drahej nál</li> </ul>  | klad  |   |  | Uložit dokla  | d Smaza   | t doklad  |  |  |   |  |
| Ridič<br>Poznámka                                       | 5 Novák<br>a Drahej nál  | klad  |   |  | Uložit dokla  | ad Smaza  | t doklad<br>© Z   | (obrazit všechn  | y záznamy 💿  | Zobrazit neukor   | nčené zázna  |
| Ridič<br>Poznámka<br>Číslo<br>okladu                    | 5 Novák<br>Drahej nál  | klad<br>SPZ<br>Přívěs   | IČO   | Druh                                   | Uložit dokla  | Bruto   | e doklad<br>v z<br>Netto  | <b>čobrazit všechn</b><br>Místo<br>určení  | <b>y záznamy</b> •<br>Řidič                              | Zobrazit neukor   | <b>nčené zázna</b><br>Poznami                            |
| Ridič<br>Poznámka<br>Číslo<br>okladu P                  | <ul> <li>Novák</li> <li>Drahej nál</li> <li>SPZ</li> <li>PEH-0001</li> </ul>   | SPZ<br>Přívěs<br>PEH-1111                                     | IČO<br>27082440   | Druh                                   | Uložit dokla<br>Tara                                | Bruto   | t doklad<br>© 2<br>Netto<br>3   | Cobrazit všechn<br>Místo<br>určení<br>Pelhrimov                                  | y záznamy ●<br>Řidič<br>Novák                            | Zobrazit neukor<br>Datum<br>25.4.2014   | nčené zázna<br>Poznami<br>Drahej nákla                   |
| Ridič<br>Poznámka<br>Číslo<br>okladu P<br>Jl            | <ul> <li>Novák</li> <li>Drahej nál</li> <li>SPZ</li> <li>SPL-0001</li> <li>IH-0002</li> </ul>                                    | SPZ<br>Přívěs<br>PEH-1111<br>JIH-2222                         | IČO<br>27082440<br>60193336   | Druh<br>Ropa<br>Voda                   | Uložit dokla<br>Tara<br>2<br>5                      | Bruto   | t doklad<br>© Z<br>Netto<br>2<br>8  | Zobrazit všechn<br>Místo<br>určení<br>Pelhrimov<br>Jihlava                       | y záznamy •<br>Řidič<br>Novák<br>Hron                    | Zobrazit neukor<br>Datum<br>25.4.2014<br>27.4.2014  | nčené zázna<br>Poznami<br>Drahej nákla<br>Super          |
| Ridič<br>Poznámka<br>Číslo<br>okladu P<br>Jl<br>A       | <ul> <li>Novák</li> <li>Drahej nál</li> <li>SPZ</li> <li>SPL-0001</li> <li>IH-0002</li> <li>AA-0003</li> </ul>                   | Klad<br>SPZ<br>Přívěs<br>PEH-1111<br>JIH-2222<br>AAA-3333     | IČO<br>27082440<br>60193336<br>26168685                                     | Druh<br>Ropa<br>Voda<br>Maso           | Uložit dokla<br>Tara<br>2<br>5<br>7                 | Bruto   | • z<br>• z<br>• z<br>• z<br>• z<br>• z<br>• z<br>• z  | Cobrazit všechn<br>Místo<br>určení<br>Pelhimov<br>Jihlava<br>Praha               | y záznamy<br>Řidič<br>Novák<br>Hron<br>Vrbycka           | Zobrazit neukor<br>Datum<br>25.4.2014<br>27.4.2014<br>30.4.2014                                   | nčené zázna<br>Poznami<br>Drahej nákla<br>Super<br>nevim |
| Ridič<br>Poznámka<br>Číslo<br>okladu<br>Jl<br>A<br>P    | <ul> <li>Novák</li> <li>Drahej nál</li> <li>SPZ</li> <li>EH-0001</li> <li>IH-0002</li> <li>AA-0003</li> <li>EH-3802</li> </ul>   | SPZ<br>Přívěs<br>PEH-1111<br>JIH-2222<br>AAA-3333<br>PEH-3729 | IČO<br>27082440<br>60193336<br>26168685<br>26168685                         | Druh<br>Ropa<br>Voda<br>Maso<br>Cement | Uložit dokla<br>Tara<br>2<br>5<br>7<br>100          | Smazal<br>Bruto<br>1<br>13<br>300   | t doklad<br>© z<br>Netto<br>8<br>20<br>200  | Zobrazit všechn<br>Místo<br>určení<br>Pehrimov<br>Jihlava<br>Praha<br>Bratislava | y záznamy<br>Řidič<br>Novák<br>Hron<br>Vrbycka<br>Gott   | Zobrazit neukov<br>Datum<br>25.4.2014<br>27.4.2014<br>30.4.2014<br>30.4.2014                      | nčené zázna<br>Poznami<br>Drahej nákla<br>Super<br>nevim |
| Ridič<br>Poznámka<br>Číslo<br>okladu P<br>J<br>A<br>P   | <ul> <li>Novák</li> <li>Drahej nál</li> <li>SPZ</li> <li>PEH-0001</li> <li>IH-0002</li> <li>AA-0003</li> <li>PEH-3802</li> </ul> | SPZ<br>Přívěs<br>PEH-1111<br>JIH-2222<br>AAA-3333<br>PEH-3729 | IČO<br>27082440<br>60193336<br>26168685<br>26168685<br>26168685<br>27082440 | Druh<br>Ropa<br>Voda<br>Maso<br>Cement | Uložit dokla<br>Tara<br>2<br>5<br>7<br>100<br>0     | Smazal<br>Bruto<br>1<br>13<br>300<br>0  | t doklad<br>Netto<br>8<br>20<br>200<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0<br>0 | Cobrazit všechn<br>Místo<br>určení<br>Pelhimov<br>Jihlava<br>Praha<br>Bratislava | y záznamy<br>Řidič<br>Novák<br>Hron<br>Vrbycka<br>Gott   | Zobrazit neuko<br>Datum<br>25.4.2014<br>27.4.2014<br>30.4.2014<br>30.4.2014<br>19.5.201422:       | nčené zázna<br>Poznami<br>Drahej nákla<br>Super<br>nevim |
| Ridič<br>Poznámka<br>Číslo<br>lokladu P<br>JI<br>A<br>P | 5 Novák<br>a Drahej nář<br>SPZ<br>2EH-0001<br>IH-0002<br>VAA-0003<br>2EH-3802  | SPZ<br>Přívěs<br>PEH-1111<br>JIH-2222<br>AAA-3333<br>PEH-3729 | IČO<br>27082440<br>60193336<br>26168685<br>27082440<br>27082440<br>27082440 | Druh<br>Ropa<br>Voda<br>Maso<br>Cement | Uložit dokla<br>Tara<br>5<br>7<br>100<br>0<br>12392 | Smazal     Smazal | t doklad  | Cobrazit všechn<br>Místo<br>určení<br>Pelhimov<br>Jihlava<br>Praha<br>Bratislava | y záznamy •<br>Řidič<br>Novák<br>Hron<br>Vrbycka<br>Gott | Zobrazit neukov<br>Datum<br>27.4.2014<br>30.4.2014<br>30.4.2014<br>19.5.2014 22:<br>19.5.2014 22: | nčené zázna<br>Poznamk<br>Drahej nákla<br>Super<br>nevim |

Obrázek 15: Ukázka hlavního okna programu

Obrázek 15 zobrazuje aplikaci po spuštění. Výchozí okno aplikace se dělí na tři části.

### 3.6.1 Popis silniční váhy

Záložka v aplikaci se jménem silniční váha, umožnuje obsluze vyplnit a následně uložit vážní lístek do databáze. Po příjezdu váženého vozidla, obsluha musí vyplnit údaje obsažené v komponentě groupBox pojmenované vážní záznam.

Kliknutím na tlačítko Nový doklad se zvýší ID o jedničku a přiřadí se do kolonky číslo dokladu, vymažeme se dosavadní text komponentách TextBox. Vyplněním potřebných údajů může obsluha uložit záznam pomocí tlačítka uložit doklad, kde se zavolá funkce, která záznam zapíše do databáze, kde s ním můžeme dále pracovat.

Funkce na ukládání dat do tabulky dbDoklad

```
public static void InsertEditDoklad(int id, string spz, string
spzPrives, int ico, string druh, int tara, int netto, int brutto,
string mistoUrceni, stringridic, DateTime datum, string poznamka)
{
    DataClassesAGPDataContext db = newDataClassesAGPDataContext();
    var nalezeno = (from d in db.dbDoklad
        where d.id == id
            select d).SingleOrDefault();
```

```
if (nalezeno == null)
{
      Table<dbDoklad> dbDoklady = Databaze.GetDokladDB();
      dbDoklad data = newdbDoklad();
      try
      {
            data.id = id;
              data.spz = spz;
              data.spzPrives = spzPrives;
              data.idIco = ico;
              data.druh = druh;
              data.tara = tara;
              data.netto = netto;
              data.bruto = brutto;
              data.mistoUrceni = mistoUrceni;
              data.ridic = ridic;
              data.datum = datum;
              data.poznamka = poznamka;
              dbDoklady.InsertOnSubmit(data);
              dbDoklady.Context.SubmitChanges();
              db.SubmitChanges();
      }
      catch (Exception ex)
      {
            throw ex;
      }
}
else
{
      try
      {
              nalezeno.spz = spz;
              nalezeno.spzPrives = spz;
              nalezeno.idIco = ico;
              nalezeno.druh = druh;
              nalezeno.tara = tara;
              nalezeno.bruto = brutto;
              nalezeno.netto = netto;
              nalezeno.mistoUrceni = mistoUrceni;
              nalezeno.ridic = ridic;
              nalezeno.datum = datum;
              nalezeno.poznamka = poznamka;
              db.SubmitChanges();
      }
      catch (Exception ex)
      {
            throw ex;
      }
}
```

}

Tato funkce umí jak ukládat, tak i editovat data v tabulce dbDoklad. Nejdříve proběhne ve funkci dotaz, který zjistí, zdali se v tabulce dbDodklad nachází zadané id. Pokud ne

vytvoří se nový záznam. Jestli je id nalezeno, tak se edituje už uložený záznam pod id, které jsme této funkci přiřadili.

Některé komponenty umístěné ve vážním záznamu obsahují funkci našeptávání, kterou má například i vyhledávač od firmy Google. Tato vlastnost se hodí pro usnadnění a zrychlení při zadávaní údajů. Našeptávač funguje při vyplňování registrační značky, registrační značky přívěsu, místo určení a druh. V případě zadaní RZ obsažené v databázi jsou automaticky vyplněné související údaje (RZ přívěs, jméno řidiče a poznámka). Jestliže chceme používat vlastnost našeptávání u komponenty TextBox, musíme zavolat funkciAutoCompletedTxtBoxSpzDoklad.

```
private void AutoCompletedTxtBoxSpzDoklad()
{
    AutoCompleteStringCollection kSPZ =
    newAutoCompleteStringCollection();
    kSPZ.AddRange(Databaze.GetVsechnaSpzDoklad());
    txtSpzDoklad.AutoCompleteCustomSource = kSPZ;
}
```

Funkce AutoComplettedTxtBoxSpzDoklad potřebuje získat data, která má zobrazit. Proto se zavolá další pomocná metoda GetVsechnaSpzDoklad z třídy Databaze.

Tato metoda vrací pole string řetězců. Pomocí metody Distinct docílíme, že opakující se položka v databázi se nám nevybere dvakrát. Metoda ToArray uloží vyfiltrované jedinečné položky do pole a vrátí metodě AutoCompletedTxtBoxSpzDoklad.

Jestliže chceme našeptávání i pro ostatní komponenty TextBox, musíme zavolat podobné dotazy, které se liší pouze v příkazu select. Například pro našeptávání u kolonky, kde zadáváme jméno řidiče, nebude select d.spz ale d.ridic. Tento vybraný obsah potom přepošleme příslušnému TextBoxu.



Obrázek 16: Ukázka našeptávání

Po vyplnění povinných údajů, je možno zahájit proces vážení. Načítání údajů o váze je možno dvěma způsoby. První způsob probíhá načítáním dat ze sériového portu váhy pomocí tlačítka Zvážit. Výsledek se zobrazí v kolonce tara (hmotnost prázdného vozidla) nebo bruto (hmotnost vozidla plus náklad) podle volby obsluhy.

```
Ukázka tlačítka Zvážit
```

```
private void btnZvazDoklad Click(object sender, EventArgs e)
{
      if (addDoklad)
      {
            if (Int32.Parse(txtTaraDoklad.Text) != 0)
            {
                  txtBrutoDoklad.Text = txtUkazatelVahyDoklad.Text;
            }
            else
            {
                  txtTaraDoklad.Text = txtUkazatelVahyDoklad.Text;
            }
      }
      else
      {
      DialogResult rslt = MessageBox.Show("Je to tato váha Tara?",
      "Tara?",
                                       MessageBoxButtons.YesNo,
                                       MessageBoxIcon.Question);
      if (rslt == DialogResult.Yes)
      {
            txtTaraDoklad.Text = txtUkazatelVahyDoklad.Text;
      }
      else
      {
```

```
txtBrutoDoklad.Text = txtUkazatelVahyDoklad.Text;
}
}
```

Pokud je to první vážení vozidla tak funkce vypíše dialogové okno, ve kterém se obsluhy zeptá, zda-li vozidlo přijelo prázdné nebo plné. Podle toho se uloží váha do příslušného textBoxu. Jestli se vozidlo váží po druhé, už se program obsluhy neptá, jestli se bude vážit tara nebo bruto, ale automaticky zjistí pomocí podmínky if, která váha nebyla ještě zvážena. V podmínce if se ptáme, jestli se hodnota v textBoxu tara je rovná nule nebo ne. Jestli tara je rovno nule, uložíme váhu do tara. Jinak uložíme váhu do kolonky bruto.

Druhý způsob je zadávat váhu ručně, tato funkce byla vyžadována zadavatelem a je velmi užitečná při vážení. Pokud známe tara váhu vozidla, můžeme vynechat jedno vážení na mostové váze a obsluha zadá tara váhu ručně. Komponenty TextBox, do kterých zaznamenáváme váhu, jsou omezeny pouze na pěti-místné číslo a ošetřeny tak, aby šlo zadávat pouze číslice.

Ukázka ošetření proti chybnému zadaní váhy.

```
private void txtBoxTara_KeyPress(object sender, KeyPressEventArgs e)
{
    char znak = e.KeyChar;
    if (!char.IsDigit(znak) && znak != 8 && znak != 46)
    {
        e.Handled = true;
    }
}
```

Podmínka if by se dala přeložit jako: jestliže se příchozí znak nerovná číslu od 0 do 9 a zároveň to není znak s hodnotou 8 (Backspace) a s hodnotu 46 (Delete), nic nedělej, jinak povol zapsání znaku. V podmínce musíme kontrolovat i právě hodnotu backspace nebo delete jinak by nám tyto klávesy přestaly fungovat.

Počítání váhy zajišťuje metoda SpocitatVahu, která se volá při změně textu v TextBox tara nebo bruto. Tím docílíme automatické přepočítávání netto při zápisu.

```
private void SpocitatVahu()
      if (txtTaraDoklad.Text != "0"&& txtTaraDoklad.Text != ""&&
          txtBrutoDoklad.Text != "0"&& txtBrutoDoklad.Text != "")
      {
            try
            {
                  txtNettoDoklad.Text =
                  Convert.ToString(Int32.Parse(txtBrutoDoklad.Text) -
                  Int32.Parse(txtTaraDoklad.Text));
            }
            catch
            {
            }
      }
      else
      {
            txtNettoDoklad.Text = "0";
      }
}
```

V metodě SpocitatVahu kontroluji, zda je zadaná hodnota jiná než 0 nebo prázdný řetězec. Pokud ano, spočítá se váha rozdílem bruto od tary a uloží do netto. V případě, že není podmínka splněna, nastaví se netto na 0. Tato podmínka je velice důležitá, kdyby se provedla funkce Parse s hodnotou prázdného řetězce a nebyla ošetřena blokem try a catch, mohl by celý program "zamrznout", vyhodit chybu a ukončit se.

Ve spodní části aplikace se zobrazuje náhled do databáze vážených vozidel a upozorňuje barevným odlišením na rozvážené (nedokončené) doklady. To je velmi

důležité pro obsluhu, aby věděla, které vážní lístky jsou už dokončeny (mají vypočítanou váhu netto) a naopak které ještě potřebují dokončit vážení. Kdyby tato aplikace neuměla rozlišit mezi rozdělanými a hotovými záznamy, mohla by se používat pouze pro vážení vždy jednoho vozidla. To znamená, že bychom vozidlo zvážili poprvé, poté by jelo vyložit nebo naložit náklad a zvážilo by se po druhé. Teprve potom by obsluha mohla uložit vážní lístek. Avšak toto řešení není realizovatelné v běžném provozu, protože by se nemohlo vážit vícero vozidel za sebou. Vždy by se muselo počkat na aktuálně vážené vozidlo, dokud nedokončí oba dva procesy vážení. Proto byla přidaná podmínka, která rozlišuje, zdali je vážní lístek dokončen, nebo je ještě potřeba vozidlo nechat zvážit. K tomuto ošetření slouží podmínka KontrolaNetta.

```
private void KontrolaNetta()
      if (dataGViewDoklad.SelectedRows.Count > 0)
      {
            DataGridViewRow SelectedRow =
      dataGViewDoklad.SelectedRows[0];
            string RowValueCell7 =
      SelectedRow.Cells[7].Value.ToString();
            if (RowValueCell7 == "0")
            {
                  dokoncitDoklad = true;
                  ZmenaNaDokoncitDoklad(true);
                  btnUlozDokoncitDoklad.Enabled = true;
            }
            else
            {
                  dokoncitDoklad = false;
                  btnUlozDokoncitDoklad.Enabled = false;
                  ZmenaNaDokoncitDoklad(false);
            }
            txtIdDoklad.Text =
      Convert.ToString(SelectedRow.Cells[0].Value);
            txtSpzDoklad.Text =
      Convert.ToString(SelectedRow.Cells[1].Value);
      }
```

Tato funkce obstarává změnu ovládacích prvků a umožnuje zapnutí tlačítka dokončit. Jestliže se netto rovná nule, funkce zjistí, že je vybrán otevřený doklad a zapne tlačítko Dokončit. Tímto tlačítkem také můžeme znovu otevřít rozdělaný doklad a zaznamenat druhé vážení. Tuto funkci volám v události, kdy se změní vybraná položka v tabulce, nebo kdy je zapnutá interakce s tabulkou.

Pro změnu barvy u rozváženého vážního lístku je využitá funkce ZmenBarvyDataGridViewDoklad()

```
privatevoid ZmenaBarvyDataGridViewDoklad()
{
    foreach (DataGridViewRow row in dataGViewDoklad.Rows)
    {
        string RowType = row.Cells[7].Value.ToString();
        if (RowType == "0")
        {
            row.DefaultCellStyle.BackColor = Color.Red;
            row.DefaultCellStyle.ForeColor = Color.White;
            row.DefaultCellStyle.SelectionForeColor =
            Color.White;
            row.DefaultCellStyle.SelectionBackColor =
            Color.MediumVioletRed;
        }
    }
}
```

Pomocí cyklu foreach procházím každý řádek v tabulce dataGViewDoklad a porovnávám hodnotu v buňce netto s nulou. Jestli porovnávaný řádek má nulu v buňce položky netta, je obarven na červeno.

Mazání je zajištěno pomocí tlačítka Smazat. Toto tlačítko zobrazí dialog, že se obsluha chystá smazat vážní lístek a jestli jej opravdu chce smazat. Podle požadavků zadavatele se smaže jenom poslední vytvořený vážní lístek. Jinak řečeno ten co má nejvyšší id (identifikační číslo).

Poslední důležitý požadavek zadavatele, je možnost editace uloženého záznamu. To jsem vyřešil pomocí zobrazení nového formuláře. Ten se vyvolá pomocí kliknutí pravého tlačítka myši na požadovaný řádek v tabulce, který je potřeba editovat.

| ,                | doklad                    |               | Tisk         | Excel  |             | / Zadat vábu u | aučně      |                   |            | СОМ             |             |
|------------------|---------------------------|---------------|--------------|--------|-------------|----------------|------------|-------------------|------------|-----------------|-------------|
| Vážní Zázna      | am                        | _             |              |        | Važení      |                |            |                   | _          |                 | _           |
| Dok              | dad 1                     |               |              |        | Tara        | 1111           |            |                   | 6          | 556             | <b>२</b> -  |
| ICO / Ná         | CO / Název 11111 • TEST • |               |              | Bruto  | 22222       |                |            |                   | 000        | J               |             |
| 5                | SPZ TEST                  | SP            | Z Přívěs TES | ST     | Netto       | 21111          |            |                   | Levá       | váha Pravá va   | iha 💿       |
| Misto uro        | TEST                      |               |              |        | Notio       | 2              |            |                   |            |                 |             |
| Ď                | idia TEST                 |               | _            |        | Liložit dok | ad Smar        | at doklad  |                   |            | Zvážit          |             |
| Poznán           | nka TEST                  |               |              |        |             |                | ● Zot      | prazit všechny zá | znamy • Zo | obrazit neukono | čené záznan |
| Číslo<br>lokladu | SPZ                       | SPZ<br>Přívěs | IČO          | Druh   | Tara        | Bruto          | Netto      | Místo<br>určení   | Řidič      | Datum           | Poznamk     |
|                  | TEST                      | TEST          | 11111        | TEST   | 1111        | 22222          | Editace zá | znamu             | TEST       | 19.5.2014 2     | TEST        |
|                  |                           | LIFOT1        | 111111       | ITEST1 | 115151      | 121544         |            | T1                | LIEST1     | 119 5 2014 2    | LIEST       |

Obrázek 17: Kontextová nabídka pro editaci

| Form_Edit       | - Secold      | and the second se |   |        |      |        | x       |
|-----------------|---------------|---|---|--------|------|--------|---------|
| -Vážní Záznam — |               |   |   | Važení |      |        | 1       |
| Doklad          | 1             |   |   | Tara   | 121  | Kg     |         |
| IČO / Název     | TEST 🗸        | 2   | • | Bruto  | 1212 | Ka     |         |
| SPZ             | TEST          | SPZ Přívěs TEST   |   | Didto  | 1212 | ity    |         |
| Místo určení    | TEST          |   |   | Netto  | 1091 | Kg     |         |
| Druh            | TEST          |   |   |        |      |        | li<br>I |
| Řidič           | TEST          |   |   | ок     |      | Storno |         |
| Poznámka        | TEST          |   |   |        |      |        |         |
| Datum           | 19. května 20 | 114 🔲 🔻   |   |        |      |        |         |

Při výběru kontextové nabídky se vyvolá nový formulář Form\_EditDoklad.

Obrázek 18: Formulář editace vážního lístku

Tento formulář umožnuje editaci vážního lístků a zpětné uložení. GroupBox vážení funguje stejně jako na hlavním formuláři.

### 3.6.2 Adresář firem - popis

Pod záložkou adresář firem se nachází formulář pro přidávání firem do databáze.

| oor Nastave | ení           |            |                          |                         |                |               |                |                                |            |
|-------------|---------------|------------|--------------------------|-------------------------|----------------|---------------|----------------|--------------------------------|------------|
| niční váha  | Adresář firem | Železniční | váha                     |                         |                |               |                |                                |            |
| Přid        | lat firmu     |            |                          |                         |                |               |                |                                |            |
| Ú4-2 6-     | X             |            |                          |                         |                |               |                |                                |            |
| Odaje o III | me            | IČO        | 2222222                  | Načíst z                | Ares           |               |                |                                |            |
| 1           | Název firmy   | TEST2      |                          | Ulice                   | TEST2          |               |                |                                |            |
| Do          | movní Číslo   | 22222      |                          | Mēsto                   | TEST2          |               |                |                                |            |
|             | DIČ           | TEST22     | 2                        | PSČ                     | 222 22         |               |                |                                |            |
| Firma       | v insolvenci  |            |                          | Platce DPH              |                |               |                |                                |            |
|             |               |            |                          |                         |                |               |                |                                |            |
|             |               |            | Llložit                  | Smazat                  |                |               |                |                                |            |
|             |               |            | Uložit                   | Smazat                  |                |               |                |                                |            |
|             |               |            | Uložit                   | Smazat                  |                |               |                |                                |            |
| IČO         | N             | lázev      | Uložit<br>Ulice          | Smazat<br>Číslo popisné | Město          | PSČ           | DIČ            | Insolvenční<br>řízení          | Platce DPH |
| IČO         | N<br>TEST1    | lázev      | Uložit<br>Ulice<br>TEST1 | Číslo popisné           | Město<br>TEST1 | PSČ<br>111 11 | DIČ<br>TEST111 | Insolvenční<br>řízení<br>False | Platce DPH |

#### Obrázek 19: Formulář adresáře firem

V tomto formuláři můžeme přidávat, editovat a mazat údaje v databázi. Chybí zde našeptávání, ale přišlo mi zbytečné ho zde použít, když každá firma, má jiný název, sídlo, DIČ, identifikační číslo (IČO). Místo toho je zde použit ARES [27], který byl již popsán v kapitole 3.3. Tato tabulka má stejnou kontextovou nabídku jako tabulka u silniční váhy, opět je intuitivně vyvolána druhým tlačítkem na myši nebo touchpadu. Poté se opět otevře editační formulář. Tentokrát ale se otevře Form\_EditFirma.

| Form_EditFirma     | and the second s |               | X      |
|--------------------|--|---------------|--------|
| Údaje o firmě      |  |               |        |
| IČO                | 111111   |               |        |
| Název firmy        | TEST   | Ulice         | TEST   |
| Město              | TEST   | Domovní Číslo | 1111   |
| DIČ                | TEST   | PSČ           | 111 11 |
| Firma v insolvenci |  | Platce DPH    |        |
|                    | ОК   | Storno        | łł.    |

Obrázek 20: Formulář editace adresář firma

# 3.6.3 Železniční váha

Poslední záložka zobrazuje formulář Železniční váha. Tento formulář je velmi podobný silniční váze, jenom zde schází ukazatel váhy ze sériové linky. Po konzultaci se zaměstnanci Agropodniku jsem se dozvěděl, že komunikace mezi železniční váhou a stávajícím PC nefunguje. A že záznamy o váze z železniční váhy zadávají stejně do stávající aplikace ručně, proto jsem tento systém zachoval. GroupBox vážení funguje úplně stejně jako v silniční váze. Pro toho už zde nebudu popisovat.

| Form1   |                        |           | W. states had   | trouble thread | and the second sec |              |                 |          |
|---|------------------------|-----------|---|----------------|--|--------------|-----------------|----------|
| Silniční váha Adre  | sář firem Železniční v | áha       |   |                |  |              |                 |          |
| Vážní Záznam<br>Doklad<br>IČO / Název<br>Číslo vágonu<br>Místo určení<br>Druh<br>Poznámka | Iad                    | ST •      | Važení<br>Tara 0<br>Bruto 0<br>Netto 0<br>Ulož doklad | Smazat doklad  |  |              |                 |          |
| Číslo dokladu   | Číslo vagónu           | IČO Drul  | n Tara  | Bruto          | Netto  | Místo určení | Datum           | Poznámka |
| 1   | 111111111 111          | IIII TEST | 1332  | 28292          | 26960  | TEST         | 19.5.2014 23:20 | TEST     |
|   |                        |           |   |                |  |              |                 |          |

Obrázek 21: Formulář železniční váha

# 4 Závěr

Hlavním cílem této práce bylo vytvořit desktopovou aplikaci pro silniční a drážní váhu s ukládáním dat do databáze. Tato aplikace bude využívána firmou Agropodnik Jihlava a.s. Hlavními požadovanými funkcemi programu byla evidence hmotnosti projíždějících vozidel a vlaků, manuální vkládání dat pro práci v offline režimu, export dat a návrh tiskových sestav dle uživatelových požadavků. Aplikace byla vytvářena na základě stávající, již nevyhovující.

Nejprve bylo nutné provést analýzu neveřejného komunikačního protokolu na sběrnici RS-232, po níž jsou data předávána do počítače a následně se sejít se zadavatelem a prokonzultovat jeho požadavky. Na základě těchto požadavků a současného nevyhovujícího stavu programu jsem vytvořil aplikaci, která je funkční na moderních operačních systémech Windows. Snažil jsem se o vytvoření programu, jehož ovládání je jednoduché a intuitivní.

Díky této práci došlo ke zlepšení mých znalostí jazyka C# v souvislosti s prací s formuláři, třídou Print, nebo s využitím databází. Bylo vhodné naučit se používat i třídu LINQ, hlavně tedy LINQ to SQL, pomocí které jsem přistupoval do databáze a psal dotazy. V programu jsem využil i třídu SerialPort, díky které byla zachytávána data ze silniční váhy.

Největší zkušeností pro mě byla přímá komunikace se zadavatelem, kdy bylo nutné správně pochopit jeho požadavky. Každé neporozumění totiž vede k nadbytečným úpravám a zbytečným komplikacím. Setkal jsem se systémem ARES, který je využíván ve všech pokročilejších informačních systémech.

I přes veškeré obtíže týkající se dekódování váhy, byly nakonec všechny zadané cíle splněny. Vznikla aplikace, která je jednodušší, lépe vypadající, lépe ovladatelná a zahrnuje požadovanou funkcionalitu, které zjednoduší práci uživatele (např. našeptávání a automatické vyplňování podle RZ).

Aplikace bude zatím v beta testování, její ostré nasazení do provozu se plánuje na polovinu června 2014, kdy dojde k odladění vzniklých chyb. Před nasazením bude vytvořen přihlašovací formulář, díky němuž bude možná existence dvou základních účtů – administrátor a uživatel.

# Seznam použité literatury

[1] Internetové stránky Váhy Švec. cz [online]. [cit. 2014-05-19]. Dostupné z: http://www.vahy-svec.cz/silnicni-mostove-mostni-vahy

[2] Internetové stránky Tenzováhy – automatické vážení za jízdy [online]. [cit. 2014-05-19]. Dostupné z: http://www.tenzovahy.cz/silnicni-mostova-vaha

[3] Internetové stránky Tenzováhy – automatické vážení za jízdy [online]. [cit. 2014-05-19]. Dostupné z: http://www.tenzovahy.cz/mobilni-silnicni-vahy-pw10

[4] Internetové stránky Omega engineering – technické reference [online]. [cit. 2014-05-19]. Dostupné z: http://www.omegaeng.cz/prodinfo/loadcells.html

[5] Weigh Indicator 1612/02 – Operating manual [online]. [cit. 2014-05-19]. Dostupné z:

[6] Komunikace po RS 232 (seriovém portu) v C# [online]. [cit. 2014-05-19]. Dostupné z: https://archive.is/sZLSp

[7] Internetové stránky HW.cz – Seriová linka RS232 [online]. [cit. 2014-05-19]. Dostupné z: http://www.hw.cz/rozhrani/hw-server-predstavuje-seriova-linka-rs-232.html#parametry

[8] [online]. [cit. 20.5.2014]. Dostupný na WWW: http://1.bp.blogspot.com/\_2BqWq9oqYaw/TRYoSPvs1gI/AAAAAAAAAWw/rcd5QpK Ah5g/s320/RS232pinout.gif

[9] Internetové stránky Papouch – Sériový port RS232[online]. [cit. 2014-05-19]. Dostupné z: http://www.papouch.com/cz/website/mainmenu/clanky/jak-na-to/rs232/

[10] Wikipedia; heslo Serial port [online]. [cit. 2014-05-19]. Dostupné z:http://en.wikipedia.org/wiki/Serial\_port#Data\_bits

[11] Internetové stránky HW.cz - Detaily o RS 232 a přenosu informací vůbec [online]. [cit. 2014-05-19]. Dostupné z: http://www.hw.cz/teorie-apraxe/dokumentace/detaily-o-rs-232-a-prenosu-informaci-vubec.html

[12] Internetové stránky Beyond Logic – Interfacing the serial / RS232 port [online]. [cit. 2014-05-19]. Dostupné z: http://retired.beyondlogic.org/serial/serial.htm#3

[13] Internetové stránky Virtual Serial Ports [online]. [cit. 2014-05-19]. Dostupné z: http://www.virtual-serial-port.com/

[14] Internetové stránky [online]. [cit. 2014-05-20]. Dostupné z: http://www.virtual-serial-port.com/img/products/vspk/case-study-garage-2.gif

[15] Internetové stránky Microsoft Developer Network – Visual C# [online]. [cit. 2014-05-19]. Dostupné z: http://msdn.microsoft.com/cs-cz/library/kx37x362.aspx

[16] Wikipedia, heslo C Sharp [online]. [cit. 2014-05-19]. Dostupné z: http://cs.wikipedia.org/wiki/C\_Sharp

[17] Běhálek, M.; Základní charakteristika jazyka C#(2007) [online]. [cit. 2014-05-19]. Dostupné z: http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text/ch02.html

[18] Internetové stránky Microsoft - Overview of the .NET Framework [online]. [cit. 2014-05-20] Dostupné z http://msdn.microsoft.com/en-us/library/zw4w595w.aspx

[19] Internetové stránky Microsoft - .NET [online]. [cit. 2014-05-19]. Dostupné z: http://www.microsoft.com/net/

[20] Internetové stránkyAdaptic [online]. [cit. 2014-05-19]. Dostupné z: http://www.adaptic.cz/znalosti/slovnicek/databaze

[21] Internetové databáze chytrak.cz [online]. [cit. 2014-05-19]. Dostupné z: http://www.databaze.chytrak.cz/modely.htm

[22] Internetová databáze chytrak.cz [online]. [cit. 2014-05-19]. Dostupné z: http://www.databaze.chytrak.cz/images/obr3.gif

[23] Wikipedia, heslo LINQ [online]. [cit. 2014-05-19]. Dostupné z: http://cs.wikipedia.org/wiki/LINQ

[24] Internetové stránky devbook.cz [online]. [cit. 2014-05-19]. Dostupné z: http://www.devbook.cz/c-sharp-tutorial-linq-dotazy

[25] SerialPort (RS-232 Serial COM Port) in C# .NET [online]. [cit. 2014-05-19]. Dostupné z: http://msmvps.com/blogs/coad/archive/2005/03/23/39466.aspx

[26] Ministerstvo financí [online]. [cit. 2014-05-19]. Dostupné z: ČR http://wwwinfo.mfcr.cz/ares/

[27] Internetové stránky nuget.org [online]. [cit. 2014-05-19]. Dostupné z: https://www.nuget.org/packages/AresWebService/

[28] Ministerstvo financí ČR [online]. [cit. 2014-05-19]. Dostupné z: http://wwwinfo.mfcr.cz/ares/xml\_doc/wsdl/basic\_1.0.2.wsdl

[29] Internetové stránky kadlecek.eu [online]. [cit. 2014-05-19]. Dostupné z: http://www.kadlecek.eu/ares/ares\_testBasic.zip

# Seznam obrázků

| Obrázek 1: Zapojeni RS-232[8]                        | 12 |
|--|----|
| Obrázek 3: Virtualní sériový port [14]               | 14 |
| Obrázek 4: Relační model databáze[22]                | 16 |
| Obrázek 5: SerialPort Terminal prostředí aplikace    | 18 |
| Obrázek 6: Komunikace v původní aplikaci             | 20 |
| Obrázek 7: Komunikace v SerialPort Terminal          | 21 |
| Obrázek 8: Stávající program s maximalizovaným oknem | 23 |
| Obrázek 9: Databázový model                          | 24 |
| Obrázek 10: Formulář nastavení                       | 27 |
| Obrázek 11: Ukázka zápisu CSV v poznámkovém bloku    | 31 |
| Obrázek 12: Ukázka zápisu csv do Microsoft Excel     | 32 |
| Obrázek 13: Ukázka zápisu csv do Open Office         | 32 |
| Obrázek 14: Ukázka formuláře pro export dat          | 33 |
| Obrázek 15: Ukázka tisku vážních lístků              | 37 |
| Obrázek 16: Ukázka hlavního okna programu            | 38 |
| Obrázek 17: Ukázka našeptávání                       | 41 |
| Obrázek 18: Kontextová nabídka pro editaci           | 46 |
| Obrázek 19: Formulář editace vážního lístku          | 46 |
| Obrázek 20: Formulář adresáře firem                  | 47 |
| Obrázek 21: Formulář editace adresář firma           | 47 |
| Obrázek 22: Formulář železniční váha                 | 48 |

# Seznam tabulek

| Tabulka 1:BeyondLogic, zkraceno autorem.[12] | 13 |
|--|----|
| Tabulka 2: dbDoklad a dbZelva                |    |
| Tabulka 3: dbFirma                           |    |
| Tabulka 4: Tabulka nastavení komunikace      |    |

# Seznam použitých zkratek

- C# Moderní programovací jazyk.
- RS-232 Sériová linka.
- LINQ to SQL Language Integrated Query to Structure Query Language.
- CSV Skládá se z dat, které jsou odděleny středníkem.
- ARES Administrativní registr ekonomických subjektů.

# Přílohy

# 1 Obsah přiloženého CD

Na přiloženém CD se v kořenovém adresáři nachází tato bakalářská práce ve formátu *bakalarska\_prace.pdf* s jednoduchým návodem *navod.txt* pro obsluhu programu.

## 2 Uživatelský manuál se nachází na přiloženém CD

# 3 Ukázka základního kódu v C# pro vytvoření Sériového portu

```
#region Namespace Inclusions
using System;
using System.IO.Ports;
using System.Windows.Forms;
#endregion
namespace SerialPortExample
{
 class SerialPortProgram
 {
   // Vytvoří serial port se základním nastavením
   private SerialPort port = new SerialPort("COM1",
     9600, Parity.None, 8, StopBits.One);
   [STAThread]
   static void Main(string[] args)
   {
     // Vytvoří instanci této třídy
     new SerialPortProgram();
   }
   private SerialPortProgram()
   {
     Console.WriteLine("Incoming Data:");
```

```
/* Tato metoda je zavolána pokud
   nějaká data čekají v port bufferu
   */
   port.DataReceived += new
     SerialDataReceivedEventHandler(port_DataReceived);
   // Otevře se port a je zahájená komunikace
   port.Open();
   // Vložení smyčky pro opakovaní aplikace
   Application.Run();
 }
 private void port DataReceived(object sender,
   SerialDataReceivedEventArgs e)
 {
   // Výpis do konsole všech příchozích dat z port bufferu
   Console.WriteLine(port.ReadExisting());
 }
}
```

}