

# DOKUMENTACE K PROGRAMU BUSGRAFIKON

## 1 Uživatelská dokumentace

### 1.1 Nutné soubory

Jízdní řády, které tato aplikace zpracovává, je nutno předložit jako soubory v poměrně specifickém formátu. Aktuální jízdní řády v tomto formátu můžete sehnat následujícím způsobem:

1. Na stránce [portal.radekpapez.cz](http://portal.radekpapez.cz) najdete vybranou (autobusovou) linku.
2. Vyberte možnost 'Stáhnout jako xls' a daný soubor otevřete v programu jako např. Microsoft Excel.
3. Vyberte buňky tak, aby byl vybrán alespoň sloupeček s tarifními čísly (vč. označení 'Tč'), názvy zastávek, kilometrickými údaji a jednotlivými spoji včetně záhlaví.
4. Je možno bez problémů vybrat celou tabulku - nadbytečné údaje jsou ignorovány. Nicméně tabulka by měla mít obdélníkový tvar.

A	B	C	D	E	F	G	H	I	J	K	L
Linka číslo 100100 Zličín - Letšňe / Terminál 1											
Plati od 31. 8. 2020 do 31. 8. 2021											
Tč											
1	Praha, Zličín	odjezd	5:20	5:20	5:30	5:44	5:50	6:08	6:10	6:25	0
2	Praha, K Letšňe		5:28	5:28	5:38	5:52	5:58	6:16	6:18	6:33	0
3	Praha, Terminál 3		5:29	5:29	5:39	5:53	5:59	6:17	6:19	6:34	0
4	Praha, U Hangleřu		5:32	5:32	5:42	5:56	6:02	6:20	6:22	6:37	0
5	Praha, Schengenská		<	<	<	<	<	<	<	<	0
6	Praha, Terminál 2		<	<	<	<	<	<	<	<	0
7	Praha, Terminál 1		5:34	5:34	5:44	5:58	6:04	6:22	6:24	6:39	10
8	Praha, Terminál 2		5:35	5:35	5:45	5:59	6:05	6:23	6:25	6:40	10
9	Praha, Letšňe	příjezd	5:36	5:36	5:46	6:00	6:06	6:24	6:26	6:41	10

5. Přípravu zajišťuje: Dopravní podnik hl. m. Prahy, a.s., Sokolovská 217/42, 190 22 Praha 9, [www.dpp.cz](http://www.dpp.cz), [dispecerik@dpp.cz](mailto:dispecerik@dpp.cz)

**do textového souboru stačí vložit  
pouze šedě označenou část**

5. Tabulku zkopírujte do textového souboru - bude tedy vypadat jako CSV s tabulátory místo čárek.
6. Stejným způsobem si můžete upravit nebo vyrobit svůj vlastní jízdní řád.
7. Vybrané textové soubory pak zkopírujte do místa, kde leží vaše aplikace - budete se na ně odkazovat pomocí relativních cest.

### 1.2 Používání aplikace

#### 1. Spuštění

Aplikaci spustíte rozkliknutím .exe souboru. Zobrazí se jediné okno s aplikací (doporučuji maximalizovat).

## 2. Výběr linky

Do textového pole pod jedním z tlačítek 'Vyber linku / Vyber linku (protisměr)' napíšete relativní cestu k souboru s linkou (relativní oproti umístění aplikace) a stisknete příslušné tlačítko. Pokud daný soubor existuje a je ve správném formátu, linka by se měla načíst v pořádku a můžete pokračovat dále. Pokud máte soubor uložený jako '.txt' a zapomněli jste tuto příponu přidat, připiše se automaticky. Pokud načtení neproběhne správně, aplikace by se neměla rozbít - pouze vypíše chybu a můžete načítat znovu. Tlačítka pro protisměr nalevo fungují nezávisle na tlačítkách napravo - rozdíl je jen ve vypisování, popsáno dále.

## 3. Načtení spojů

Při stisknutí tlačítka pro načtení spojů pro určitý typ dne začne aplikace vyhledávat data v záhlaví spojů, a to zdali obsahují příslušnou značku. Běžná praxe v Česku je označovat pracovní dny značkou 'X', soboty značkou '6' a neděle+svátky značkou '+'. Tak tomu bývá alespoň u krajských či městských linek, pro které má grafikon největší smysl.

Jinak jsou řešeny školní prázdniny. V celostátní databázi JŘ se totiž necharakterizují spoje jako 'školní/prázdninové', ale pouze negativní značkou, příp. se vyvěsí samostatný jízdní řád (např. PID).

Z tohoto důvodu jsou v ovládacím oknu připravena dvě textová pole, kam můžete příslušné značky napsat (oddělené mezerou). Značky v dolním okně slouží pro uvedení spojů, které jezdí v běžné pracovní dny, ale ne o prázdninách. Naopak Značky v horním okně slouží pro uvedení spojů, které jezdí pouze o prázdninách (tedy nikoli v 'běžné' pracovní dny).

Pokud jsou spoje načteny podle očekávání, odemknou se zaškrťovací políčka pro výběr spojů pro vykreslení (viz 'Vykreslení').

## 4. Načtení vzdáleností (nepovinné)

Použitím tlačítka pro načtení vzdáleností načtete vzdálenosti pro různé trasy spojů (vždy označené 'km' v záhlaví), ovšem s malou odlišností oproti skutečnému souboru.

Jelikož se u zastávek jedná o tarifní (tzn. pro výpočet ceny) vzdálenost a ne skutečnou, bývají vzdálenosti zaokrouhleny na celé kilometry. To může vést k tomu, že více zastávek bude mít uvedeno stejnou vzdálenost. Při vykreslování je však potřeba, aby zastávky mezi sebou měly nějaký rozestup, jinak by splývaly. Tento rozestup můžete změnit v poli označeném 'Rozestup mezi zast.' Při opětovném načtení vzdáleností pak bude dodržován tento (změněný) rozestup.

Pro ilustraci: Jsou-li vzdálenosti zastávek '0 - 3 - 3 - 3 - 5' a rozestup je 0,1, pak budou upravené vzdálenosti vypadat následovně: '0 - 2,9 - 3 - 3,1 - 5'. Ve výsledném grafikonu se ovšem zaokrouhlují na celá čísla, takže uvedené vzdálenosti budou '0 - 3 - 3 - 3 - 5', jen bude změněna vzdálenost mezi body v grafu.

Po načtení tras je pak možno hýbat s posuvníkem, jehož rozsah hodnot závisí na počtu různých tras. Trasy se však budou vykreslovat pouze při zaškrtnutí příslušného čtverečku.

#### 5. Vykreslení grafu (jeden směr)

Pro vykreslení grafu (grafikonu) je nutno zaškrtnout dny, které chceme vykreslit, v checkboxech (zaškrťovacích čtverečcích) ‘Pracovní dny / Školní prázdniny / Sobota / Neděle’. Chceme-li znát vzdálenosti pro určité trasy, zaškrtneme checkbox u tlačítka ‘Načíst vzdálenosti’ a posuvník posuneme na příslušnou trasu (o jakou trasu se konkrétně jedná, zjistíme až po vykreslení, nebo přečtení jízdního řádu).

Pokud je vybraný některý den, pak se zobrazí graf, na jehož ose X jsou uvedeny hodiny a na ose Y jsou uvedeny zastávky. Není-li vybraná trasa, pak jsou uvedeny všechny zastávky v jízdním řádu s uniformními rozestupy. Pokud používáme pravá tlačítka, pak jsou uspořádány zdola nahoru (tedy opačně, než v jízdním řádu), v případě levých tlačítek jsou pak uspořádány shora dolů - zde může být název ‘protisměr’ poněkud matoucí.

V případě, že je vybraná trasa, pak jsou uvedeny pouze zastávky ležící na dané trase a nalevo od názvu zastávky je uvedena příslušná vzdálenost. Jak je uvedeno v sekci ‘Načtení vzdálenosti’, nemusí odpovídat tarifním ani reálným vzdálenostem - v případě malého rozestupu ovšem zpravidla budou odpovídat.

Každý ze spojů je pak reprezentován jako lomená čára odpovídající barvy (spoje stejného typu dne mají stejnou barvu). Pokud spoje splývají (např. sobota+neděle), jedna barva většinou nepřekrývá druhou úplně dokonale, což je spíš výhoda (dá se poznat, který spoj jede v sobotu i neděli, který pouze v sobotu, a který pouze v neděli). Spoje vykreslené pomocí pravého tlačítka tak představují rostoucí čáry, spoje vykreslené pomocí levého tlačítka představují klesající čáry.

Zastávky jsou reprezentovány červenými body. Pokud spoj danou zastávku projíždí nebo jede jinudy, bod není vykreslen.

Poněkud komplikovanější situace nastává v případě, kdy je vybraná konkrétní trasa - vykreslují se totiž i spoje, které danou trasu přímo nenásledují, ale sdílí několik zastávek (byť i jedinou - pak jsou takové spoje zobrazeny jako samostatné body). Do budoucna plánuji připravit možnost vybrat výlučně spoje kopírující danou trasu.

#### 6. Vykreslení grafu (oba směry)

Pokud do aplikace nahrajete jízdní řády obou směrů téže linky, pak je možno vykreslit oba najednou. Aby tato funkce fungovala, je potřeba, aby oba jízdní řády měly skutečně opačné směry zastávek. U běžných jízdních řádů je tato podmínka splněna automaticky, ovšem někdy se stává, že dvojice okružních linek má přímo protisměrné zastávky, je tedy možno použít i tyto dvojice JŘ.

Tlačítko ‘Vykresli obě’pak ukáže grafy spojů v obou směrech najednou. Algoritmus vykreslování pro tuto možnost není bohužel optimální, takže vždy se vykreslí pouze verze bez udání vzdálenosti (tras).

#### 7. Zvětšení grafu

Windowsí grafy disponují možností přiblížení. Horizontálním takžením přes část grafu získáte přiblížení na danou část. Pohybovat se pak můžete pomocí posuvníku pod osou X, případně se vrátit zpět pomocí tlačítka v levém dolním rohu.

Pokud by toto nestačilo, je možno zvětšit celé okno pomocí posuvníku ‘Změna velikosti grafu’. Tato změna je ovšem poměrně radikální - na  $n$ -té pozici se velikost okna změní  $n$ -krát (tedy výška i šířka  $\sqrt{n}$ -krát) při dalším vykreslení.

#### 8. Vytisknutí grafu

Tlačítko ‘Export jako PDF’ otevře dialogové okno s tisknutím - graf je tak možno uložit jako PDF nebo přímo vytisknout.

## 2 Programátorská dokumentace

### 2.1 Struktura programu

- Jedná se o běžný projekt okenní aplikace WinForms v jazyce C#.
- Většina ovládacích prvků se nachází v části “Návrh” a jejich vlastnosti se po spuštění programu nemění jinak než na příkaz uživatele.
- Obslužné funkce tlačítek a většina logiky programu se nechází v souboru “Form1.cs”.
- Část sloužící pro načtení vstupního souboru linky se nachází v souboru “SheetLoader.cs”.
- Část pro rozdělení jízdního řádu je v souboru “TimeTableParser.cs”.
- Část pro přiřazování zastávek a vzdáleností k jednotlivým spojům je v souboru “ConnectionGroup.cs”.

### 2.2 Soubor SheetLoader.cs

Kód v tomto souboru se skládá pouze z jedné třídy SheetLoader.

#### 1. ReadExcelInput

Tato funkce načte vstupní soubor ve formátu CSV (oddělovač podle parametru *separator*), přičemž přeskočí prázdné řádky.

## 2. RowifyTable

Pomocí této funkce se načtená tabulka transponuje a případně se zarovnají řádky (přidáním prázdných prvků pole) tak, aby tabulka byla obdélníková.

Tabulka je transponovaná kvůli tomu, že jízdní řády se lépe analyzují po sloupcích (a to platí jak pro časové údaje spojů, tak názvy zastávek a vzdálenosti)

## 2.3 Soubor TimeTableParser.cs

Soubor obsahuje opět jen jednu třídu, TimeTableParser. Jsou zde hard-codované značky pro pracovní dny, soboty, neděle atd. V budoucnu je možno vylepšit toto umožněním uživatelům, aby si dané značky nastavili sami.

### 1. Konstruktor

Třída potřebuje v konstruktoru transponovaný jízdní řád a poté případné seznam negativních značek pro prázdniny. Jízdní řád je získán z předchozí třídy, značky jsou brány jako celý obsah textového pole, do kterého je píše uživatel.

### 2. Cutout

Tato funkce redukuje jízdní řád na pouze ty řádky, kde by se měla nacházet relevantní data pro jízdní řád. Spoléhá se na to, že v levém horním rohu bude značka "Tě", poté jede volný řádek, a pod touto značkou pak tarifní čísla jednotlivých zastávek.

### 3. GetKilometrageTable

Udělá tabulku pouze s řádky (orig. sloupci) obsahující položku "km" a názvy zastávek. Názvy zastávek jsou vždy jako první řádek.

### 4. ExtractKilometragesFromTable

Z kilometrických údajů v jízdním řádu (které jsou jako string) vytvoří pole polí čísel jakožto vzdáleností.

V první fázi se řetězcové údaje přetvoří na číselné údaje. Pokud je údaj prázdný nebo je uvedeno, že spoj přes zastávku nejede (značkou "j" v kilometrových údajích), pak je tato hodnota zapsána jako -1 - záporné údaje jsou ignorovány (pochoptelně, tarifní vzdálenost je nezáporná, takže v normálním JŘ by se záporná čísla objevovat neměla).

Poté jsou tyto kilometrické údaje tzv. normalizovány - jednak jsou přeškálovány podle určité maximální vzdálenosti (funkci momentálně nepoužívám, ale může se hodit) a dále jsou vzdálenosti posunuty tak, aby mezi zastávkami byly dané rozestupy (více v popisu příslušného souboru ConnectionGroup.cs)

### 5. CreateStopsAndConnections

Parsuje část jízdního řádu se zastávkami a časy. První dva řádky oříznutého JŘ jsou určeny pro záhlaví - u zastávek se přeskočí, u spojů je první řádek

uveden pro označení spoje (např. "Spoj 123"), které je pak použito jako klíč ve slovníku spojů.

#### 6. CreateWorkdayTable (apod.)

Z jízdního řádu vyřízne tu část, která obsahuje názvy zastávek a pro spoje, které jedou v příslušné dny - ty se poznají podle záhlaví v prvním řádku (značky "X" apod.) Význam značek je uveden v uživatelské dokumentaci, pro rozšíření programu je dobré navrhnout nějakou flexibilitu ve volbě značek.

## 2.4 Soubor ConnectionGroup.cs

Soubor obsahuje třídu ConnectionGroup (jako skupina spojů) , Stop (pro záznamy zastávek), statickou třídu ArrayExtensions pro obecné operace s poli (nebudu rozepisovat) a ArrayCalculations, která slouží pro číselné operace, především úpravy tarifních vzdáleností.

#### 1. Třída ConnectionGroup

Tato třída reprezentuje seznam spojů stejného typu; v proměnné Connections jsou uvedeny názvy spojů a seznam časových údajů pro daný název.

Ostatní údaje slouží pro to, jak se bodu dané spoje vykreslovat v grafu - tzn. jakou barvou, zdola nahoru (direction) a zdali vůbec (enabled)

#### 2. Struktura Stop

Tato struktura obsahuje pouze dvě hodnoty: Order (pořadí = tarifní číslo) a Distance (vzdálenost). Název zastávky zde není uložen a dohledává se pomocí tarifního čísla.

#### 3. ArrayCalculations.MirrorPositives

Přepíše všechny kladné hodnoty v poli na doplněk největšího čísla - využívá se pro výpočet vzdáleností u spojů v protisměru.

#### 4. ArrayCalculations.Normalize

Přeskáluje hodnoty v poli (nezáporné hodnoty musí být uspořádané), a to tím způsobem, že největší hodnota v poli bude rovna zadané hodnotě "maximumDistance" (je-li hodnota tohoto parametru nekladná, škálování se neodehrává), a přitom minimální rozdíl mezi sousedními prvky je roven hodnotě minDiff(po škálování).

K vynucení minimálního rozdílu slouží níže zmíněná funkce - ta však nemusí vyřešit všechno, proto hodnoty posouvám směrem nahoru, není-li rozestup ddržen dokonale. (Např. z 0 - 2 - 2 - 3 a rozestupem 1 bych vytvořil 0 - 2 - 3 - 4, pokud bych nepoužíval algoritmus NormalizeA)

## 5. ArrayCalculations.NormalizeA

Tato funkce vytvoří žádané výše zmíněné rozestupy. Ukazatele "firstInd" a "secondInd" skenují pole, a pokud je rozdíl mezi sousedními prvky malý, ukazatel secondInd se posunuje dále. Jakmile je označen vhodný úsek, spočítá se průměr daného úseku a hodnoty v daném úseku se posunou tak, aby průměr úseku byl zachován a stejně tak dané rozestupy. Další úsek pak začíná na prvním prvku "spravovaného" úseku. Pokud se doběhne na konec pole a bylo třeba něco spravovat, začíná další iterace.

V některých situacích však algoritmus nemusí skončit - je-li rozdíl mezi prvním a posledním prvkem malý, rozestupy jsou velké a prvků je hodně. Z tohoto důvodu algoritmus (pro výpočty vzdáleností zastávek) končím hned po první iteraci a případné nedodělky spravuji tím, že následující číslo přímo zvýším tak, aby byl dodržen rozestup (a neohlížím se tak zpět).

## 2.5 Soubor Form1.cs

Soubor s ústřední třídou celé aplikace. Přiznávám, že je momentálně trochu chaotický. Třída Form1 obsahuje datové položky, vlastnosti, algoritmy vykreslování i obslužné události po kliknutí na jednotlivá tlačítka.

### 1. Datové položky a vlastnosti

Některé datové položky jsou uvedeny jako konstanty (*const* nebo *static readonly*) - do budoucna je určitě je možné k nim přiřadit nástroje pro uživatelské rozhraní, ať si je může nastavit uživatel dle svého uvážení (např. barvy nebo velikost bodů zastávek).

Položky, které jsou přiřazeny k uživatelskému rozhraní (např. min. vzdálenost mezi zastávkami) jsou uvedeny jako vlastnosti jen pro čtení - dle mého názoru logický přístup, ale lidé zkušenější v programování UI si mohou myslet něco jiného.

Položky StopsF/StopsB... slouží pro uchovávání dat z načtených JŘ (viz popis funkcí tlačítek, a funkcí z jiných souborů)

### 2. Události kliknutí

Většina událostí je popsána zvlášť pro tlačítko pro směr tam i zpět - do budoucna by určitě bylo dobré, aby tyto události volaly tutéž funkci rozlišenou např. parametrem 0/1.

#### (a) btnChooseLine\_Click

Načte uvedený JŘ do zmíněných proměnných.

#### (b) btnWorkday\_Click (ap.)

Ořízne JŘ a vytáhne z něj příslušné spoje. V případě úspěšného načtení se odemkne checkbox pro uživatele (takže dané spoje lze připravit k vykreslení).

(c) `btnLoadDistsF_Click`

Načte možné trasy a přizpůsobí rozsah posuvníku pro výběr tras.

(d) `chbWorkday_CheckedChanged`

Při zaškrtnutí příslušného checkboxu se změní typ skupiny spojů, zdali se bude vykreslovat či nikoli. Místo události by se také dalo napsat pomocí přístupové vlastnosti u jednotlivé skupiny spojů, já jsem se rozhodl pro momentální řešení.

(e) `btnRenderFront_Click`

Pokud nebyly žádné spoje vybrány nebo načteny, vyčistí okno (s grafem) a nic se nestane. Jinak zavolá algoritmus pro vykreslování s příslušnými parametry (viz níže). Parametry je myšleno, zdali se bude vykreslovat konkrétní trasa (podle vzdáleností) nebo všechny zastávky, a poté pro který směr se budou spoje vykreslovat.

(f) `btnExport_Click`

Otevře dialog tisknutí. Funkce tisknutí je přidána pouze v základní verzi, určitě by se hodilo přidat i možnost vytisknutí ve větším formátu.

### 3. Metody (funkce)

(a) `ClearGraphicon`

Nastaví okno grafu a vyčistí všechny linie.

Sem by měly přijít všechny akce, které nějakým způsobem nastavují vlastnosti okna grafu (tzn. s výjimkou samotných spojů a zastávek). Tato metoda se volá při každém překreslování grafu, a slouží i pro update nastavení od uživatele - veškerá nastavení se projevují až při novém vykreslování grafu, nikoli na již vykresleném.

(b) `RenderGraphicon`

Tato metoda je onou ústřední, která přidá všechny vybrané skupiny spojů do grafu.

Metoda nejprve nastaví graf, zejména přidání zastávek na osu Y (podle předloženého seznamu zastávek a vzdáleností). V případě vykreslování obou směrů je seznam vzdáleností ignorován - sub-algoritmy na toto ještě nejsou ideálně stavěné.

Poté se přejde k samotnému výpočtu jednotlivých bodů a čar (Series), dle níže uvedených metod. Skupiny spojů, které nebyly načteny nebo vybrány, jsou přirozeně ignorovány. Zde však ještě zmíním samotné vykreslování zastávek.

Vzdálenosti jsem nijak neškáloval, takže graf bude vždy v okně roztažený odshora dolů pro všechny zastávky.

Důležité jsou parametry *labelRange* a *lineWidth*. Jelikož se mi nepovedlo přiřadit text k jednomu konkrétnímu bodu na ose Y, tak *labelWidth* určuje, jakou výšku bude zabírat daný text. Z důvodu zaokrouhlovacích chyb je pak od šířky ještě odečtena hodnota 0.01.



Od každé zastávky pak ještě vede horizontální čára (*StripLine*) přes budoucí body (časy zastavení spojů). Důležité je, že šířka této čáry roste lineárně s maximální vzdáleností. Kratší vzdálenost trasy totiž znamená více roztažený graf (aby se vlezl do okna), a díky této úpravě by si čára měla zachovat konstatní velikost.

(c) `MakeStopSeriesForTable`

Vytvoří čáry pro jednu skupinu spojů. Jednomu spoji odpovídá jedna čára (*Series*), přičemž parametry čar, s výjimkou bodů, jsou pro jednu skupinu spojů totožné.

(d) `AddConnection`

Vytvoří body pro danou sérii.

Až zde se konvertují časové údaje na minutové údaje (což může být slabina aplikace, jsou-li jízdní řády zadány nesprávně).

Zde se rozhoduje, jak je jízdní řád orientovaný (směrem nahoru nebo dolů) a podle toho se rozhoduje, v jaké výšce bude daný bod zaznamenán (tzn. jaké zastávce odpovídá).

Ukazuje se, že v případě obousměrného řádu s danými vzdálenostmi (trasami) toto rozhodování nestačí - tzn. není možno nastavit zastávky (funkce *ClearGraphicon*) a body (funkce *AddConnection*) aby bylo pokryto všech 7 případů.

Z tohoto důvodu nejsou 2 případy (vzdálenosti zpět + obousměrný / vzdálenosti vpřed + obousměrný) naimplementovány.

©František Mrkus, 2021