



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Aplikace pro automatické vytváření rozvrhu pro základní školy

Bakalářská práce

Studijní program:

B2646 Informační technologie

Studijní obor:

Informační technologie

Autor práce:

Martin Kracman

Vedoucí práce:

Ing. Josef Chudoba, Ph.D.

Ústav nových technologií a aplikované informatiky





Zadání bakalářské práce

Aplikace pro automatické vytváření rozvrhu pro základní školy

Jméno a příjmení: **Martin Kracman**
Osobní číslo: M18000084
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: 2021/2022

Zásady pro vypracování:

1. Popište faktory ovlivňující tvorbu rozvrhů, např. požadavky na vyučující, místnosti a zákonné předpoklady.
2. Popište algoritmus provádějící školní rozvrh na základní škole.
3. Vytvořte aplikaci, která bude automaticky řešit problematiku rozvrhu pro obecnou základní školu.
4. Ověřte aplikaci na datech z reálné základní školy.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby dokumentace
30-40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] Dong Fengming, Eng Guan Tay a Khee Meng Koh. Introduction to graph theory: h3 mathematics [online].
- [2] Singapur: World Scientific Publishing Co. Pte., 2007 [cit. 2021-04-04]. ISBN 10 981-270-525-2.
- [3] Černý, J. (2013). Základní grafové algoritmy. Praha: České vysoké učení technické v Praze. ISBN 978-80-01-05258-7.
- [4] C# 8.0 Pocket Reference: Instant Help for C# 8.0 Programmers. O'Reilly, 2019. ISBN 9781492051213.

Vedoucí práce:

Ing. Josef Chudoba, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce:

12. října 2021

Předpokládaný termín odevzdání:

16. května 2022

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2021

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

3. května 2022

Martin Kracman

Poděkování

Tímto bych rád poděkoval vedoucímu své bakalářské práce Ing. Josefu Chudobovi, Ph.D. za častou pomoc při konzultacích, rady a připomínky při tvorbě této práce.

Dále bych rád poděkoval za umožnění spolupráce a konzultace se školou ZŠ Oblačná ředitelce školy Ing. Aleně Routové a zástupkyni ředitelky Mgr. Elišce Handšuhové.

Anotace

Cílem bakalářské práce „Aplikace pro automatické vytváření rozvrhu pro základní školy“ je vytvořit aplikaci pro automatické generování rozvrhu a následně vybrat reálnou základní školu a ověřit na ní funkčnost aplikace. Bakalářská práce se v teoretické části zabývá obecnými faktory škol, které ovlivňují tvorbu rozvrhu a porovnáním existujících aplikací, které dokáží vytvářet rozvrhy. K praktické části byla vytvořena desktopová aplikace, kterou lze vytvářet, načítat a ukládat vstupní soubory potřebné pro vytváření rozvrhu, generovat řešení rozvrhu a ukládat vygenerované řešení do šesti výstupních souborů ve formátu CSV a XLSX určených pro zobrazení výsledného rozvrhu z pohledu žáka, učitele a místnosti. Aplikace byla otestována na datech z reálné školy ZŠ Oblačná a na druhé uměle vytvořené škole. Aplikace našla funkční řešení pro obě testované školy. V bakalářské práci je popsáno, jak vytvořené algoritmy řeší generování rozvrhu, jejich vstupní a výstupní soubory. Dále práce popisuje vytvořenou aplikaci a testování aplikace.

Klíčová slova: visual studio, barvení grafu, školní rozvrh

Abstract

The aim of the bachelor thesis "Application for automatic schedule creation for elementary schools" is to create an application for automatic school schedule generation and then choose a real elementary school to verify the functionality of the application. In the theoretical part the bachelor thesis deals with the general factors of schools that affect the creation of school schedules and the comparison of existing applications that create schedules. For the practical part, a desktop application was created that can create, load and save input files, which are needed to generate school schedule solutions, generate a solution and save the generated solution in six output files in CSV and XLSX formats, which are designed to display the final schedule from the perspective of the student, teacher and room. The application was tested on the data from a real school, ZŠ Oblačná, and at second artificially created school. The application found a functional solution for both tested schools. The bachelor thesis describes how the created algorithms solve the generation of the schedule, their input and output files. The bachelor thesis also describes the created application and testing of the application.

Key words: visual studio, greedy coloring algorithm, school schedule

Obsah

1	Úvod.....	11
2	Obecné faktory ovlivňující tvorbu rozvrhu	12
2.1	Nutná omezení pro tvorbu rozvrhu.....	12
2.2	Zákony omezující tvorbu rozvrhů.....	12
2.3	Požadavky na rozvrh od učitelů a studentů	12
3	Porovnání aplikací na trhu	14
3.1	Asc Timetables	14
3.2	Prime Timerable	15
3.3	Aplikace FET	15
4	Vstupní soubory aplikace.....	18
4.1	Textový soubor „skola.txt“	18
4.2	Textový soubor „místnosti.txt“	19
4.3	Textový soubor „tridy.txt“	19
4.4	Textový soubor „ucitele.txt“	20
4.5	Textový soubor „predmety.txt“	20
5	Algoritmus hledání rozvrhu	23
5.1	Přidělování finálních místností k předmětům	23
5.2	Přidělování finálních učitelů k předmětům.....	25
5.3	Označování časů, ve kterých se předmět nemůže vyučovat.....	27
5.4	Přiřazování časů k vyučovacím blokům	28
5.5	Vyhodnocování výsledného rozvrhu	34
6	Výstupní soubory aplikace.....	40
6.1	Výstupní soubory pro žáky	40
6.2	Výstupní soubory pro učitele.....	41
6.3	Výstupní soubory pro místnosti.....	42
7	Aplikace Rozvrh	44
7.1	Spuštění aplikace	44
7.2	Popis aplikace	44
8	Testování aplikace	50
8.1	ZŠ Oblačná	50
8.2	Škola vytvořená autorem práce.....	53
9	Závěr	57

Seznam obrázků a tabulek

Obrázky:

Obrázek 1: Ukázka dat ze souboru „skola.txt“	18
Obrázek 2: Ukázka dat ze souboru „místnosti.txt“	19
Obrázek 3: Ukázka dat ze souboru „tridy.txt“	19
Obrázek 4: Ukázka dat ze souboru „ucitele.txt“	20
Obrázek 5: Ukázka dat ze souboru „predmety.txt“	22
Obrázek 6: Vývojový diagram algoritmu pro přidělování finálních místností	24
Obrázek 7: Vývojový diagram algoritmu pro přidělování finálních učitelů	26
Obrázek 8: Vývojový diagram přípravy dat pro použití rekurzivního algoritmu	30
Obrázek 9: Vývojový diagram rekurzivní funkce pro přiřazování barev.....	31
Obrázek 10: Vývojový diagram první části hodnotícího algoritmu	35
Obrázek 11: Vývojový diagram druhé části hodnotícího algoritmu	36
Obrázek 12: Vývojový diagram třetí části hodnotícího algoritmu	38
Obrázek 13: Výstup do CSV souboru z pohledu žáka	40
Obrázek 14: Výstup do XLSX souboru z pohledu žáka.....	41
Obrázek 15: Výstup do CSV souboru z pohledu učitele.....	41
Obrázek 16: Výstup do XLSX souboru z pohledu učitele	42
Obrázek 17: Výstup do CSV souboru z pohledu místnosti.....	42
Obrázek 18: Výstup do XLSX souboru z pohledu místností	43
Obrázek 19: Záložka aplikace „Škola“	44
Obrázek 20: Záložka aplikace „Místnosti“	45
Obrázek 21: Záložka aplikace „Vyučující“	46
Obrázek 22: Záložka aplikace „Třídy“	47
Obrázek 23: Záložka aplikace „Předměty“	48
Obrázek 24: Záložka aplikace „Rozvrh“	49
Obrázek 25: Plán budovy ZŠ Oblačná. Převzato z [5].....	50

Tabulky:

Tabulka 1: Tabulka funkcí pro tvorbu rozvrhu. [10] [11] [14]	17
Tabulka 2: Tabulka výsledků přidělení místností (ZŠ Oblačná)	52
Tabulka 3: Tabulka výsledků přidělení učitelů (ZŠ Oblačná).....	53
Tabulka 4: Tabulka výsledků přidělení místností (škola vytvořená autorem práce)	54
Tabulka 5: Tabulka výsledků přidělení učitelů (škola vytvořená autorem práce).....	55

Seznam zkratk

Seznam zkratk použitých v kapitole 3:

- FET** Free Timetabling Software, desktopová aplikace
CZK česká koruna
USD americký dolar
 A_i aktivita, kterou je možné umístit do výsledného rozvrhu
 T_j časový úsek v rozvrhu, do kterého je možné umístit aktivitu A_i
 N_A počet aktivit A_i

Seznam zkratk použitých v kapitole 4.5:

- V** počet vyučujících, kteří mohou vyučovat předmět
M počet místností, kde může být vyučován předmět
 V_1 počet vyučujících, kteří mohou vyučovat první předmět
 M_1 počet místností, kde může být vyučován první předmět
 V_2 počet vyučujících, kteří mohou vyučovat druhý předmět
 M_2 počet místností, kde může být vyučován druhý předmět

Seznam zkratk použitých v kapitole 5.1:

- P** předmět konkrétní skupiny osob
 L_{pm} list místností vhodných k vyučování předmětu **P**
 $L_{mís}$ list všech místností na škole
 $M_{mís}$ místnost v listu $L_{mís}$
 $D_{mís}$ číselná hodnota reprezentující důležitost místnosti $M_{mís}$ v rozvrhu
 M_n místnost z listu $L_{mís}$, která má nejmenší hodnotu důležitosti $D_{mís}$
 P_{nm} předmět **P**, který má největší počet vyučovacích bloků, vybraný ze všech předmětů **P**, které nemají přiřazenou finální místnost a zároveň mají místnost M_n zapsanou v listu místností L_{pm}

Seznam zkratk použitých v kapitole 5.2:

- P** předmět konkrétní skupiny osob
 L_{pu} list učitelů vhodných k vyučování předmětu **P**
 $L_{uč}$ list všech učitelů na škole
 $U_{uč}$ učitel v listu $L_{uč}$
 $D_{uč}$ číselná hodnota reprezentující důležitost učitele $U_{uč}$ v rozvrhu
 U_n učitel z listu $L_{uč}$, který má nejmenší hodnotu důležitosti $D_{uč}$
 P_{nu} předmět **P**, který má největší počet vyučovacích bloků, vybraný ze všech předmětů **P**, které nemají přiřazeného finálního učitele a zároveň mají učitele U_n zapsaného v listu učitelů L_{pu}

Seznam zkratk použitých v kapitole 5.3:

- L** list všech vyučovacích bloků na škole
- B** vyučovací blok v listu **L**
- P_b** pole vyučovacího bloku **B**

Seznam zkratk použitých v kapitole 5.4:

- M_s** zjednodušená matice sousednosti
- N** je počet všech vstupních vyučovacích bloků v rozvrhu
- B** jeden vyučovací blok konkrétní skupiny osob
- C** časový blok k vyučování
- barva** celé číslo mezi 0 až 49 odkazující na konkrétní časový blok **C**
- L** list s **N** vyučovacími bloky **B**
- P_{hran}** počet hran konkrétního bloku **B** v listu **L**
- P_{bar}** počet **barev**, které nelze přiřadit konkrétnímu bloku **B** v listu **L**
- P_b** pole vyučovacího bloku **B** (viz 5.3)
- P_{rek}** počet rekurzí
- K_{alg}** **bool** hodnota konec algoritmu
- P_{fun}** počet nalezených funkčních řešení rozvrhu
- P_{nefun}** počet nalezených nefunkčních řešení rozvrhu
- B_{bez}** blok **B** v listu **L**, který má přidělenou **barvu** -1
- B_{vyb}** vybraný blok v rekurzi číslo **P_{rek}**
- L_{bar}** list **barev**, které se pokusí algoritmus bloku **B_{vyb}** v rekurzi **P_{rek}** přiřadit
- barva_{fin}** **barva** z listu **L_{bar}**, kterou algoritmus přiřadí bloku **B_{vyb}**
- B_{spol}** vyučovací blok **B** z listu **L**, který má společnou hranu s blokem **B_{vyb}**
- I_{přič}** hodnota **bool**, která určuje, jestli se u hodnoty **P_{bar}** bloku **B_{spol}** v rekurzi číslo **P_{rek}** přičetlo číslo 1

Seznam zkratk použitých v kapitole 5.5:

- L_{uč}** list všech učitelů na škole
- U_{uč}** učitel v listu **L_{uč}**
- V_u** vyučovací den učitele **U_{uč}** v týdnu
- H_b** hodnota po sobě jdoucích vyučovaných bloků
- B_{vu}** časový blok vyučovacího dne **V_u**
- L_{tř}** list všech tříd na škole
- T** třída v listu **L_{tř}**
- V_t** vyučovací den třídy **T** v týdnu
- B_{vt}** časový blok vyučovacího dne **V_t**
- B_n** časový blok, který následuje za blokem **B_{vt}** ve dni **V_t**
- H** hodnota pro odečtení od celkového hodnocení rozvrhu

Seznam zkratk použitých v kapitole 6:

- XLSX** Microsoft Excel Open XML Spreadsheet, formát souboru
- CSV** Comma separated values, formát souboru

1 Úvod

Pro provoz základních škol se musí, například z důvodu odchodů a příchodů učitelů, změn vyučovacích osnov a počtů tříd v ročníku, většinou každý školní rok vytvořit nový školní rozvrh. Ten určí, jak bude výuka jednotlivých předmětů členěna do vyučovacích bloků. Pro školy, které dodnes řeší vytváření rozvrhu ručně, bez použití jakékoliv aplikace nebo počítače, může vytvoření rozvrhu představovat velký problém. Ruční přístup je pomalý a velmi jednoduše lze lidskou chybou vytvořit nefunkční rozvrh.

Cílem bakalářské práce je vytvoření aplikace pro automatické vytváření rozvrhu pro základní školy. Jako konkrétní škola k otestování aplikace, byla vybrána základní škola Oblačná.

Bakalářská práce je členěna do několika kapitol.

- **Kapitola 2:** Obecné faktory, které ovlivňují tvorbu rozvrhu, například nutná omezení pro sestavení funkčního rozvrhu a zákony omezující tvorbu rozvrhu.
- **Kapitola 3:** Porovnání existujících aplikací dnešního trhu pro automatické vytváření rozvrhu (aplikace Asc Timetables, Prime Timetable a aplikace FET).
- **Kapitola 4:** Vstupní soubory aplikace, jejich formát a omezení vstupních dat.
- **Kapitola 5:** Algoritmus hledání rozvrhu (přidělování finálních místností a učitelů k předmětům, označování časů, ve které se předmět nemůže vyučovat, přiřazování časů k vyučovacím blokům, vyhodnocování výsledného rozvrhu).
- **Kapitola 6:** Výstupní soubory aplikace (výstupní soubory pro rozvrhy z pohledu žáků, učitelů a místností).
- **Kapitola 7:** Spuštění a popis aplikace.
- **Kapitola 8:** Testování aplikace (ZŠ Oblačná, škola vytvořená autorem práce)

2 Obecné faktory ovlivňující tvorbu rozvrhu

2.1 Nutná omezení pro tvorbu rozvrhu

Pro úspěšné a funkční sestavení rozvrhu je nutné, aby došlo k dodržení následujících skutečností, které brání vytváření konfliktů:

- Žádný učitel nesmí v jeden časový blok vyučovat více předmětů.
- Žádný žák nesmí mít v jeden časový blok více předmětů.
- V každé místnosti se v jednom časovém bloku může vyučovat pouze jeden předmět (pokud se ovšem nejedná o místnost, jejíž kapacita by to dovolovala).

[4]

2.2 Zákony omezující tvorbu rozvrhů

Výsledný školní rozvrh musí dodržovat ustanovení vyhlášky č. 48/2005 Sb., o základním vzdělávání a některých náležitostech plnění povinné školní docházky. Důležité je tedy, aby výsledný rozvrh dodržoval, že:

- § 1 odst. 1 Vyhl. 48/2005 Sb. „Vyučování začíná zpravidla v 8 hodin, nesmí však začínat dříve než v 7 hodin. Vyučování musí být ukončeno nejpozději do 17 hodin.“
- § 1 odst. 3 Vyhl. 48/2005 Sb. „Žáci mohou mít v dopoledním vyučování nejvýše 6 vyučovacích hodin a v odpoledním vyučování nejvýše 6 vyučovacích hodin. Konkrétní počet vyučovacích hodin stanoví škola s přihlédnutím k charakteru vzdělávací činnosti a k základním fyziologickým potřebám žáků.“
- § 1 odst. 5 Vyhl. 48/2005 Sb. „Přestávky mezi vyučovacími hodinami jsou nejméně desetiminutové. Během dopoledního vyučování, zpravidla po druhé vyučovací hodině, se zařazuje alespoň jedna přestávka v délce nejméně 15 minut. Přestávka mezi dopoledním a odpoledním vyučováním trvá nejméně 50 minut. V případech hodných zvláštního zřetele lze zkrátit některé desetiminutové přestávky na nejméně 5 minut a přestávku mezi dopoledním a odpoledním vyučováním na nejméně 30 minut. Při zkracování přestávek ředitel školy přihlédne k základním fyziologickým potřebám žáků.“ [1]

2.3 Požadavky na rozvrh od učitelů a studentů

Většina osob má na školní rozvrh mnoho požadavků. Vlastním průzkumem, kterého se zúčastnilo 6 osob, bylo zjištěno, jak by školní rozvrh měl správně vypadat podle představ

učitelů a studentů. Průzkumu se zúčastnili 2 vyučující ze ZŠ Dobiášova a 4 studenti z Technické univerzity v Liberci. Všem osobám byla položena otázka, jak by podle jejich představ měl vypadat rozvrh na základní škole a co by měl obsahovat. Mezi zmíněné požadavky patří:

- Prioritně by se měly vyučovat předměty jako matematika a český jazyk v prvních čtyřech vyučovacích blocích.
- V pátek a jeden další den v týdnu by žádný učitel neměl mít odpolední vyučování (mělo by se končit po šestém vyučovacím bloku a to především kvůli častým odpoledním schůzím učitelského sboru).
- Stejně předměty by měla mít každá třída pouze jednou denně.
- Kluci a holky by měli mít oddělený tělocvik.
- Učitelé by neměli učit více než 3 hodiny po sobě.
- Všechny třídy by neměly končit vyučování ve stejný čas a to kvůli tvořícím se frontám ve školní jídelně.
- Žáci by neměli mít příliš mnoho volných hodin mezi další výukou.
- Některé předměty (například fyzika, chemie a předměty cizích jazyků) by se měly vždy vyučovat ve specializovaných místnostech.
- Rozvrh by měl obsahovat čas, kdy začíná a končí vyučovací blok, zkratku vyučovaného předmětu a zároveň místnost, ve které se předmět vyučuje.

3 Porovnání aplikací na trhu

Mezi přední aplikace dnešního trhu pro automatické vytváření rozvrhu patří například Prime Timetable [6], dále asc Timetables [7] a Edookit [8], jehož součástí je generátor rozvrhů Skolaris [9].

K porovnání byly vybrány tři aplikace. Dvě nejvyužívanější placené (asc Timetables, Prime Timetable) a jedna bezplatná (FET). Porovnávány byly v oblastech:

- 1) Funkce pro tvorbu rozvrhu
- 2) Využívaný algoritmus
- 3) Jejich omezení
- 4) Cena softwaru

3.1 Asc Timetables

Odkaz na web: <https://www.asctimetables.com/>

Asc Timetables je desktopová aplikace. Zahrnuje verzi pro tvorbu rozvrhů na základních školách a rozšířené verze, které by měly stačit i pro vytváření rozvrhů na středních školách.[7]

Algoritmus

Informace ohledně implementovaného algoritmu nešly na oficiálních stránkách asc Timetables dohledat a na zasláný email s dotazem ohledně tohoto algoritmu, společnost neodpověděla.

Omezení aplikace

Požadavky na rozvrh s menší prioritou nebudou aplikací vyřešeny, pokud by kvůli nim nemohly být splněny požadavky s vysokou prioritou (priority požadavků si nastavuje aplikace). [10]

Cena

Cena aplikace pro vytváření rozvrhů na základní škole je 4 400 CZK. Je možné ji vyzkoušet před zaplacením a zaplatit až po vytvoření rozvrhu. [7]

3.2 Prime Timerable

Odkaz na web: <https://primetimetable.com/>

Prime Timetable je webová aplikace sloužící k manuálnímu i automatickému vytváření rozvrhů. Je vhodná pro všechny druhy vzdělávacích organizací. [6]

Algoritmus

Informace ohledně implementovaného algoritmu nešly na oficiálních stránkách Prime Timetable dohledat a na zasláný email s dotazem ohledně tohoto algoritmu společnost neodpověděla.

Omezení aplikace

Požadavky na rozvrh s menší prioritou nebudou aplikací vyřešeny, pokud by kvůli nim nemohly být splněny požadavky s vysokou prioritou. [11]

Cena

Aplikaci lze vyzkoušet zdarma na 30 dnů. Další užívání je zpoplatněno a rozděleno na dvě kategorie podle počtu učitelů na škole. Pro školy, kde působí méně než 41 učitelů je cena 299 USD za rok. Pro školy, kde působí méně než 101 učitelů je cena za rok 499 USD. [6]

3.3 Aplikace FET

Odkaz na web: <https://lalescu.ro/liviu/fet/>

FET (Free Timetabling Software) je desktopová aplikace s otevřeným zdrojovým kódem [13]. Dokáže vytvářet rozvrh pro základní, střední i vysoké školy. [12]

Algoritmus

Projekt FET byl spuštěn v roce 2002. Na začátku využíval k tvorbě genetický algoritmus, který se ale ukázal být příliš pomalý a neuměl vygenerovat řešení pro složitější rozvrhy. V roce 2007 byl zaveden heuristický algoritmus (rekurzivní swapovací algoritmus), který byl mnohem rychlejší a dokázal generovat i složitější rozvrhy. [15]

Popis rekurzivně swapovacího algoritmu

- 1) Algoritmus nejprve seřadí aktivity (aktivita, kterou je možné umístit do výsledného rozvrhu = A_i) od nejvíce omezených aktivit po nejméně omezené, čímž se algoritmus několikanásobně zrychlí.

- 2) Algoritmus zkusí umístit každou aktivitu A_i do povoleného časového úseku (časový úsek v rozvrhu, do kterého je možné umístit aktivitu $A_i = T_j$). A to postupně jednu po druhé podle seřazeného pořadí. Pokud je k dispozici více časových úseků T_j vybere náhodný z nich, do kterého A_i přiřadí. Pokud není k dispozici žádný, provede rekurzivní swapování:
- U každého časového úseku T_j zjistí a uchová seznam konfliktních aktivit, pokud by do něj aktivitu A_i uložil.
 - Vybere časový úsek T_j s nejnižším počtem konfliktních aktivit (například časový úsek označený T_{12} s konfliktními aktivitami označenými A_2 , A_5 a A_{23}).
 - Přiřadí aktivitu A_i do T_j a odstraní aktivitám A_2 , A_5 a A_{23} přiřazené časové úseky.
 - Rekurzivně zkusí přiřadit A_2 , A_5 a A_{23} jako v kroku 2) (Pokud není úroveň rekurze příliš vysoká, například 14 a pokud celkový počet rekurzivních úrovní počítaných od začátku kroku 2) na A_i není příliš velký, například $2 \cdot N_A$, kdy N_A je počet aktivit A_i).
 - Pokud se úspěšně přiřadí A_2 , A_5 a A_{23} časovým úsekům T_j , tak se algoritmus vrátí s úspěšným přiřazením do 2), jinak se vrátí na b) a vybere jiný časový úsek.
 - Pokud algoritmus vyzkouší všechny (nebo jinak nadefinovaný počet) časových úseků, vrátí se bez úspěchu.
 - Pokud je algoritmus na úrovni rekurze 0 a při umístění A_i neměl úspěch, umístí A_i jako v krocích b) a c) ale bez rekurze. Nyní algoritmus připíše do seznamu aktivit 3 další aktivity (a to například A_2 , A_5 a A_{23}) a přechází na krok 2) (zde algoritmus používá některé metody, které umožňují se vyhnout zacyklení).

Výše uvedený algoritmus je volně přeložen z [15].

Omezení aplikace

Aplikace nemusí najít dokonalé řešení pro výsledný rozvrh. Uživatel si zde vlastnoručně nastavuje prioritu (váhu) jednotlivých požadavků na rozvrh. Pokud aplikace není schopna najít dokonalé řešení, snaží se najít takové, které má nejlepší hodnocení podle uživatelsky zadaných priorit. Uživatel také nemůže jednoduše ručně přesouvat hodiny ve vygenerovaném rozvrhu. [14]

Cena

Aplikace FET je bezplatná. [12]

Funkce pro tvorbu rozvrhu

V následující tabulce je uvedeno, jaké funkce mají jednotlivé aplikace.

Funkce	Asc Timetables	Prime Timetable	FET aplikace
Definovat základní informace o škole	Ano	Ano	Ano
Přidávat a upravovat: předměty, třídy, učitele, hodiny a místnosti	Ano	Ano	Ano
Vytvářet skupiny a spojené třídy	Ano	Ano	Ano
Upravovat počty hodin a jejich jednotlivé časy pro konkrétní třídy v konkrétní dny	Ano	Ano	Ano
Možnost učit i o víkendu a vytvářet delší než týdenní rozvrhy	Ano	Ano	Ano
Vytvářet aktivity pro libovolnou skupinu osob (např. schůze učitelů, oběd)	Ano	Ano	Ano
Generovat rozvrh	Ano	Ano	Ano
Vlastnoručně předělávat vygenerovaný rozvrh	Ano	Ano	-
Přidat budovy školy a práci s nimi v rozvrhu, jako například délka přesunu	Ano	-	Ano
Otestovat upravený rozvrh a upozornit při konfliktu událostí	Ano	Ano	-
Kontrolovat uživatelsky zadané podmínky na rozvrh	Ano	-	-
Designově upravit výsledný rozvrh	Ano	Ano	-
Nastavit dozor v místnostech v době přestávek	Ano	-	-
Exportovat rozvrh do různých formátů a vytisknout	Ano	Ano	Ano
Importovat data z předchozích let	Ano	Ano	Ano
Přidávat vlastní tagy k rychlému filtrování obsahu v rozvrhu	-	Ano	-

Tabulka 1: Tabulka funkcí pro tvorbu rozvrhu. [10] [11] [14]

4 Vstupní soubory aplikace

Aplikace si načítá vstupy pomocí pěti textových souborů, nebo může uživatel vstupy vytvářet pomocí aplikace a kdykoliv je do těchto souborů ukládat. V následující části je popsána struktura jednotlivých souborů a jednotlivá vstupní data aplikace a jejich omezení.

4.1 Textový soubor „skola.txt“

Na každém řádku souboru „skola.txt“ je o rozvrhu uvedena jedna informace a to následovně:

- 1) název rozvrhu,
- 2) začátek školního roku ve formátu **rrrr**,
- 3) konec školního roku ve formátu **rrrr**,
- 4) délka malé přestávky v minutách,
- 5) délka velké přestávky v minutách,
- 6) čas začátku první vyučovací hodiny (počet hodin),
- 7) čas začátku první vyučovací hodiny (počet minut).

Textový soubor „skola.txt“ se vytváří pomocí vstupů zadaných do aplikace v záložce „Škola“.

```
Oblacna
2022
2023
10
20
7
55
```

Obrázek 1: Ukázka dat ze souboru „skola.txt“

Omezení při zadávání vstupů:

- Délka názvu rozvrhu je omezena na 20 znaků. Zároveň je v názvu zakázáno používat středník.
- Začátek první vyučovací hodiny je omezen na sedmou až desátou hodinu (viz [2.2 § 1 odst. 1 Vyhl. 48/2005 Sb.](#)).
- Délka malé přestávky v rozvrhu je omezena na 5 až 15 minut (viz [2.2 § 1 odst. 5 Vyhl. 48/2005 Sb.](#)).

Na každém řádku v souboru jsou uvedeny informace (odděleny jsou středníkem), definující právě jeden druh vyučovacího bloku. Pokud se jedná o vyučovací blok **Typ 1**, pak jsou informace uvedeny v následujícím pořadí:

- 1) Počet takto definovaných vyučovacích bloků v týdnu.
- 2) Zkratka třídy, pro kterou je vyučovací blok určen.
- 3) Číslo 0, které udává, že se jedná o vyučovací blok **Typ 1**.
- 4) Zkratka předmětu v tomto vyučovacím bloku.
- 5) Počet vyučujících **V**, kteří mohou tento předmět vyučovat.
- 6) Počet místností **M**, kde může být tento předmět vyučován.
- 7) **V** zkratkou vyučujících, kteří mohou daný předmět vyučovat (všechny oddělené středníkem).
- 8) **M** zkratkou místností, kde lze daný předmět vyučovat (všechny oddělené středníkem).

Pokud se jedná o vyučovací blok **Typ 2**, pak jsou informace uvedeny v následujícím pořadí:

- 1) Počet takto definovaných vyučovacích bloků v týdnu.
- 2) Zkratka třídy, pro kterou je vyučovací blok určen.
- 3) Číslo 1, které udává, že se jedná o vyučovací blok **Typ 2**.
- 4) Zkratky dvou předmětů v tomto vyučovacím bloku (oddělené jsou lomítkem).
- 5) Počet vyučujících **V₁**, kteří mohou vyučovat první předmět.
- 6) Počet místností **M₁**, kde může být první předmět vyučován.
- 7) **V₁** zkratkou vyučujících, kteří mohou první předmět vyučovat (všechny oddělené středníkem).
- 8) **M₁** zkratkou místností, kde lze první předmět vyučovat (všechny oddělené středníkem).
- 9) Počet vyučujících **V₂**, kteří mohou vyučovat druhý předmět.
- 10) Počet místností **M₂**, kde může být druhý předmět vyučován.
- 11) **V₂** zkratkou vyučujících, kteří mohou druhý předmět vyučovat (všechny oddělené středníkem).
- 12) **M₂** zkratkou místností, kde lze druhý předmět vyučovat (všechny oddělené středníkem).

Textový soubor „predmety.txt“ se vytváří pomocí vstupů zadaných do aplikace v záložce „Předměty“.

```
2;6.A;0;tv;2;1;Učitel7;Učitelka5B;Tělocvična
1;6.A;0;vz;2;6;Učitelka8;Učitelka9;6.A;6.B;7.A;8.A;9.A;9.B
1;6.A;1;aj/pč;1;4;Učitelka5B;6.A;UJ1;UJ2;8.A;1;6;Učitel13;6.A;6.B;7.A;8.A;9.A;9.B
1;6.A;1;aj/pč;1;4;Učitelka7B;6.A;UJ1;UJ2;8.A;1;6;Učitel13;6.A;6.B;7.A;8.A;9.A;9.B
1;6.A;1;inf/inf;2;2;Učitel13;Učitelka8;VTA;VTB;2;2;Učitel13;Učitelka8;VTA;VTB
```

Obrázek 5: Ukázka dat ze souboru „predmety.txt“

Omezení při zadávání vstupů:

- Délka zkratky předmětu je omezena na 6 znaků. Zároveň je ve zkratce zakázáno používat středník a lomítko.
- Každý vytvořený vyučovací blok typu 1 musí mít jinou zkratku předmětu.
- Každý vytvořený vyučovací blok typu 2 musí mít jinou kombinaci zkratek předmětů.
- Počet takto definovaných vyučovacích bloků v týdnu je omezen na 1 až 12.
- Každý vyučovací blok musí mít označenou alespoň jednu místnost, kde lze předmět vyučovat.
- Každý vyučovací blok musí mít označeného alespoň jednoho učitele, který může předmět vyučovat.
- Při vytváření vyučovacích bloků typu 2, musí uživatel vybrat takový počet možných místností a učitelů k vyučování předmětů, aby bylo reálně možné přidělit jiné finální učitele a místnosti k předmětům v tomto vyučovacím bloku.

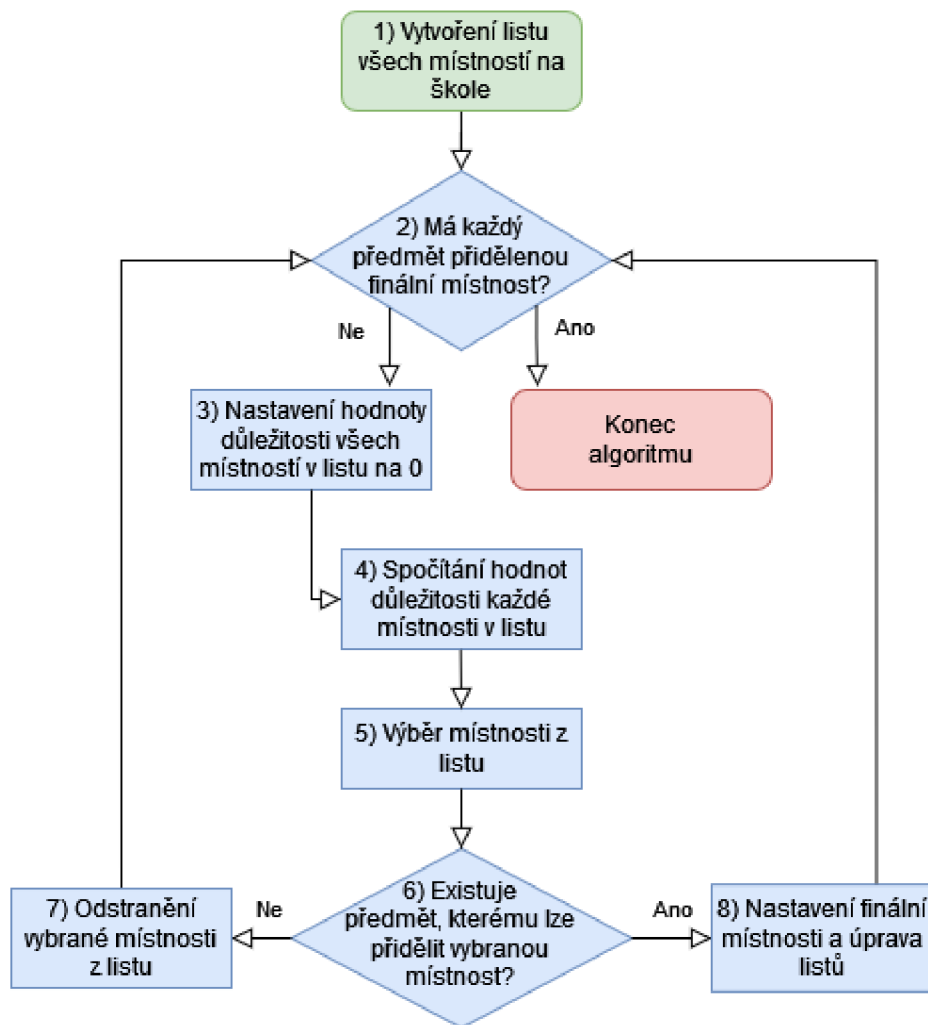
5 Algoritmus hledání rozvrhu

Jednotlivé části této kapitoly jsou věnovány popisu algoritmů, které jsou pro tuto práci zásadní, zabývají se konkrétně:

- přidělením vyučovacích místností k předmětům (viz 5.1),
- přiřazením učitelů k předmětům (viz 5.2),
- označením časů, ve které nemohou konkrétní vyučovací bloky probíhat (viz 5.3),
- vytvářením grafu a přiřazováním konkrétních časů k jednotlivým vyučovacím blokům (viz 5.4),
- vyhodnocováním nalezených řešení (viz 5.5).

5.1 Přidělování finálních místností k předmětům

Před hledáním funkčního řešení rozvrhu je potřeba každému předmětu konkrétní skupiny osob (předmět konkrétní skupiny osob = \mathbf{P}) přidělit právě jednu finální místnost k vyučování. Ta je vybrána z listu místností, který slouží jako výpis vhodných místností k vyučování předmětu \mathbf{P} (list místností vhodných k vyučování předmětu $\mathbf{P} = \mathbf{L}_{\mathbf{pm}}$). Pro přidělení finálních místností k předmětům \mathbf{P} , byl vytvořen vlastní algoritmus. Ten se snaží přidělit místnosti tak, aby každá místnost na škole byla používána pro vyučování stejně často. Vlastní algoritmus funguje následovně:



Obrázek 6: Vývojový diagram algoritmu pro přidělování finálních místností

- 1) Algoritmus vytvoří list všech místností na škole (list všech místností na škole = $L_{mís}$). Každé místnosti v listu $L_{mís}$ (místnost v listu $L_{mís}$ = $M_{mís}$) se v průběhu algoritmu přepočítává číselná hodnota, která reprezentuje důležitost místnosti $M_{mís}$ v rozvrhu (číselná hodnota reprezentující důležitost místnosti $M_{mís}$ v rozvrhu = $D_{mís}$).
- 2) Algoritmus zjistí, jestli má každý předmět P přidělenou finální místnost.
 - Pokud ano, algoritmus končí.
 - Pokud ne, algoritmus pokračuje.
- 3) Hodnota důležitosti $D_{mís}$ se u všech místností v listu $L_{mís}$ nastaví na 0.
- 4) Vybere se první místnost $M_{mís}$ z listu $L_{mís}$. Poté jsou postupně procházeny všechny předměty P na škole.

- Pokud se u konkrétního předmětu P vyskytuje v jeho listu místností L_{pm} místnost $M_{mís}$, potom se vždy k hodnotě důležitosti $D_{mís}$ místnosti $M_{mís}$ přičte hodnota, která se vypočítá následovně:

$$(\text{počet vyučovacích bloků předmětu } P) / (\text{počet místností předmětu } P \text{ v listu } L_{pm})$$

Takto algoritmus spočítá hodnotu důležitosti $D_{mís}$ každé místnosti $M_{mís}$ z listu $L_{mís}$.

- 5) Algoritmus vybere takovou místnost z listu $L_{mís}$, která má nejmenší hodnotu důležitosti $D_{mís}$ (místnost z listu $L_{mís}$, která má nejmenší hodnotu důležitosti $D_{mís} = M_n$).
- 6) Algoritmus prochází všechny předměty P , které nemají přiřazenou finální místnost a zároveň mají místnost M_n zapsanou v listu místností L_{pm} . Z nich algoritmus vybere takový předmět P , který má největší počet vyučovacích bloků (tento předmět je označen P_{nm}) a pokračuje krokem 8).
 - Pokud existuje více takových předmětů P , vybere jeden náhodně (tento předmět je označen P_{nm}) a pokračuje krokem 8).
 - Pokud takový předmět P neexistuje, pokračuje algoritmus krokem 7).
- 7) Algoritmus nedokáže přiřadit místnost M_n k žádnému předmětu P , a proto odstraní místnost M_n z listu $L_{mís}$. Algoritmus se vrací ke kroku 2).
- 8) Algoritmus u předmětu P_{nm} odstraní z listu místností L_{pm} všechny místnosti, kromě místnosti M_n . Místnost M_n se u předmětu P_{nm} nastaví jako finální místnost.
 - Pokud je předmět P_{nm} součástí vyučovacího bloku, který se dělí do více předmětů (viz 4.5 typ 2), potom u tohoto druhého předmětu P algoritmus odstraní místnost M_n z listu místností L_{pm} .

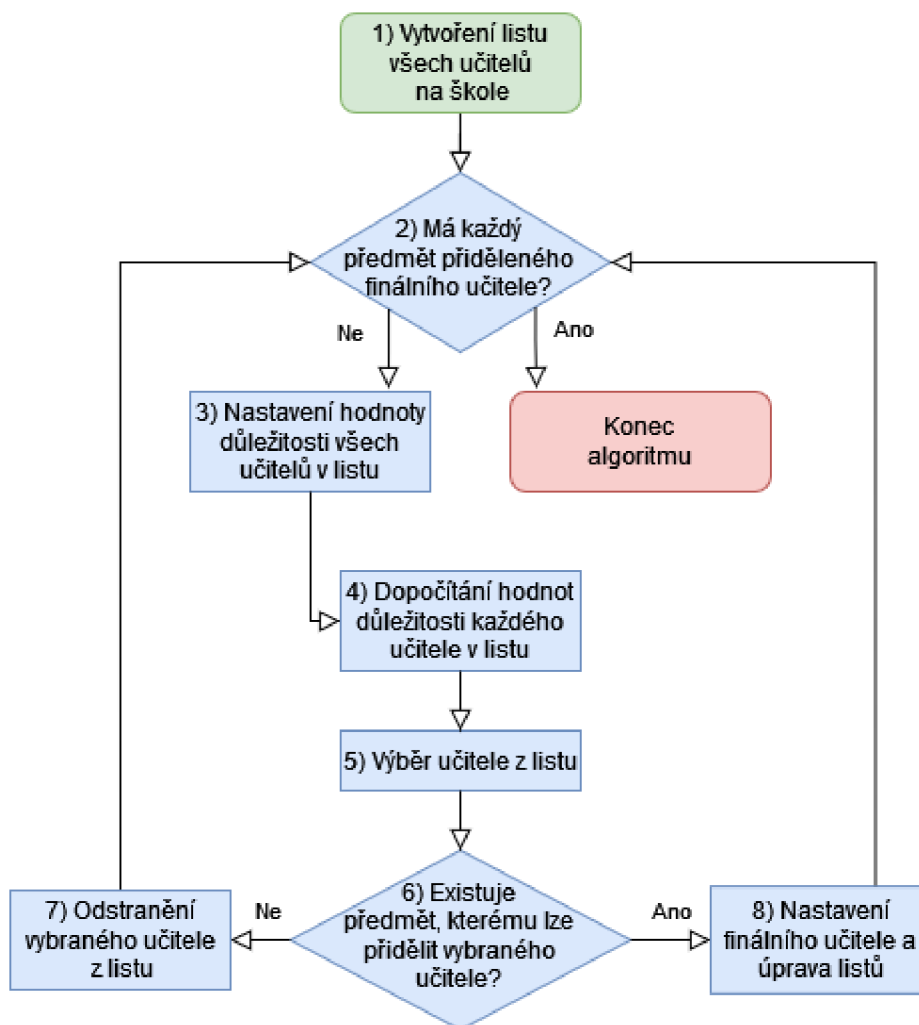
Algoritmus se vrací ke kroku 2).

5.2 Přidělování finálních učitelů k předmětům

Před hledáním funkčního řešení rozvrhu je potřeba každému předmětu konkrétní skupiny osob (předmět konkrétní skupiny osob = P) přidělit právě jednoho finálního učitele k vyučování. Algoritmus pro přidělování finálních učitelů k předmětům P funguje velmi podobně jako algoritmus pro přidělování finálních místností k předmětům P .

Finální učitel je vybrán z listu učitelů, který slouží jako výpis vhodných učitelů k vyučování předmětu P (list učitelů vhodných k vyučování předmětu $P = L_{pu}$). Pro přidělení finálních učitelů k předmětům P , byl vytvořen vlastní algoritmus. Ten se

snaží přidělit učitele tak, aby se každý učitel odchyloval stejně od počtu hodin, které má vyučovat v jednom týdnu (viz 4.4). Vlastní algoritmus funguje následovně:



Obrázek 7: Vývojový diagram algoritmu pro přidělování finálních učitelů

- 1) Algoritmus vytvoří list všech učitelů na škole (list všech učitelů na škole = $L_{u\check{c}}$). Každému učiteli v listu $L_{u\check{c}}$ (učitel v listu $L_{u\check{c}}$ = $U_{u\check{c}}$) se v průběhu algoritmu přepočítává číselná hodnota, která reprezentuje důležitost učitele $U_{u\check{c}}$ v rozvrhu (číselná hodnota reprezentující důležitost učitele $U_{u\check{c}}$ v rozvrhu = $D_{u\check{c}}$).
- 2) Algoritmus zjistí, jestli má každý předmět P přiděleného finálního učitele.
 - Pokud ano, algoritmus končí.
 - Pokud ne, algoritmus pokračuje.
- 3) Hodnota důležitosti $D_{u\check{c}}$ se u všech učitelů v listu $L_{u\check{c}}$ nastaví na 0 a od každé hodnoty $D_{u\check{c}}$, která patří učiteli $U_{u\check{c}}$, se odečte počet hodin, které má vyučovat učitel $U_{u\check{c}}$ v jednom týdnu.

- 4) Vybere se první učitel $U_{u\check{c}}$ z listu $L_{u\check{c}}$. Poté jsou postupně procházeny všechny předměty P na škole.
 - Pokud se u konkrétního předmětu P vyskytuje v jeho listu učitelů L_{pu} učitel $U_{u\check{c}}$, potom se vždy k hodnotě důležitosti $D_{u\check{c}}$ učitele $U_{u\check{c}}$ přičte hodnota, která se vypočítá následovně:

$$(\text{počet vyučovacích bloků předmětu } P) / (\text{počet učitelů předmětu } P \text{ v listu } L_{pu})$$
 Takto algoritmus spočítá hodnotu důležitosti $D_{u\check{c}}$ každého učitele $U_{u\check{c}}$ z listu $L_{u\check{c}}$.
- 5) Algoritmus vybere takového učitele z listu $L_{u\check{c}}$, který má nejmenší hodnotu důležitosti $D_{u\check{c}}$ (učitel z listu $L_{u\check{c}}$, který má nejmenší hodnotu důležitosti $D_{u\check{c}} = U_n$).
- 6) Algoritmus prochází všechny předměty P , které nemají přiřazeného finálního učitele a zároveň mají učitele U_n zapsaného v listu učitelů L_{pu} . Z nich algoritmus vybere takový předmět P , který má největší počet vyučovacích bloků (tento předmět je označen P_{nu}) a pokračuje krokem 8).
 - Pokud existuje více takových předmětů P , vybere jeden náhodně (tento předmět je označen P_{nu}) a pokračuje krokem 8).
 - Pokud takový předmět P neexistuje, pokračuje algoritmus krokem 7).
- 7) Algoritmus nedokáže přiřadit učitele U_n k žádnému předmětu P , a proto odstraní učitele U_n z listu $L_{u\check{c}}$. Algoritmus se vrací ke kroku 2).
- 8) Algoritmus u předmětu P_{nu} odstraní z listu učitelů L_{pu} všechny učitele, kromě učitele U_n . Učitel U_n se u předmětu P_{nu} nastaví jako finální učitel.
 - Pokud je předmět P_{nu} součástí vyučovacího bloku, který se dělí do více předmětů (viz 4.5 typ 2), potom u tohoto druhého předmětu P algoritmus odstraní učitele U_n z listu učitelů L_{pu} .

Algoritmus se vrací ke kroku 2).

5.3 Označování časů, ve kterých se předmět nemůže vyučovat

Každému vyučovacím bloku na škole (list všech vyučovacích bloků na škole = L) bude další algoritmus (viz 5.4) přiřazovat finální čas vyučování.

Předtím, než bude vyučovacím blokům v listu L (vyučovací blok v listu $L = B$) přiřazován finální čas, algoritmus označí časové bloky, které vyučovacím blokům B nelze přiřadit. Každému vyučovacím bloku B je možné přiřadit právě 50 různých časových bloků v týdnu. Proto je všem vyučovacím blokům B vytvořeno pole s padesáti hodnotami datového typu **bool** (pole vyučovacího bloku $B = P_b$), kdy každá hodnota **bool** patří jednomu konkrétnímu časovému bloku v týdnu.

- Hodnota **False** v poli P_b značí, že tento časový blok v týdnu lze přiřadit jako finální čas vyučovacího bloku **B**.
- Hodnota **True** v poli P_b značí, že tento časový blok v týdnu nelze přiřadit jako finální čas vyučovacího bloku **B**.

U všech vyučovacích bloků **B** je každá hodnota pole P_b ze začátku nastavena na **False**. Poté jsou postupně procházeny všechny vyučovací bloky **B** na škole.

- Pokud je ke konkrétnímu vyučovacímú bloku **B** přiřazen finální učitel, který má v některý z časových bloků v týdnu zakázáno vyučovat, potom se u hodnoty v poli P_b , která reprezentuje tento časový blok, nastaví hodnota na **True**.
- Pokud je ke konkrétnímu vyučovacímú bloku **B** přiřazena třída, která se má v některý z časových bloků v týdnu zakázáno učit, potom se u hodnoty v poli P_b , která reprezentuje tento časový blok, nastaví hodnota na **True**.

5.4 Přiřazování časů k vyučovacím blokům

K použití algoritmu pro přiřazování časů (časových bloků) k vyučovacím blokům je zapotřebí vytvořit graf, do kterého je možné si zaznamenávat data.

Vytváření grafu

K vytvoření rozvrhu se využije neorientovaný graf, kde vrcholy představují jednotlivé vyučovací bloky a hrany představují společné učitele, místnosti nebo žáky. Graf je v aplikaci reprezentován zjednodušenou maticí sousednosti (zjednodušená matice sousednosti = M_s). Za předpokladu, že je číslo N počet všech vstupních vyučovacích bloků v rozvrhu, potom v zjednodušené matici sousednosti M_s reprezentuje každý řádek jeden z těchto N vyučovacích bloků. Každý sloupec v matici M_s také reprezentuje právě jeden z těchto N vyučovacích bloků. Jedná se tedy o čtvercovou matici $N \times N$ vyučovacích bloků, kde pokud platí alespoň jedno z následujících pravidel, potom mezi sebou mají vyučovací bloky v daném řádku a v daném sloupci hranu (v aplikaci je zde zapsána v matici hodnota **True**).

- Vyučovací blok v daném sloupci a v daném řádku mají společného finálního učitele.
- Vyučovací blok v daném sloupci a v daném řádku mají společnou finální místnost.
- Vyučovací blok v daném sloupci a v daném řádku má společné žáky.

Pokud neplatí žádné z uvedených pravidel, potom mezi sebou tyto vyučovací bloky hranu nemají (v aplikaci je zde zapsána v matici hodnota **False**).

Řešící algoritmus

Přiřazování časových bloků k vyučovacím blokům je velmi podobné problému počítání chromatického čísla grafu, kdy chromatické číslo grafu vyjadřuje minimální počet barev (barvy reprezentují jednotlivé časové bloky ve školním rozvrhu), který je nutný k obarvení grafu. [2]

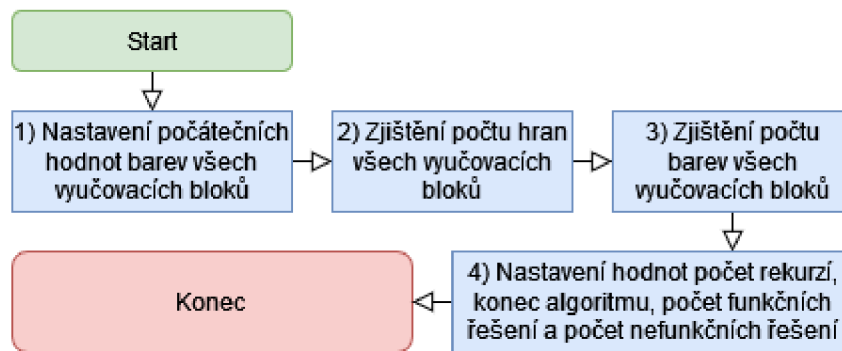
Zjištění chromatického čísla grafu je NP úplný problém a do této doby zatím neexistuje žádná efektivní cesta, kterou lze chromatické číslo jednoduše spočítat. Existuje však heuristický algoritmus, který se nazývá **greedy coloring algorithm**. Ten se snaží chromatické číslo grafu aproximovat. [2] [3] [17]

V aplikaci není hlavním úkolem upraveného greedy coloring algoritmu aproximovat chromatické číslo grafu zapsaného v matici \mathbf{M}_s , ale přiřazovat vyučovacím blokům barvy. Umožňuje tedy přiřadit barvy grafu zapsaného v matici \mathbf{M}_s vytvořeného ze vstupních dat od uživatele a zpracovaných algoritmy (viz 5.1, 5.2 a 5.3). V aplikaci je tento algoritmus upraven tak, aby vyučovacím blokům zkoušel přiřazovat pouze barvy, které jsou označeny předchozím algoritmem (viz 5.3).

K přiřazování barev předmětům v grafu existuje mnoho heuristik. V aplikaci je využita Brélazova heuristika **DSATUR**, která vybírá nejvíce nasycený vrchol. [16]

Příprava barvení vyučovacích bloků

Při hledání funkčního řešení rozvrhu je potřeba každému jednomu vyučovacím bloku konkrétní skupiny osob (jeden vyučovací blok konkrétní skupiny osob = **B**) přidělit právě jeden časový blok k vyučování (časový blok k vyučování = **C**). Na existujících 50 časových bloků **C** odkazují hodnoty celých čísel mezi 0 až 49, kterým se říká **barvy** (celé číslo mezi 0 až 49 odkazující na konkrétní časový blok **C** = **barva**). Vyučovací bloky **B** jsou uvedeny v listu, kde je přesně **N** vyučovacích bloků (list s **N** vyučovacími bloky **B** = **L**). Následující část algoritmu je příprava dat pro použití rekurzivního algoritmu, který se snaží každému vyučovacím bloku **B** v listu **L** přiřadit právě jednu **barvu**:



Obrázek 8: Vývojový diagram přípravy dat pro použití rekurzivního algoritmu

1) **Barva** každého bloku **B** v listu **L** se nastaví na -1. **Barva** -1 neodkazuje na žádný časový blok **C** v rozvrhu.

2) Algoritmus zjistí všem blokům **B** v listu **L** počet hran v matici **M_s** na ostatní bloky **B** v listu **L**. Nejprve se každému bloku **B** v listu **L** nastaví počet hran na 0 (počet hran konkrétního bloku **B** v listu **L** = **P_{hran}**).

Vybere se první blok **B** z listu **L**. Poté je postupně procházen řádek matice **M_s**, který odpovídá indexu vybraného bloku **B** v listu **L**.

- Pokud je v matici **M_s** na tomto řádku hodnota **True**, potom se vždy k hodnotě **P_{hran}** bloku **B** přičte číslo 1.

Takto algoritmus spočítá hodnotu **P_{hran}** každého bloku **B** z listu **L**.

3) Algoritmus zjistí všem blokům **B** v listu **L** počet **barev**, které nelze přiřadit bloku **B**. Nejprve se každému bloku **B** v listu **L** nastaví počet **barev**, které nelze přiřadit na 0 (počet **barev**, které nelze přiřadit konkrétnímu bloku **B** v listu **L** = **P_{bar}**).

Vybere se první blok **B** z listu **L**. Poté je postupně procházeno pole **P_b** (pole vyučovacího bloku **B**, viz 5.3), které patří vybranému bloku **B** z listu **L**.

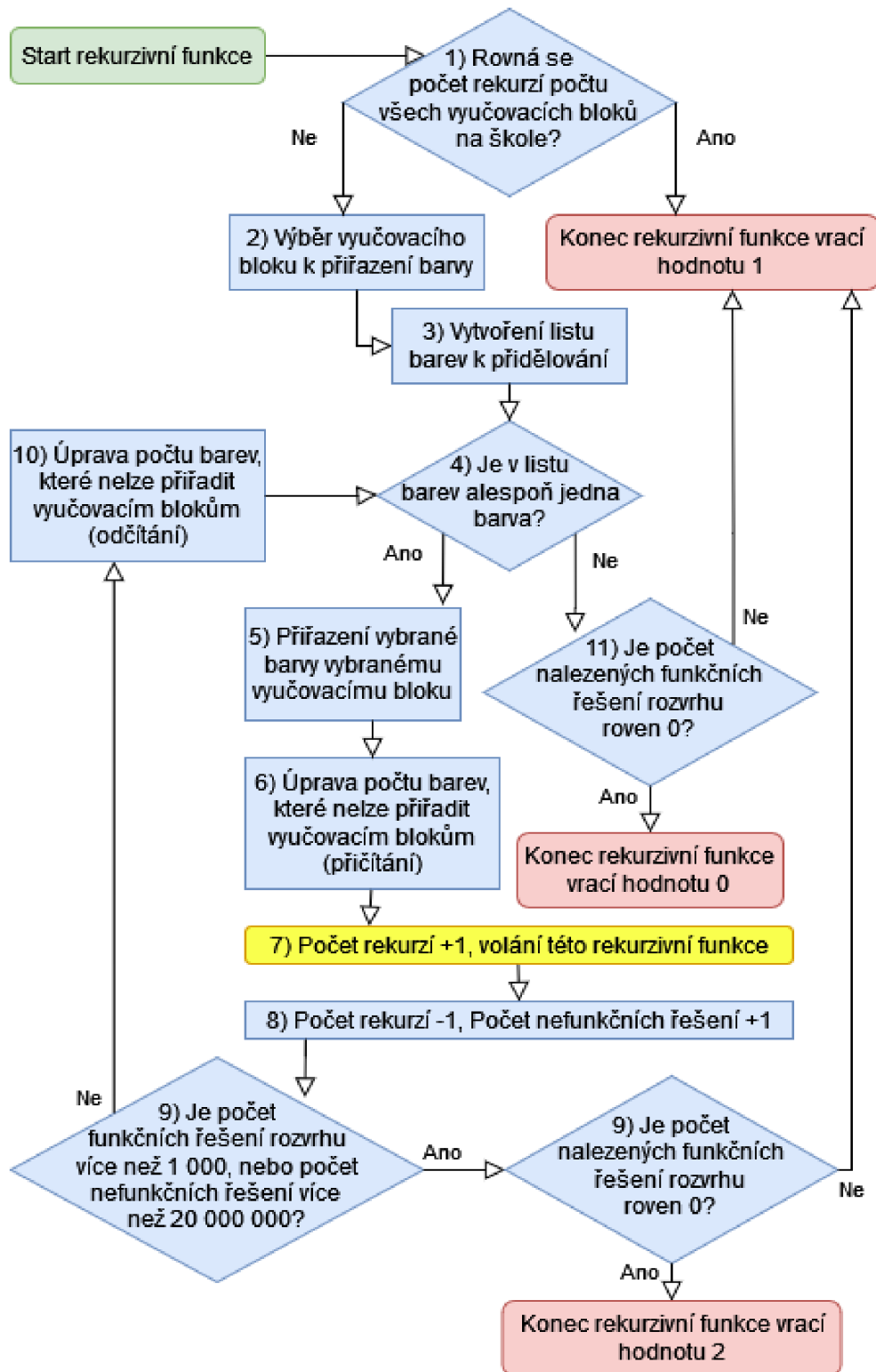
- Pokud je v poli **P_b** hodnota **True**, potom se vždy k hodnotě **P_{bar}** bloku **B** přičte číslo 1.

Takto algoritmus spočítá hodnotu **P_{bar}** každého bloku **B** z listu **L**.

4) Algoritmus nastaví počet rekurzí (počet rekurzí = **P_{rek}**) na 0, hodnotu konec algoritmu (**bool** hodnota konec algoritmu = **K_{alg}**) na **False**, počet nalezených funkčních řešení rozvrhu (počet nalezených funkčních řešení rozvrhu = **P_{fun}**) na 0 a počet nalezených nefunkčních řešení rozvrhu (počet nalezených nefunkčních řešení rozvrhu = **P_{nefun}**) na 0.

Barvení vyučovacích bloků

Následující část je rekurzivní funkce algoritmu, která se snaží každému vyučovacím bloku **B** v listu **L** přiřadit právě jednu **barvu**:



Obrázek 9: Vývojový diagram rekurzivní funkce pro přiřazování barev

- 1) Pokud se počet rekurzí P_{rek} rovná počtu N , potom:
 - Algoritmus vyhodnotí nalezené řešení rozvrhu (viz 5.5) a pokud je hodnocení výsledného rozvrhu větší, než hodnocení předchozího řešení, tak si algoritmus ukládá **barvy** přiřazené blokům B v listu L jako nejlepší nalezené řešení.
 - Algoritmus k počtu P_{fun} přičte 1.
 - Funkce vrací hodnotu 1.
- 2) Algoritmus vybírá z bloků B v listu L , které mají přidělenou **barvu** -1 (blok B v listu L , který má přidělenou **barvu** -1 = B_{bez}), takový blok, který má ze všech bloků B_{bez} z listu L největší hodnotu P_{bar} .
 - Pokud existuje více takových bloků B_{bez} z listu L , potom z nich vybírá ten, který má největší hodnotu P_{hran} .
 - Pokud existuje více takových bloků, vybere jeden náhodně. Tento vybraný blok rekurze číslo P_{rek} označí B_{vyb} (vybraný blok v rekurzi číslo P_{rek} = B_{vyb}).
- 3) Algoritmus vytvoří bloku B_{vyb} list **barev**, které se bloku B_{vyb} pokusí přiřadit (list **barev**, které se pokusí algoritmus bloku B_{vyb} v rekurzi P_{rek} přiřadit = L_{bar}).
Algoritmus prochází bloku B_{vyb} všechny hodnoty pole P_b .
 - Pokud je v poli P_b hodnota **False** a index (index pole P_b je číslo mezi 0 až 49), na kterém je tato hodnota **False** je **barva**, která je přidělena nějakému bloku B v listu L a pokud ani jeden blok B , který má společnou hranu s blokem B_{vyb} nemá přiřazenou **barvu**, která by se rovnala číslu tohoto indexu, potom číslo indexu přidám do listu L_{bar} .
 - Pokud je v poli P_b hodnota **False** a index (index pole P_b je číslo mezi 0 až 49), na kterém je tato hodnota **False** je **barva**, která není přidělena žádnému bloku B v listu L a pokud ani jeden blok B , který má společnou hranu s blokem B_{vyb} nemá přiřazenou **barvu**, která by se rovnala číslu tohoto indexu, potom číslo indexu přidám do listu L_{bar} pouze v případě, že se jedná o první nalezený index, který vyhovuje těmto podmínkám (tato podmínka zrychlí algoritmus).
- 4) Pokud je v listu L_{bar} nějaká **barva**, potom algoritmus vybere jednu, kterou se pokusí bloku B_{vyb} přiřadit (**barva** z listu L_{bar} , kterou algoritmus přiřadí bloku B_{vyb} = $barva_{fin}$). Nejprve se algoritmus snaží vybrat takovou **barvu**_{fin} z listu L_{bar} , která je přiřazena některému bloku B z listu L .

- Pokud taková **barva** v listu L_{bar} není, vybere algoritmus **barvu_{fin}**, která není přiřazena žádnému bloku B z listu L .
 - Pokud je v listu L_{bar} alespoň jedna **barva**, potom algoritmus pokračuje krokem 5).
 - Pokud není v listu L_{bar} žádná **barva**, potom algoritmus pokračuje krokem 11).
- 5) Algoritmus odstraní **barvu_{fin}** z listu L_{bar} a přiřadí **barvu_{fin}** bloku B_{vyb} .
- 6) Algoritmus prochází všechny bloky B z listu L , které mají společnou hranu s blokem B_{vyb} (vyučovací blok B z listu L , který má společnou hranu s blokem $B_{\text{vyb}} = B_{\text{spol}}$). Každému procházenému blok B_{spol} algoritmus přiřadí hodnotu **bool**, která určuje, jestli se u hodnoty P_{bar} bloku B_{spol} v rekurzi číslo P_{rek} přičetlo číslo 1 (hodnota **bool**, která určuje, jestli se u hodnoty P_{bar} bloku B_{spol} v rekurzi číslo P_{rek} přičetlo číslo 1 = $I_{\text{přič}}$).
- Pokud má procházený blok B_{spol} společnou hranu s alespoň jedním jiným blokem B , než je blok B_{vyb} , který má přiřazenou **barvu_{fin}**, potom algoritmus bloku B_{spol} nastaví hodnotu $I_{\text{přič}}$ na **False**.
 - Pokud nemá procházený blok B_{spol} společnou hranu s žádným jiným blokem B , než je blok B_{vyb} , který má přiřazenou **barvu_{fin}**, potom algoritmus bloku B_{spol} nastaví hodnotu $I_{\text{přič}}$ na **True**, k hodnotě P_{bar} bloku B_{spol} přičte číslo 1.
- 7) Algoritmus přičte k počtu rekurzí P_{rek} číslo 1 a volá tuto rekurzivní funkci algoritmu.
- 8) Algoritmus přičte k počtu P_{nefun} číslo 1 a odečte od počtu rekurzí P_{rek} číslo 1.
- 9) Pokud je počet řešení P_{nefun} větší, než 20 000 000 (hodnota nastavená autorem práce) nebo je počet řešení P_{fun} větší, než 1 000 (hodnota nastavená autorem práce), potom algoritmus nastaví hodnotu K_{alg} na **True**.
- Pokud je hodnota K_{alg} nastavena na **True**, a počet řešení P_{fun} je roven 0, potom funkce vrací hodnotu 2.
 - Pokud je hodnota K_{alg} nastavena na **True**, a počet řešení P_{fun} je více než 0, potom funkce vrací hodnotu 1.
 - Pokud je hodnota K_{alg} nastavena na **False**, potom algoritmus pokračuje krokem 10).
- 10) Algoritmus od všech bloků B_{spol} , které mají hodnotu $I_{\text{přič}}$ nastavenou na **True**:
- Odečte od jejich hodnoty P_{bar} číslo 1.

Algoritmus nastaví hodnotu **barvy** bloku **B_{vyb}** na -1 a vrací se ke kroku **4**).

11) Pokud je počet řešení **P_{fun}** roven 0, potom funkce vrací hodnotu 0. Pokud je počet řešení **P_{fun}** více než 0, potom funkce vrací hodnotu 1.

Význam návratových hodnot funkce:

- Pokud funkce v rekurzi **P_{rek}** číslo 0 vrací hodnotu 0, potom algoritmus vyzkoušel méně než 20 000 000 **P_{nefun}** řešení a funkční rozvrh pravděpodobně nelze vytvořit.
- Pokud funkce v rekurzi **P_{rek}** číslo 0 vrací hodnotu 1, potom algoritmus našel alespoň jedno, nebo více funkčních řešení rozvrhu.
- Pokud funkce v rekurzi **P_{rek}** číslo 0 vrací hodnotu 2, potom algoritmus vyzkoušel 20 000 000 **P_{nefun}** řešení, ale nenalezl žádné funkční řešení rozvrhu.

5.5 Vyhodnocování výsledného rozvrhu

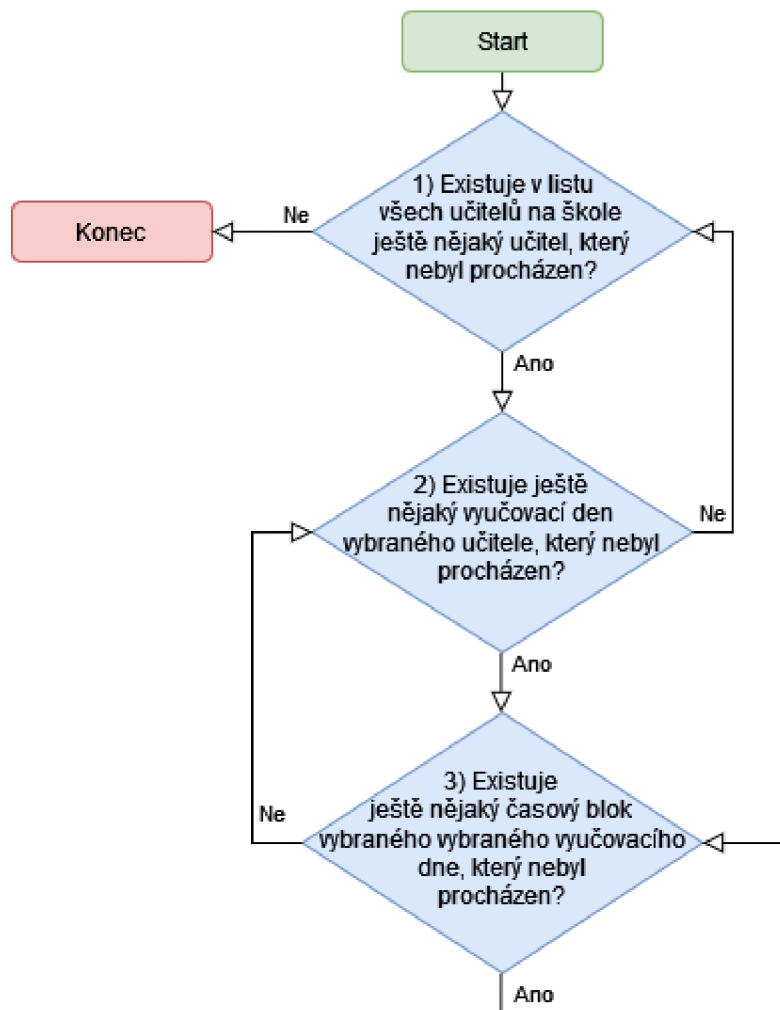
Z množiny všech nalezených funkčních řešení rozvrhu vybírá algoritmus to, které nejlépe ohodnotí. U nalezeného funkčního řešení hodnotí algoritmus následující 3 požadavky na rozvrh od učitelů a studentů (viz 2.3) a to:

- 1) Učitelé by neměli učit více než 3 hodiny po sobě.
- 2) Stejně předměty by měla mít každá třída pouze jednou denně.
- 3) Žáci by neměli mít příliš mnoho volných hodin mezi další výukou.

Na začátku je každý nově nalezený rozvrh ohodnocen 0. Pokud nalezený rozvrh porušuje některý z těchto tří požadavků, tak se jeho hodnocení zhorší.

1) Učitelé by neměli učit více než 3 hodiny po sobě

Část algoritmu, která hodnotí v rozvrhu požadavek „**učitelé by neměli učit více než 3 hodiny po sobě**“ zjišťuje, jestli konkrétní učitel nevyučuje více než 3 hodiny po sobě. Tato část algoritmu funguje následovně:

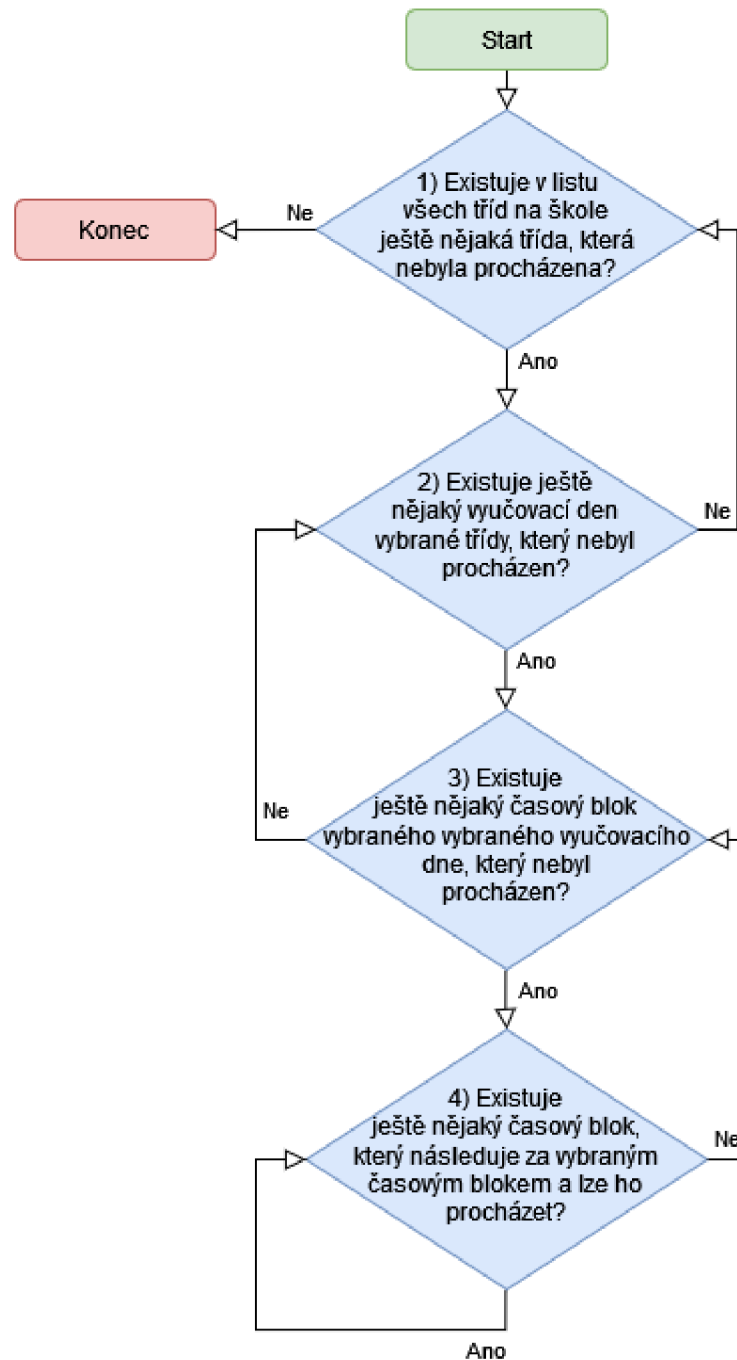


Obrázek 10: Vývojový diagram první části hodnotícího algoritmu

- 1) Algoritmus prochází list všech učitelů na škole (list všech učitelů na škole = $L_{u\check{c}}$). U každého učitele v listu $L_{u\check{c}}$ (učitel v listu $L_{u\check{c}} = U_{u\check{c}}$) prochází postupně všechny vyučovací dny učitele $U_{u\check{c}}$ v týdnu (vyučovací den učitele $U_{u\check{c}}$ v týdnu = V_u).
- 2) U procházeného vyučovacího dne V_u algoritmus nastaví hodnotu po sobě jdoucích vyučovaných bloků (hodnota po sobě jdoucích vyučovaných bloků = H_b) na 0. Algoritmus prochází postupně všechny časové bloky vyučovacího dne V_u (časový blok vyučovacího dne $V_u = B_{vu}$).
- 3) Pokud v procházený časový blok B_{vu} učitel $U_{u\check{c}}$ vyučuje nějaký předmět, potom algoritmus přičte k hodnotě H_b číslo 1. Pokud v procházený časový blok B_{vu} učitel $U_{u\check{c}}$ nevyučuje žádný předmět, potom algoritmus nastaví hodnotu H_b na 0. Pokud je hodnota H_b větší než 3, tak algoritmus od hodnocení rozvrhu odečte 1.

2) Stejné předměty by měla mít každá třída pouze jednou denně

Část algoritmu, která hodnotí v rozvrhu požadavek „stejné předměty by měla mít každá třída pouze jednou denně“ zjišťuje, jestli se v rozvrhu konkrétní třídy vyskytuje v jeden konkrétní den vícekrát stejný předmět. To vyhodnocuje pouze u vyučovacích bloků typu 1 (viz 4.5 typ 1), protože v algoritmu není způsob jak jednoznačně určit, které konkrétní skupině ve třídě patří předmět ve vyučovacím bloku typu 2 (viz 4.5 typ 2). Tato část algoritmu funguje následovně:

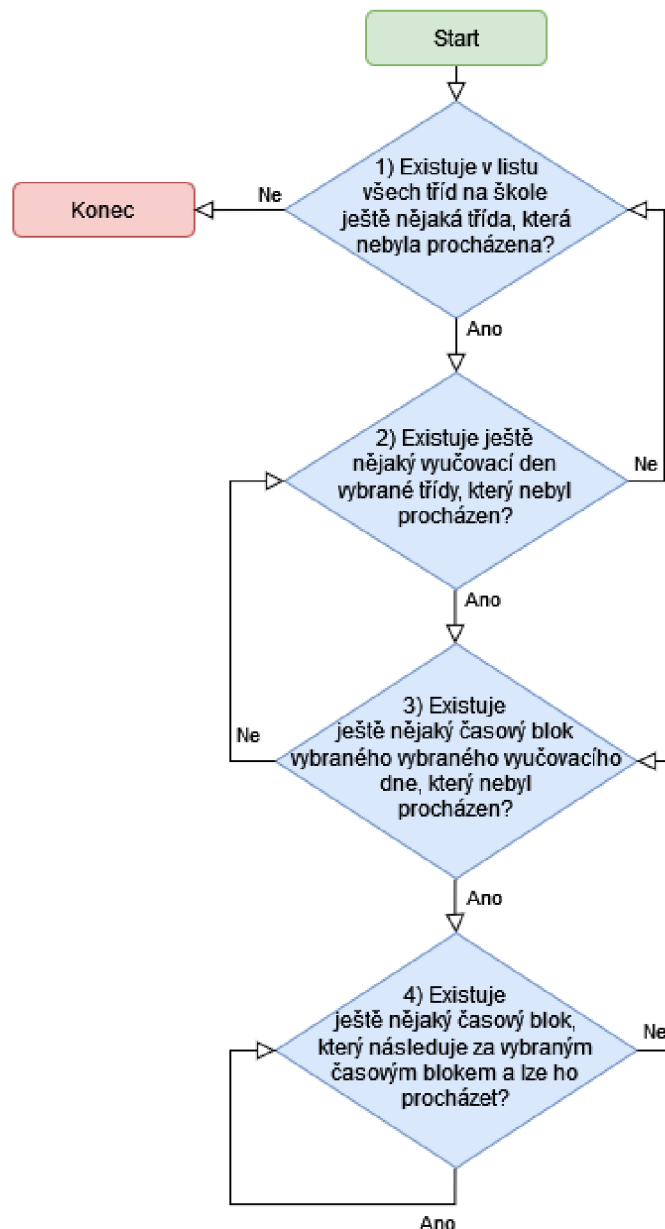


Obrázek 11: Vývojový diagram druhé části hodnotícího algoritmu

- 1) Algoritmus prochází list všech tříd na škole (list všech tříd na škole = L_{tr}). U každé třídy v listu L_{tr} (třída v listu $L_{tr} = T$) prochází postupně všechny vyučovací dny třídy T v týdnu (vyučovací den třídy T v týdnu = V_t).
- 2) U procházeného vyučovacího dne V_t algoritmus prochází postupně všechny jeho časové bloky (časový blok vyučovacího dne $V_t = B_{vt}$).
- 3) Pokud má v procházený časový blok B_{vt} třída T nějaký vyučovací blok typu 1, potom algoritmus prochází postupně všechny časové bloky, které následují za tímto blokem B_{vt} ve dni V_t (časový blok, který následuje za blokem B_{vt} ve dni $V_t = B_n$).
- 4) Pokud je v rozvrhu třídy T v procházený časový blok B_n stejná zkratka předmětu vyučovacího bloku typu 1 jako zkratka předmětu u časového bloku B_{vt} , potom algoritmus přestane procházet časové bloky B_n a od hodnocení rozvrhu odečte 5.

3) **Žáci by neměli mít příliš mnoho volných hodin mezi další výukou**

Část algoritmu, která hodnotí v rozvrhu požadavek „**žáci by neměli mít příliš mnoho volných hodin mezi další výukou**“ zjišťuje, jestli má konkrétní třída volné časové bloky mezi vyučováním a zároveň se podle zadaných vstupů (viz 4.3) mohla v tyto volné časové bloky výuky účastnit. Tato část algoritmu funguje následovně:



Obrázek 12: Vývojový diagram třetí části hodnotícího algoritmu

- 1) Algoritmus prochází list všech tříd na škole (list všech tříd na škole = $L_{tř}$). U každé třídy v listu $L_{tř}$ (třída v listu $L_{tř}$ = T) prochází postupně všechny vyučovací dny třídy T v týdnu (vyučovací den třídy T v týdnu = V_t).
- 2) U procházeného vyučovacího dne V_t algoritmus prochází postupně všechny jeho časové bloky (časový blok vyučovacího dne V_t = B_{vt}).
- 3) Pokud má v procházený časový blok B_{vt} třída T nějaké vyučování, potom algoritmus nastaví hodnotu pro odečtení od celkového hodnocení rozvrhu (hodnota pro odečtení od celkového hodnocení rozvrhu = H) na 0 a poté prochází postupně všechny časové bloky, které následují za tímto blokem B_{vt} ve dni V_t (časový blok, který následuje za blokem B_{vt} ve dni V_t = B_n).

- 4) Pokud je v rozvrhu třídy **T** v procházený časový blok **B_n** nějaké vyučování, potom algoritmus přestane procházet časové bloky **B_n** a od hodnocení rozvrhu odečte hodnotu **H**. Pokud není v rozvrhu třídy **T** v procházený časový blok **B_n** žádné vyučování a přitom se třída **T** podle zadaných vstupů v tyto volné časové bloky mohla výuky účastnit, potom algoritmus k hodnotě **H** přičítá 10.

6 Výstupní soubory aplikace

Výstupní data aplikace v sobě nesou informace o jednotlivých předmětech a to:

- zkratka předmětu,
- třídy, pro které je předmět určen,
- který den je předmět vyučován,
- ve kterém vyučovacím bloku je vyučován,
- kterým učitelem je vyučován,
- ve které místnosti výuka proběhne.

Tyto informace jsou přehledně zobrazeny ve výstupních souborech určených pro žáky, vyučující a místnosti. Výstupní informace se ukládají do souborů ve formátu:

- **XLSX** (určené pro uživatele, kteří preferují práci s kancelářskou aplikací Excel)
- **CSV** (určené pro uživatele, kteří nemají nainstalovanou kancelářskou aplikaci Excel nebo preferují práci s jinou aplikací pro zobrazení CSV souborů)

Tyto soubory se ukládají do složky automaticky pojmenované podle názvu rozvrhu, data a času. Složka s výstupními soubory se nachází ve složce aplikace „**Rozvrh\vystupni data**“. Uložené rozvrhy již může uživatel libovolně editovat.

6.1 Výstupní soubory pro žáky

Výstupní soubory pro žáky jsou takové soubory, ve kterých může žák školy jednoduše a rychle vyhledávat informace o svém vyučování.

V souboru **CSV** i **XLSX** je uveden rozvrh všech tříd na škole. Na prvním řádku rozvrhu konkrétní třídy je vždy uvedena třída, pro kterou je rozvrh určen, název rozvrhu a školní rok. Na ostatních řádcích je celotýdenní rozvrh výuky této třídy.

Výstup do CSV

Výstupní **CSV** soubor určený pro žáky se jmenuje „**Pro studenty.csv**“.

```
;;aj/Ucitelka6B;aj/Ucitelka6B;aj/Ucitelka6B;aj/Ucitelka6B;;;;  
;;4.A;4.A;4.A;4.A;;;;  
UT;;hv/Ucitel5;tv/Ucitelka4;cj/Ucitelka4;cj/Ucitelka4;cj/Ucitelka4;;;;  
;;4.A;Telocvicna;4.A;4.A;4.A;;;;
```

Obrázek 13: Výstup do CSV souboru z pohledu žáka

Výstup do XLSX

Výstupní XLSX soubor určený pro žáky se jmenuje „Pro studenty.xlsx“. Výstupní rozvrhy jsou graficky zvýrazněny.

4.A	Oblacna test	2022/2023				
	7:00 - 7:45	7:55 - 8:40	8:50 - 9:35	9:55 - 10:40	10:50 - 11:35	11:45 - 12:30
PO		aj/Ucitelka7B	aj/Ucitelka7B	aj/Ucitelka7B	aj/Ucitelka7B	cj/Ucitelka4
		UJ1	UJ1	UJ1	UJ1	4.A
		aj/Ucitelka6B	aj/Ucitelka6B	aj/Ucitelka6B	aj/Ucitelka6B	
		4.A	4.A	4.A	4.A	
UT		hv/Ucitel5	tv/Ucitelka4	cj/Ucitelka4	cj/Ucitelka4	cj/Ucitelka4
		4.A	Telocvicna	4.A	4.A	4.A
ST		inf/Ucitel7	tv/Ucitelka4	cj/Ucitelka4	cj/Ucitelka4	cj/Ucitelka4
		VTB	Telocvicna			4.A
		inf/Ucitelka6				
		VTA				
CT		pr/Ucitelka4	pc/Ucitelka4	ma/Ucitelka4	ma/Ucitelka4	ma/Ucitelka4
		4.A	4.A	4.A	4.A	4.A
PA		ma/Ucitelka4	vl/Ucitelka4	vl/Ucitelka4	vv/Ucitelka4	vv/Ucitelka4
		4.A	4.A	4.A	4.A	4.A

Obrázek 14: Výstup do XLSX souboru z pohledu žáka

6.2 Výstupní soubory pro učitele

Výstupní soubory pro učitele jsou takové soubory, ve kterých může učitel školy jednoduše a rychle vyhledávat informace o svém vyučování.

V souboru CSV i XLSX je uveden rozvrh všech učitelů na škole. Na prvním řádku rozvrhu konkrétního učitele je vždy uveden učitel, pro kterého je rozvrh určen, název rozvrhu a školní rok. Na ostatních řádcích je celotýdenní rozvrh výuky daného učitele.

Výstup do CSV

Výstupní CSV soubor určený pro učitele se jmenuje „Pro ucitele.csv“.

```
ST;;;inf/4.A;;tv/5.A;z/9.B;tvch/6.B;z/9.A;;z/9.A;  
;;VTB;;Telocvicna;7.A;Telocvicna;7.A;;7.A;  
CT;;;tv/5.A;tvch/6.A;tvch/6.B;tv/9.A;tvch/6.A;  
;;;Telocvicna;Telocvicna;Telocvicna;Telocvicna;Telocvicna;
```

Obrázek 15: Výstup do CSV souboru z pohledu učitele

Výstup do XLSX

Výstupní XLSX soubor určený pro učitele se jmenuje „Pro ucitele.xlsx“. Výstupní rozvrhy jsou graficky zvýrazněny.

Ucitel7	Oblacna test	2022/2023						
	7:00 - 7:45	7:55 - 8:40	8:50 - 9:35	9:55 - 10:40	10:50 - 11:35	11:45 - 12:30	12:40 - 13:25	
PO		z/7.A			z/6.B	tvch/8.A	tvch/8.A	
UT				z/6.A 7.A	6.B	Telocvicna	Telocvicna	
ST		inf/4.A VTB				h/6.B locvicna	z/9.A 7.A	
CT						h/6.B Telocvicna	tv/9.A Telocvicna	
PA		z/8.A 7.A		z/6.B 6.B	tv/9.A Telocvicna			

Obrázek 16: Výstup do XLSX souboru z pohledu učitele

6.3 Výstupní soubory pro místnosti

Výstupní soubory pro místnosti jsou takové soubory, ve kterých je možné jednoduše a rychle vyhledávat informace o vyučování v daných místnostech.

V souboru CSV i XLSX je uveden rozvrh všech místností na škole. Na prvním řádku rozvrhu konkrétní místnosti je vždy uvedena místnost, pro kterou je rozvrh určen, název rozvrhu a školní rok. Na ostatních řádcích je celotýdenní rozvrh výuky v dané místnosti.

Výstup do CSV

Výstupní CSV soubor určený pro místnosti se jmenuje „Pro mistnosti.csv“

```
CT;;nj/9.A;aj/7.A;nj/8.A;aj/8.A;aj/6.A;;;  
;;Ucitelka15;Ucitelka4B;Ucitelka10;Ucitelka5B;Ucitelka5B;;;  
PA;;aj/7.A;aj/7.A;nj/8.A;aj/8.A;aj/6.A;;;  
;;Ucitelka4B;Ucitelka4B;Ucitelka10;Ucitelka5B;Ucitelka5B;;;
```

Obrázek 17: Výstup do CSV souboru z pohledu místnosti

Výstup do XLSX

Výstupní XLSX soubor určený pro místnosti se jmenuje „Pro mistnosti.xlsx“. Výstupní rozvrhy jsou graficky zvýrazněny.

VTA	Oblacna test	2022/2023					
	7:00 - 7:45	7:55 - 8:40	8:50 - 9:35	9:55 - 10:40	10:50 - 11:35	11:45 - 12:30	12:40 - 13:25
PO			inf/5.A		inf/8.A		ecdl/9.A
			Ucitelka8		Ucitel13		Ucitel13
UT			/8.A			inf/9.B	inf/9.B
		Ucitel13	Ucitel13			Ucitelka8	Ucitelka8
ST		inf/4.A	inf/6.B			Ucitelka8	Ucitelka8
		Ucitelka6	Ucitelka6			Ucitelka6	Ucitelka6
CT		ecdl/7.A				Ucitelka6	Ucitelka6
		Ucitel13		Ucitel13		Ucitelka8	Ucitelka8
PA		inf/5.A	inf/9.A	inf/8.A		inf/9.A	inf/9.A
		Ucitelka8	Ucitelka6	Ucitel13		Ucitelka6	Ucitelka6

Obrázek 18: Výstup do XLSX souboru z pohledu místnosti

7 Aplikace Rozvrh

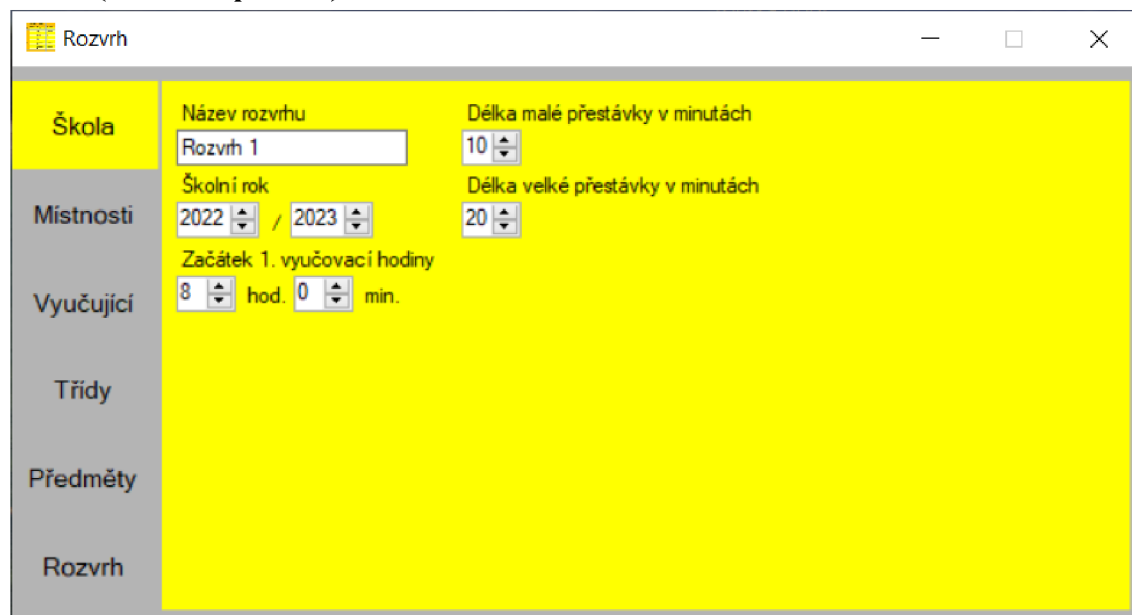
Rozvrh je aplikace Windows Forms a byla vytvořena ve vývojovém prostředí Visual Studio 2019. Byla naprogramována v jazyce C# a slouží k automatickému generování rozvrhů pro základní školy. V následujících bodech je popsáno, jak lze aplikaci spustit a práce s aplikací.

7.1 Spuštění aplikace

Ve složce aplikace Rozvrh se nachází spustitelný soubor „**setup.exe**“, který slouží k instalaci a spuštění aplikace. Aplikaci lze spustit na 64 bitovém operačním systému Windows 10.

7.2 Popis aplikace

Škola (záložka aplikace)



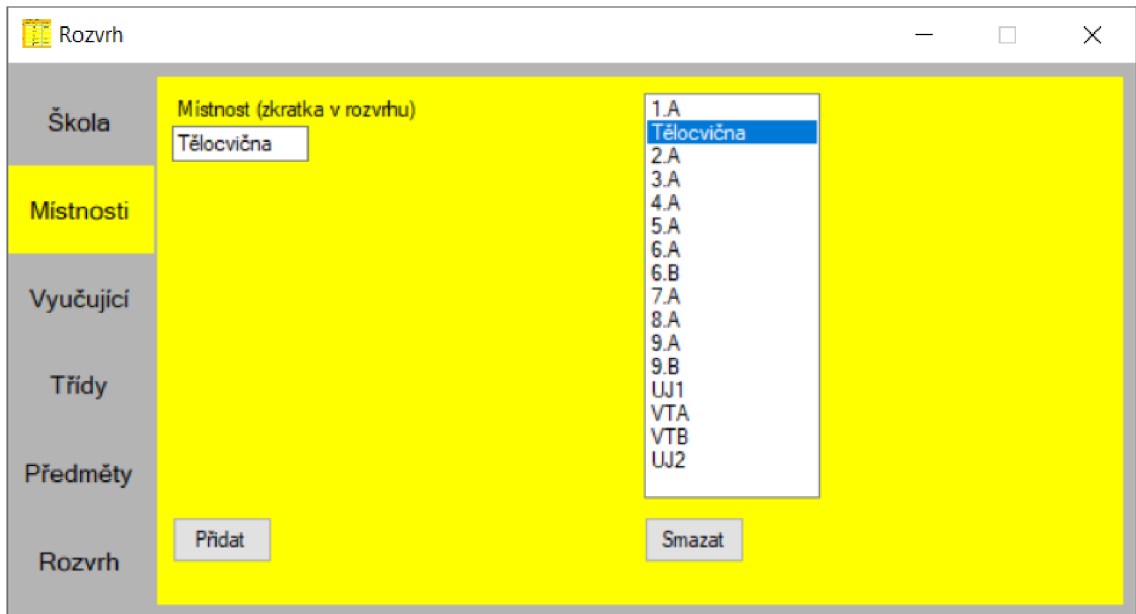
Obrázek 19: Záložka aplikace „Škola“

V záložce škola lze nastavit obecné informace a to:

- název rozvrhu (délka názvu rozvrhu je omezena na 20 znaků a je zakázáno používat středník),
- školní rok,
- čas začátku první vyučovací hodiny (začátek první vyučovací hodiny je omezen na sedmou až desátou hodinu viz 2.2 § 1 odst. 1 Vyhl. 48/2005 Sb.),
- délku malé přestávky (délka je omezena na 5 až 15 minut viz 2.2 § 1 odst. 5 Vyhl. 48/2005 Sb.),

- délku velké přestávky (délka je omezena na 15 až 30 minut viz 2.2 § 1 odst. 5 Vyhl. 48/2005 Sb.).

Místnosti (záložka aplikace)



Obrázek 20: Záložka aplikace „Místnosti“

V záložce místnosti lze přidávat a mazat jednotlivé místnosti na škole.

- Délka názvu místnosti je omezena na 10 znaků.
- V názvu je zakázáno používat středník.
- Každá místnost v rozvrhu musí mít jiný název.
- Místnost nelze odstranit, pokud je přiřazena některému z vyučovacích bloků.

Vyučující (záložka aplikace)

Škola Vyučující (zkratka v rozvrhu)
Učitelka6B

Místnosti Počet vyučovaných hodin v týdnu
15

Vyučující Výber, kdy tento učitel nemůže vyučovat
(doporučuje se nic nezaškrtnout)

	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.
Po	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Út	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
St	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Čt	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Pá	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Třídy

Předměty

Rozvrh

Přidat / Přepsat Smazat

Učitelka3
Učitelka4
Učitel5
Učitel7
Učitelka9
Učitelka10
Učitelka11
Učitelka8
Učitelka6
Učitel13
Učitelka7B
Učitelka5B
Učitelka15
Učitelka4B
Učitelka14
Učitelka12
Učitelka6B

Obrázek 21: Záložka aplikace „Vyučující“

V záložce vyučující lze přidávat, upravovat a mazat jednotlivé učitele na škole. U každého učitele lze nastavit:

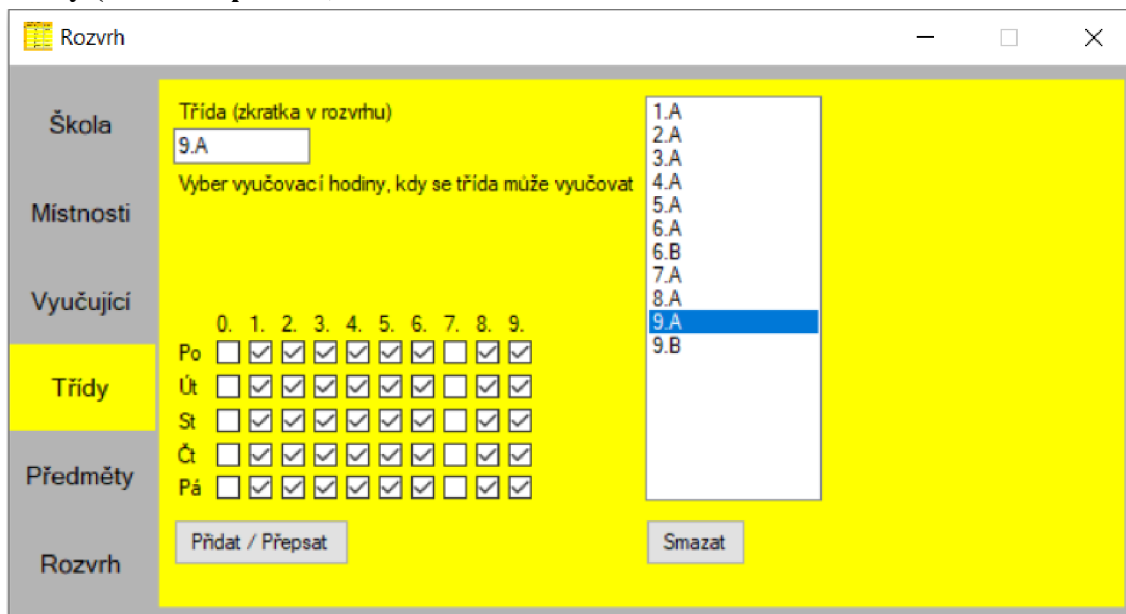
- zkratku učitele v rozvrhu,
- počet vyučovacích hodin v týdnu,
- časové bloky, kdy nemůže daný učitel vyučovat.

Omezení při zadávání vstupů:

- Délka pojmenování učitele je omezena na 11 znaků. Zároveň je v pojmenování zakázáno používat středník.
- Každý učitel v rozvrhu musí mít jiné pojmenování.
- Počet vyučovacích hodin v týdnu je omezen na 1 až 50.

Učitel nelze odstranit, pokud je přiřazen některému z vyučovacích bloků, lze ho pouze upravit. Vytvořeného učitele lze načíst kliknutím na jeho zkratku v seznamu.

Třídy (záložka aplikace)



Obrázek 22: Záložka aplikace „Třídy“

V záložce třídy lze přidávat, upravovat a mazat jednotlivé třídy na škole. U každé třídy lze nastavit:

- zkratku třídy v rozvrhu,
- časové bloky, kdy se daná třída může vyučovat.

Omezení při zadávání vstupů:

- Délka pojmenování třídy je omezena na 10 znaků. Zároveň je v pojmenování zakázáno používat středník.
- Každá třída v rozvrhu musí mít jiné pojmenování.

Třída nelze odstranit, pokud je přiřazena některému z vyučovacích bloků, lze ji pouze upravit. Vytvořenou třídu lze načíst kliknutím na její zkratku v seznamu.

Předměty (záložka aplikace)

Obrázek 23: Záložka aplikace „Předměty“

V záložce předměty lze přidávat, upravovat a mazat jednotlivé vyučovací bloky. U každého vyučovacího bloku lze nastavit:

- třídu, které daný vyučovací blok patří,
- počet takových vyučovacích bloků,
- zkratku předmětu nebo předmětů, které se v daný vyučovací blok vyučují,
- učitele, kteří mohou daný předmět vyučovat,
- místnosti, kde lze daný předmět vyučovat.

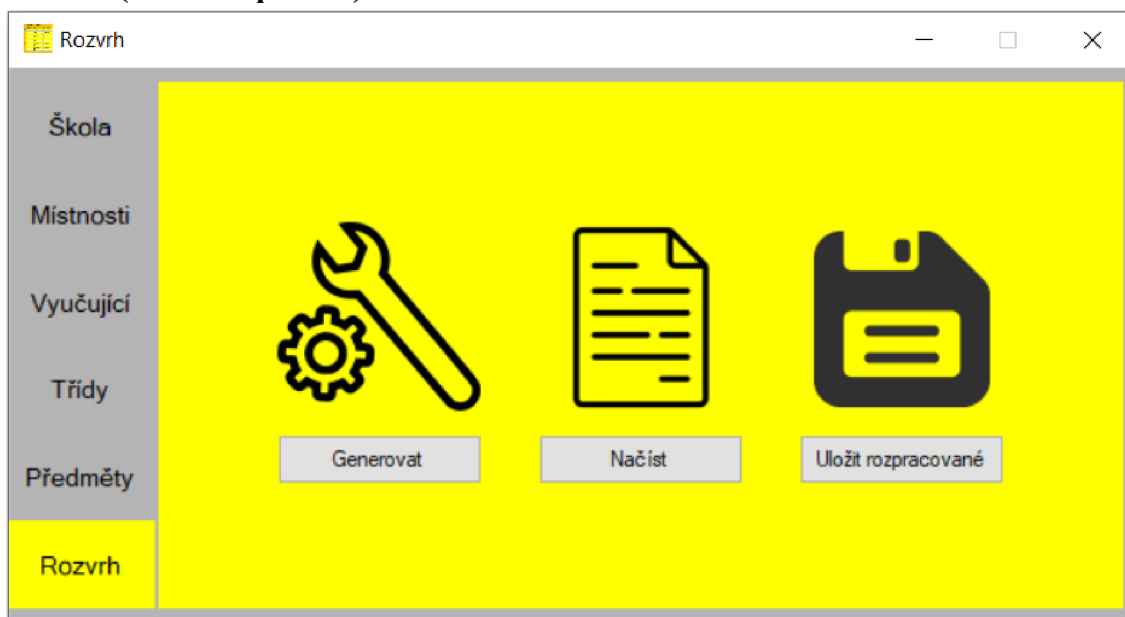
Každý vyučovací blok lze konkrétní třídu rozdělit na dvě skupiny, které se vyučují odděleně. Omezení při zadávání vstupů jsou následující:

- Délka zkratky předmětu je omezena na 6 znaků. Zároveň je ve zkratce zakázáno používat středník a lomítko.
- Každý vytvořený vyučovací blok typu 1 musí mít jinou zkratku předmětu.
- Každý vytvořený vyučovací blok typu 2 musí mít jinou kombinaci zkratek předmětů.
- Počet takto definovaných vyučovacích bloků v týdnu je omezen na 1 až 12.
- Každý vyučovací blok musí mít označenou alespoň jednu místnost, kde lze předmět vyučovat.
- Každý vyučovací blok musí mít označeného alespoň jednoho učitele, který může předmět vyučovat.

- Při vytváření vyučovacích bloků typu 2, musí uživatel vybrat takový počet možných místností a učitelů k vyučování předmětů, aby bylo reálně možné přidělit jiné finální učitele a místnosti k předmětům v tomto vyučovacím bloku.

Každému předmětu lze označit libovolný počet místností, kde lze předmět vyučovat a libovolný počet učitelů, kteří mohou předmět vyučovat.

Rozvrh (záložka aplikace)



Obrázek 24: Záložka aplikace „Rozvrh“

V záložce rozvrh lze generovat řešení rozvrhu, načítat vstupní data aplikace a ukládat vstupní data aplikace.

- Generovat rozvrh lze v aplikaci pomocí tlačítka „**Generovat**“ a to za podmínky, že byl vytvořen alespoň jeden vyučovací blok. Funkční (vygenerované) řešení rozvrhu se ukládá do dokumentů do složky „**Rozvrh soubory\ vystupni data**“.
- Načítat vstupní data aplikace lze pomocí tlačítka „**Načíst**“. Následně musí být vybrána data k načtení, která jsou uložena v souborech s názvy „místnosti.txt“, „predmety.txt“, „skola.txt“, „tridy.txt“ a „ucitele.txt“. Tyto soubory se při načítání musí nacházet ve stejné složce.
- Uložit vstupní data aplikace lze pomocí tlačítka „**Uložit rozpracované**“. Vstupní data aplikace se ukládají do dokumentů do složky „**Rozvrh soubory\ vstupni data**“.

8 Testování aplikace

K otestování aplikace na datech z reálné základní školy, byla vybrána škola ZŠ Oblačná (viz 8.1). Pro zjištění dat, z této základní školy, byla využita konzultace s vedením školy, konkrétně s ředitelkou školy Ing. Alenou Routovou a zástupkyní ředitelky Mgr. Eliškou Handšuhovou.

Z důvodu, že je ZŠ Oblačná malá škola, tak byla autorem práce uměle vytvořena i druhá větší základní škola, která se snaží napodobit požadavky reálné základní školy, ale má objemnější data pro testování aplikace (viz 8.2).

8.1 ZŠ Oblačná

Požadavky školy pro vytváření rozvrhu

2. patro	Učebna výpočetní techniky B VT - B		
	Učebna jazyků UJ	Učebna výpočetní techniky A VT - A	
1. patro	Učebna chemie 9. B	Učebna přírodopisu 7. A	Učebna hudební výchovy 6. A
		Učebna dějepisu 6. B	
přízemí	4. A	2. A	3. A Družina
	Družina	5. A Klub volného času	
Suterén 1	Družina		
	Učebna českého jazyka 9. A	1. A	
Suterén 2	Učebna jazyků (malá) UJ	Učebna jazyků (velká) 8. A	

Obrázek 25: Plán budovy ZŠ Oblačná. Převzato z [5]

- První třída je při výuce vždy v **Suterénu 1**, protože žáci první třídy mají snadný přístup do místnosti rovnou z šatny. Zároveň je tato místnost vybavena nejmenším vybavením (židle, lavice).
- Kmenové místnosti prvních stupňů, kromě první třídy se nachází v **přízemí**, pokud je to možno a to kvůli střední velikosti vybavení (židle, lavice).
- Výuka prvních stupňů probíhá pouze v kmenové místnosti, pokud se nejedná o předmět, který není možné v této místnosti vyučovat. Výjimku tvoří pouze hodiny tělesné výchovy, cizích jazyků a informatiky.
- Při výuce tělesné výchovy nejsou třídy prvního stupně rozděleny do skupin.
- Při výuce tělesné výchovy jsou třídy druhého stupně rozděleny na skupinu chlapců a skupinu dívek, kdy jedna ze skupin má výuku v tělocvičně a druhá skupina má ve stejnou dobu jiné nebo žádné vyučování.
- Škola má několik specializovaných místností (místnost chemie, přírodopisu, hudební výchovy, dějepisu, českého jazyka, místnosti výpočetní techniky, jazyků a tělocvična), které slouží především k vyučování určených předmětů a využívají se převážně pro druhý stupeň.
- U předmětu tělesná výchova a informatika je nutné, aby probíhaly vždy v jim určených místnostech kvůli nepřenositelnému vybavení.

Data školy (viz [příloha](#))

- Počet učitelů na škole: 19
 - Jména učitelů byla při testování nahrazena anonymním označením.
 - Každému učiteli byl v aplikaci nastaven počet, který udává, kolik by měl vyučovat hodin v týdnu.
 - Každému učiteli byly vybrány vyučovací bloky, které by mohl vyučovat.
- Počet místností na škole: 16
 - Každé místnosti byly vybrány vyučovací bloky, ve kterých by se daný blok mohl vyučovat.
- Počet tříd na škole: 11
 - Každé třídě byly označeny časové bloky, ve které se mohou vyučovat.
 - Každé třídě byly vytvořeny vyučovací bloky, které odpovídají jejich učebnímu plánu.
- Počet vyučovacích bloků na škole: 312 (vyučovacích hodin 387)

Výsledek přidělení finálních místností

Dle počtu vyučovacích hodin na celé škole se v průměru v jedné učebně učí 24,1875 hodin za týden. Odchylka představuje rozdíl mezi průměrným počtem vyučovaných hodin a skutečností.

Název místnosti	Počet vyučovacích hodin v místnosti	Odchylka od průměrného počtu vyučování v místnosti (24,1875)
1. A	20	-4,1875
2. A	20	-4,1875
3. A	23	-1,1875
4. A	22	-2,1875
5. A	22	-2,1875
6. A	27	2,8125
6. B	27	2,8125
7. A	27	2,8125
8. A	27	2,8125
9. A	27	2,8125
9. B	27	2,8125
Tělocvična	30	5,8125
UJ1	27	2,8125
UJ2	27	2,8125
VTA	17	-7,1875
VTB	17	-7,1875

Tabulka 2: Tabulka výsledků přidělení místností (ZŠ Oblačná)

Výsledek přidělení finálních učitelů

Označení učitele	Kolik by měl vyučovat hodin	Počet přidělených vyučovacích hodin	Odchylka od počtu hodin, kolik by měl vyučovat
Učitelka 1	22	22	0
Učitelka 2	22	22	0
Učitelka 3	22	22	0
Učitelka 4	22	22	0
Učitel 5	22	22	0
Učitelka 6	22	22	0
Učitel 7	22	22	0
Učitelka 8	22	23	1
Učitelka 9	22	22	0
Učitelka 10	22	22	0
Učitelka 11	22	23	1
Učitelka 12	12	12	0
Učitel 13	22	22	0
Učitelka 14	4	4	0
Učitelka 15	22	21	-1
Učitelka 4B	22	23	1
Učitelka 5B	22	23	1
Učitelka 6B	15	15	0
Učitelka 7B	22	23	1

Tabulka 3: Tabulka výsledků přidělení učitelů (ZŠ Oblačná)

Výsledek generování rozvrhu

Bylo nalezeno 1 000 funkčních řešení rozvrhu. Z nich bylo vybráno nejlépe ohodnocené řešení (způsob hodnocení viz 5.5). U vybraného řešení bylo otestováno, že dodržuje nutná omezení pro tvorbu rozvrhu (viz 2.1).

8.2 Škola vytvořená autorem práce

Data školy (viz příloha)

- Počet učitelů na škole: 27
- Počet místností na škole: 24
- Počet tříd na škole: 17
- Počet vyučovacích bloků na škole: 462 (vyučovacích hodin 564)

Výsledek přidělení finálních místností

Dle počtu vyučovacích hodin na celé škole se v průměru v jedné učebně učí 23,5 hodin za týden. Odchylka představuje rozdíl mezi průměrným počtem vyučovaných hodin a skutečností.

Název místnosti	Počet vyučovacích hodin v místnosti	Odchylka od průměrného počtu vyučování v místnosti (23,5)
1. A	20	-3,5
1. B	20	-3,5
2. A	20	-3,5
2. B	20	-3,5
3. A	23	-0,5
3. B	23	-0,5
4. A	22	-1,5
4. B	22	-1,5
5. A	22	-1,5
5. B	22	-1,5
6. A	28	4,5
6. B	28	4,5
7. A	28	4,5
7. B	24	0,5
8. A	27	3,5
9. A	28	4,5
9. B	28	4,5
TV1	22	-1,5
TV2	22	-1,5
UJ1	28	4,5
UJ2	28	4,5
UJ3	14	-9,5
VTA	23	-0,5
VTB	22	-1,5

Tabulka 4: Tabulka výsledků přidělení místností (škola vytvořená autorem práce)

Výsledek přidělení finálních učitelů

Označení učitele	Kolik by měl vyučovat hodin	Počet přidělených vyučovacích hodin	Odchylka od počtu hodin, kolik by měl vyučovat
U01	22	22	0
U02	15	15	0
U03	22	22	0
U04	22	22	0
U05	5	5	0
U06	12	12	0
U07	22	22	0
U08	22	22	0
U09	22	22	0
U10	22	22	0
U11	22	22	0
U12	22	23	1
U13	22	22	0
U14	22	23	1
U15	22	22	0
U16	22	23	1
U17	22	22	0
U18	22	23	1
U19	22	22	0
U20	22	22	0
U21	22	22	0
U22	22	22	0
U23	22	22	0
U24	22	22	0
U25	22	22	0
U26	22	21	-1
U27	22	23	1

Tabulka 5: Tabulka výsledků přidělení učitelů (škola vytvořená autorem práce)

Výsledek generování rozvrhu

Bylo nalezeno 1 000 funkčních řešení rozvrhu. Z nich bylo vybráno nejlépe ohodnocené řešení (způsob hodnocení viz [5.5](#)). U vybraného řešení bylo otestováno, že dodržuje nutná omezení pro tvorbu rozvrhu (viz [2.1](#)).

9 Závěr

V bakalářské práci byly zjištěny obecné faktory ovlivňující tvorbu rozvrhů. Mezi tyto faktory byly řazeny nutné omezení pro vytvoření funkčního řešení rozvrhu, zákony, které omezují tvorbu rozvrhů škol a vlastním průzkumem byly zjištěny osobní požadavky vyučujících a studentů na dobrý rozvrh.

Pro rešeršní účely byly vybrány tři aplikace, určené k automatickému vytváření rozvrhu. Ty byly následně porovnány v oblastech funkce pro tvorbu rozvrhu, využívaný algoritmus, jejich omezení a cena.

Pro splnění hlavního cíle bakalářské práce byla naprogramována aplikace Rozvrh, v programovacím jazyce C#. Aplikací lze vytvářet, načítat a ukládat vstupní soubory potřebné pro vytváření rozvrhu. Formátování vstupních souborů a omezení vstupních dat bylo v práci popsáno. Následně byly popsány algoritmy aplikované v aplikaci (algoritmus pro přidělování finálních místností k předmětům, algoritmus pro přidělování finálních učitelů k předmětům, algoritmus pro přiřazování časů k vyučovacím blokům, algoritmus pro vyhodnocování výsledného rozvrhu) a výstupní soubory v datovém typu CSV a XLSX určené pro zobrazení výsledného rozvrhu z pohledu žáka, učitele a místnosti. V práci je poté popsána aplikace Rozvrh a její spuštění.

K otestování aplikace na datech z reálné základní školy, byla vybrána škola ZŠ Oblačná. Od té byly pomocí konzultací zjištěny požadavky školy pro vytváření rozvrhu a data pro testování. Z důvodu, že je ZŠ Oblačná malá škola, tak byla autorem práce uměle vytvořena i druhá větší základní škola, která se snaží napodobit požadavky reálné základní školy, ale má objemnější data pro testování aplikace.

Naprogramovaná aplikace dokáže pomocí algoritmů vygenerovat a vybrat nejlépe ohodnocené funkční řešení rozvrhu pro školu ZŠ Oblačná i pro uměle vytvořenou základní školu. Pokud aplikace nenalezne řešení pro vstupní data, je vhodné přidat jednotlivým třídám na škole počet vyučovacích bloků, ve kterém se může daná třída vyučovat.

Seznam použité literatury

- [1] ČESKO. Vyhláška č. 48/2005 Sb. ze dne 25. ledna 2005 o základním vzdělávání a některých náležitostech plnění povinné školní docházky. In: Sbírnka zákonů České republiky. 2005, částka 11, Dostupný také z: <https://www.zakonyprolidi.cz/cs/2005-48>
- [2] Dong Fengming, Eng Guan Tay a Khee Meng Koh. Introduction to graph theory: h3 mathematics [online]. Singapur: World Scientific Publishing Co. Pte., 2007 [cit. 2022-03-04]. ISBN 10 981-270-525-2. Dostupné z: https://books.google.cz/books?id=7_bQa4SJTQQC
- [3] ERICKSON, Jeff. Algorithms. 2019. ISBN 978-1-792-64483-2.

Elektronické zdroje:

- [4] Timetabling: A State-of-the-Art Evolutionary Approach. International Journal of Machine Learning and Computing [online]. 2013, 3(3), 255 - 258 [cit. 2021-10-17]. Dostupné z: <http://www.ijmlc.org/index.php?m=content&c=index&a=show&catid=37&id=326>
- [5] HANDŠUHOVÁ, Eliška. Plánek [elektronická pošta]. Message to: m.kracm@email.cz. 26. listopadu 2020 12:43 [cit. 2020-11-26].
- [6] Prime Timetable [online]. 2021 [cit. 2022-01-03]. Dostupné z: <https://primetimetable.com/>
- [7] Asc Timetables [online]. 2021 [cit. 2022-01-04]. Dostupné z: <https://www.asctimetables.com/>
- [8] Edookit [online]. 2021 [cit. 2022-01-05]. Dostupné z: <https://www.edookit.com/cs/>
- [9] Skolaris [online]. 2021 [cit. 2022-01-05]. Dostupné z: <https://skolaris.net/>
- [10] ASc TimeTables - Online Help [online]. [cit. 2022-01-10]. Dostupné z: <https://help.asctimetables.com/text.php?id=592&lang=en>
- [11] Prime Timetable [online]. 2021 [cit. 2022-01-12]. Dostupné z: <https://primetimetable.com/help/>
- [12] FET Free Timetabling Software [online]. 2021 [cit. 2022-01-15]. Dostupné z: <https://lalescu.ro/liviu/fet/>
- [13] Features. In: FET Free Timetabling Software [online]. [cit. 2022-01-16]. Dostupné z: <https://www.lalescu.ro/liviu/fet/features.html>
- [14] *FET Manual* [online]. 2018 [cit. 2022-01-20]. Dostupné z: <https://www.timetabling.de/manual/FET-manual.en.html>

- [15] LALESCU, Liviu. The description of the FET timetable generation algorithm [online]. In: FET Free Timetabling Software. 2019 [cit. 2022-03-20]. Dostupné z: <https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-description.html>
- [16] LÉVÊQUE Benjamin, MAFFRAY Frédéric. Coloring Meyniel graphs in linear time [online]. In: hal archives ouvertes. 2005 [cit. 2022-04-10]. Dostupné z: <https://hal.archives-ouvertes.fr/hal-00001574v2>
- [17] NP-completeness. In: Wikipedia [online]. San Francisco (CA): Wikimedia Foundation, 2022 [cit. 2022-03-04]. Dostupné z: <https://en.wikipedia.org/wiki/NP-completeness>

Obsah přiloženého CD

Desktopová aplikace Rozvrh

Manuál k aplikaci

Testovací soubory pro generování rozvrhu (nachází se v testovací aplikaci ve složce

„Rozvrh\ vstupni data“):

- Oblačná (počet tříd: 11, počet učitelů: 19, počet místností: 16)
- Velká škola (počet tříd: 17, počet učitelů: 27, počet místností: 24)