Rozvrh

Disclaimer: program se nezaobírá kroužky

Cíl programu: 1) vytvořit databázi s požadovanými informacemi k vytvoření rozvrhu

2) vytvořit funkční rozvrh, s podmínkami dříve určenými

3) předat udělaný rozvrh v uživatelsky přívětivé podobě

1 - Databáze

typ databáze bude sqlite, z důvodu snadné použitelnosti a nenutnosti spouštět celou mysql databázi.

Bude uchovávat tři základní data typy: null, text, integer

Základní stoly: učitel, předmět, místnost, studenti, list

místnost – id místnosti, jméno, matrix, váha

studenti – id studentů, jméno třídy, id preferované místnosti, id studentů (dělené hodiny), id studentů (semináře), id předmětu

List – datová struktura ve formě tabulky, která udává čas

Učitel – id učitele, jméno, matrix, váha, počet tříd

Předmět – id předmětu, list učitelů, list místností, jméno, váha, počet, dělitelnost, seminář.

2 – tvorba

Pomocí databáze se vytvoří základní objekty:

místnost, studenti, učitel, předmět, zpracování

studenti – jméno, čas (matrix)

učitel – váha, jméno, čas (matrix)

předmět – váha, jméno

místnost – jméno, čas (matrix), váha

Zpracování – list studenti

Metody objektů:

studenti – vstup do matrixu()

učitel – výpočet váhy(), kolik hodin zbývá(), vstup do matrixu()

předmět – výpočet váhy(), check dotaci hodin(),

místnost – výpočet váhy(), kolik hodin zbývá(), vstup do matrixu()

Zpracování – byl ten den předmět(), najdi předmět s nejvyšší vahou(), jsou čtyři semináře volné(), check jestli políčko plné(), přepiš matrixy vloženého předmětu/učitele/místnosti(),

3) output – Modul tkinter

3.1) uživatelský input

Viz databáze

3.2) output

Output bude ve formě tabulek s vytvořenými rozvrhy, buď čerpaný z databáze, nebo rovnou vytvořený pomocí zpracování.

4) Popis + diskuse

1. Databáze

Problémy: Efektivnost tvoření matrixů – nejspíše blob datová struktura??

1. Tvorba

Celý program bude fungovat „dynamicky“, databázové hodnoty se budou přepisovat zatím co program bude fungovat a váhy se budou znovu počítat po každé školního hodině.

Každá váha bude mít default hodnotu 1, když váha času nějaké třídy bude null váha půjde automaticky do mínusu

Místnost – váha místnosti přímo ve třídě místnost se bude počítat na základě času, který místnosti zbývá volný. Dále později ve třídě předmět, který drží list místností, se bude počítat s počtem místností, a to tak, že čím méně místností na daný předmět tím více dostane k váze

Učitel – váha učitele se bude počítat obdobně jako u místnosti, ale s menším časem bude dostávat více bodů do váhy.

Předmět – váha předmětu bude součtem místnosti a učitele + další vlastnosti a podmínky. Bude dávat plusové body za to, když bude daný předmět dělený, nebo seminář, nebo jeho počet bude vyšší než 1. a další podmínky

Zpracování – vezme všechny třídy a bude postupně měnit databázi, bude měnit hodnoty matrixů, počtu předmětu. Bude dopočítávat semináře\* a dělené\* hodiny. Bude dávat pozor na podmínky, jestli daný den předmět již byl, jestli daný předmět na dotaci hodinovou. Bude měnit matrixy na základě pravidla, že nejdříve vždy zaplní prvních pět – šest hodin (podle ročníku) a když nedokáže efektivně zaplnit těchto několik hodin u všech dnů, začne přidávat odpolední, prodloužené ranní hodiny.

\*Dělené – když bude možnost dělené hodiny program se zastaví a zůstane na stejném místě a zkontroluje předměty se kterými se dělí, jestli všichni budou mít čas pozmění rovnou v druhé třídě rozvrh

\*semináře – semináře budou fungovat na principu vah, ale s malou změnou a to, že když bude dosazen seminář, tak musí být možnost dosadit další tři semináře na to stejné místo, bude si také nárokovat místo na prostor 2, nebo tři a učitele bude vybírat náhodně podle jejich možností z listu učitelů. Samozřejmě u dalších tříd udělá také změnu

1. Uživatelské prostředí

Tkinter ?