

Tvorba rozvrhu hodin

Vygenerováno programem Doxygen 1.13.0



# Kapitola 1

## Popis programu

Tento program tvoří rozvrhy hodin pro vysoukou školu ČVUT pro děčínské pracoviště. Program tvoří maximální počet rozvrhů hodin, které dokáže vytvořit. Po vytvoření se rozvrhy vypisují do gui, kde uživatel má možnost si vytvořené rozvrhy prohlížet.

### 1.1 Vstupní data

Program čerpá data zapsaná v 7 vstupních souborech formátu CSV nebo JSON:

#### 1.1.1 *obor\_predmet.csv*

```
obor, předmět, počet přednášek
AIPV_Dc-Bc-1_1, 80DTB2, 1
AIPV_Dc-Bc-1_1, 80LI2, 1
AIPV_Dc-Bc-1_1, 80LI2cv, 1
```

#### 1.1.2 *predmet\_delkaPredmetu.csv*

```
predmět, délka předmětu v hodinách
16PADR, 2
16SEM, 2
16ZPRD, 3
```

#### 1.1.3 *predmet\_mistnost\_poradi.csv*

Pořadím se rozumí pořadí, ve kterém se budou volit místnosti pro zápis předmětů. Pokud první bude obsazená, na řadu přijde druhá, atd.

```
předmět, místnost, pořadí
16PADR, Dc-PL1, 1
16SEM, Dc-FL, 1
16ZPRD, Dc-P4, 1
```

#### 1.1.4 *predmety\_napevno.csv*

Tento soubor obsahuje předměty, které chcete umístit do rozvrhu předem. Při tvorbě rozvrhu se tyto předměty zapíší do šablony rozvrhu a teprve pak program tak začne vpisovat předměty. Den je brán od 1 (1 = pondělí). Čas je hodina, od které se přednáška začíná vyučovat zmenšená o 7 (1 = od 8h).

```
Předmět, Učitel, Místnost, Den, Čas, Délka přednášky
16SEM, JmenoVyucujiciho_1, Dc-pom, 3, 2, 2
80MAN2cv, JmenoVyucujiciho_1, Dc-P2, 5, 1, 2
80MAN2cv, JmenoVyucujiciho_2, Dc-P2, 5, 3, 2
```

### 1.1.5 vyucujici\_predmet\_pocetPrednasek.csv

Jméno vyučujícího, předmět, počet přednášek  
 JmenoVyucujiciho\_1,80XA0,1  
 JmenoVyucujiciho\_1,80XA2,1  
 JmenoVyucujiciho\_2,80FY2,1

### 1.1.6 mistnosti\_casy.json

Tento soubor obsahuje volna učeben. Pokud je na první pozici 1, tak učebna je pro časový slot od 8h do 9h volná. Pokud je tam 0, tak se obsazená. Tohoto se dá využít, když potřebujete místnost využít pro nějaký kroužek, aby se do této místnosti nezapisovaly předměty ve zvolený čas.

```
[
  {
    "name": "Dc-P2",
    "Po": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ut": [1,1,1,1,1,1,1,1,0,0,0,1],
    "St": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ct": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Pa": [1,1,1,1,1,1,1,1,1,1,1,1]
  },
  {
    "name": "Dc-P4",
    "Po": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ut": [1,1,1,1,1,1,1,1,1,1,1,1],
    "St": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ct": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Pa": [1,1,1,1,1,1,1,1,1,1,1,1]
  }
]
```

### 1.1.7 vyucujici\_casy.json

Tento soubor obsahuje volna vyučujících. Pokud je na první pozici 1, tak vyučující má pro časový slot od 8h do 9h volno. Pokud je tam 0, tak volno nemá, tedy nemůže vyučovat.

```
[
  {
    "name": "JmenoVyucujiciho_1",
    "Po": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ut": [1,1,1,1,1,1,1,1,0,0,0,1],
    "St": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ct": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Pa": [1,1,1,1,1,1,1,1,1,1,1,1]
  },
  {
    "name": "JmenoVyucujiciho_2",
    "Po": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ut": [1,1,1,1,1,1,1,1,1,1,1,1],
    "St": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Ct": [1,1,1,1,1,1,1,1,1,1,1,1],
    "Pa": [1,1,1,1,1,1,1,1,1,1,1,1]
  }
]
```

Soubory formátu CSV mají první řádek volný. Tento řádek je určen pro případné komentáře a při běhu kódu bude vynechán. Ani před ani za (pokud je za tímto sloupcem další sloupec) sloupce nesmí být mezera.

## 1.2 Běh programu

Program ukončí svůj běh a vypíše všechny chyby co našel do souboru *logs/chyby.txt*.

[CHYBA] V souboru: predmety\_napevno.csv, na řádku: 20, místnost NENÍ volná tento čas.

Pokud chyby nalezeny nebyly, pak program pokračuje ke tvorbě rozvrhů. Program vypisuje do konzole uživateli informaci, zda nějaké rozvrhy našel. Pokud žádné nenašel je potřeba se podívat do XLSX souboru *vlivy\_predmety\_hodnoty\_nezapPredmety.xlsx*. Tento excelový soubor obsahuje data o nevytvořených rozvrzích obsažená ve 4 listech:

1. vektory vlivů
2. posloupnost předmětů, ve kterém se předměty zapisovaly
3. hodnocení předmětů v primárním řazení (první řádek)
4. nezapsané předměty

Tento soubor se vytvoří i v případě, že nějaké rozvrhy byly nalezeny. Data tohoto souboru by měli pomoci nalézt případné příčiny nenalezení rozvrhů. Pokud se rozvrhy najdou, tak se spustí gui: Nahoře máte možnost vybírat vytvořené rozvrhy a poté stačí zmáčknout "Zobrazit rozvrhy". Tím se zobrazí rozvrhy pro všechny obory. Vytvořené rozvrhy jsou různé. Číslo značí index sady rozvrhů. Do konzole se také vypíše hodnoty:

```
Maximální povolený počet přepsání předmětů: [3, 23]
Nalezeno rozvrhu: 5376
Ruznych rozvrhu je: 55
Kód běžel 25.36 sekund.
```

Pokud se spustí gui, tak to znamená že se rozvrh našel. V případě, že ne, tak se jedná a chyby, vypsané do souboru *chyby.txt*, nebo o příliš striktní omezení vyučujících a místností.

## 1.3 Požadavky

- Python 3.9
- pip (Python package installer)

## 1.4 Instalace

1. Přesuňte se do složky s algoritmem
2. V průzkumníku souborů klikněte na "Domů" a spusťte PowerShell
3. Vytvořte virtuální prostředí:

```
py -m venv venv
```

4. Aktivujte virtuální prostředí:

- Windows:

```
venv\Scripts\activate
```

- macOS/Linux:

```
source venv/bin/activate
@icode
```

5. Nainstalujte závislosti:

```
py -m pip install -r requirements.txt
```

6. Spusťte program

```
py main.py
```



## Kapitola 2

# Rejstřík jmenných prostorů

### 2.1 Seznam jmenných prostorů

Zde naleznete seznam všech jmenných prostorů se stručným popisem:

<a href="#">gui</a>	??
<a href="#">hledej_a_zapis_pred_napevno</a>	??
<a href="#">hodnoceni_predmetu</a>	??
<a href="#">hodnoceni_vytvoreneho_rozvrhu</a>	??
<a href="#">kontroly</a>	??
<a href="#">LP2</a>	??
<a href="#">main</a>	??
<a href="#">nacti_data</a>	??
<a href="#">pomocne_funkce</a>	??
<a href="#">tvorba_rozvrhu</a>	??
<a href="#">tvorba_seznamu_z_nactenych_dat</a>	??
<a href="#">vypis_pomocnych_udaju</a>	??
<a href="#">vypis_vytvorených_rozvrhu</a>	??





## Kapitola 3

# Rejstřík souborů

### 3.1 Seznam souborů

Zde naleznete seznam všech souborů se stručnými popisy:

<a href="#">gui.py</a>	??
<a href="#">hledej_a_zapis_pred_napevno.py</a>	??
<a href="#">hodnoceni_predmetu.py</a>	??
<a href="#">hodnoceni_vytvoreneho_rozvrhu.py</a>	??
<a href="#">kontroly.py</a>	??
<a href="#">LP2.py</a>	??
<a href="#">main.py</a>	??
<a href="#">nacti_data.py</a>	??
<a href="#">pomocne_funkce.py</a>	??
<a href="#">tvorba_rozvrhu.py</a>	??
<a href="#">tvorba_seznamu_z_nactenych_dat.py</a>	??
<a href="#">vypis_pomocnych_udaju.py</a>	??
<a href="#">vypis_vytvorenych_rozvrhu.py</a>	??



## Kapitola 4

# Dokumentace jmenných prostorů

### 4.1 Dokumentace jmenného prostoru gui

#### Funkce

- [Prevod\\_Rozvrhu](#) (rozvrhy, index, seznam\_delkaPredmetu\_predmet)
- [Proved\\_PrevodUpravRozvrhuProOboryNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [Proved\\_PrevodUpravRozvrhuProMistnostiNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [Proved\\_PrevodUpravRozvrhuProVyucujiciNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [Vypis\\_rozvrhu](#) (rozvrh\_oboru, canvas\_frame, obor\_name, index)
- [Vykresli\\_Rozvrh](#) (rozvrh\_combobox, canvas\_frame, canvas\_container, rozvrhy, seznam\_oboru, seznam\_mistnosti, seznam\_vyucujících, seznam\_predmetu, delky\_predmetu, rozvrh\_combobox2)
- [on\\_close](#) (root)
- [on\\_mouse\\_wheel](#) (event, canvas\_container)
- [Vypis\\_Gui](#) (rozvrhy, seznam\_oboru, seznam\_mistnosti, seznam\_vyucujících, seznam\_predmetu, delky\_predmetu)

#### Proměnné

- list [dny](#) = ["Pondělí", "Úterý", "Středa", "Čtvrtek", "Pátek"]

#### 4.1.1 Dokumentace funkcí

##### 4.1.1.1 on\_close()

```
gui.on_close (  
    root)
```

```
zavírá okno při kliknutí na křížek  
:param root: root  
:return: None
```

#### 4.1.1.2 on\_mouse\_wheel()

```
gui.on_mouse_wheel (
    event,
    canvas_container)

umožňuje scrollování kolečkem myši
:param event: událost scrollování
:param canvas_container: kontejnej pláten
:return:
```

#### 4.1.1.3 Prevod\_Rozvrhu()

```
gui.Prevod_Rozvrhu (
    rozvrhy,
    index,
    seznam_delkaPredmetu_predmet)

převádí zapsané předměty do seznamu
:param rozvrhy:
:param index:
:param seznam_delkaPredmetu_predmet:
:return: seznam [den, predmet, cas, cas + delka_prednasky, vyucujici, mistnost, obor]
```

#### 4.1.1.4 Proved\_PrevodUpravRozvrhuProMistnostiNaSlovník()

```
gui.Proved_PrevodUpravRozvrhuProMistnostiNaSlovník (
    upraveny_rozvrh,
    seznam_oboru,
    seznam_predmetu,
    seznam_mistnosti,
    seznam_vyucujicich)

převádí seznamy rozvrhů, které obsahují potřebná data pro výpis rozvrhů pro místnosti, na slovník
:param upraveny_rozvrh: rozvrh (seznam), kde každý předmět je převeden na seznam [den, predmet, cas, cas + delka_prednasky, vyucujici, mistnost, obor]
:param seznam_oboru: seznam oborů
:param seznam_predmetu: seznam předmětů
:param seznam_mistnosti: seznam místností
:param seznam_vyucujicich: seznam vyučujících
:return: rozvrh v podobě slovníku
```

#### 4.1.1.5 Proved\_PrevodUpravRozvrhuProOboryNaSlovník()

```
gui.Proved_PrevodUpravRozvrhuProOboryNaSlovník (
    upraveny_rozvrh,
    seznam_oboru,
    seznam_predmetu,
    seznam_mistnosti,
    seznam_vyucujicich)

převádí seznamy rozvrhů, které obsahují potřebná data pro výpis rozvrhů pro obory, na slovník
:param upraveny_rozvrh: rozvrh (seznam), kde každý předmět je převeden na seznam [den, predmet, cas, cas + delka_prednasky, vyucujici, mistnost, obor]
:param seznam_oboru: seznam oborů
:param seznam_predmetu: seznam předmětů
:param seznam_mistnosti: seznam místností
:param seznam_vyucujicich: seznam vyučujících
:return: rozvrh v podobě slovníku
```

#### 4.1.1.6 Proved\_PrevodUpravRozvrhuProVyucejiciNaSlovník()

```
gui.Proved_PrevodUpravRozvrhuProVyucejiciNaSlovník (
    upraveny_rozvrh,
    seznam_oboru,
    seznam_predmetu,
    seznam_mistnosti,
    seznam_vyucujicich)
```

převádí seznamy rozvrhů, které obsahují potřebná data pro výpis rozvrhů pro vyučující, na slovník  
:param upraveny\_rozvrh: rozvrh (seznam), kde každý předmět je převeden na seznam [den, predmet, cas, cas + delka]  
:param seznam\_oboru: seznam oborů  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_mistnosti: seznam místností  
:param seznam\_vyucujicich: seznam vyučujících  
:return: rozvrh v podobě slovníku

#### 4.1.1.7 Vykresli\_Rozvrh()

```
gui.Vykresli_Rozvrh (
    rozvrh_combobox,
    canvas_frame,
    canvas_container,
    rozvrhy,
    seznam_oboru,
    seznam_mistnosti,
    seznam_vyucujicich,
    seznam_predmetu,
    delky_predmetu,
    rozvrh_combobox2)
```

vykresluje rozvrh do gui  
:param rozvrh\_combobox: combobox výběru indexu vytvořeného rozvrhu  
:param canvas\_frame: plátno  
:param canvas\_container: kontejnej pláten  
:param rozvrhy: rozvrhy všech oborů  
:param seznam\_oboru: seznam oborů  
:param seznam\_mistnosti: seznam místností  
:param seznam\_vyucujicich: seznam vyučujících  
:param seznam\_predmetu: seznam předmětů  
:param delky\_predmetu: seznam délek předmětů  
:param rozvrh\_combobox2: combobox výběru indexu dat pro výpis (rozvrh pro Obory / Místnosti / Vyučující)  
:return: None

#### 4.1.1.8 Vypis\_Gui()

```
gui.Vypis_Gui (
    rozvrhy,
    seznam_oboru,
    seznam_mistnosti,
    seznam_vyucujicich,
    seznam_predmetu,
    delky_predmetu)
```

hlavní funkce pro výpis a interakce s gui  
:param rozvrhy: vytvořené rozvrhy  
:param seznam\_oboru: seznam oborů  
:param seznam\_mistnosti: seznam místností  
:param seznam\_vyucujicich: seznam vyučujících  
:param seznam\_predmetu: seznam předmětů  
:param delky\_predmetu:  
:return: None

#### 4.1.1.9 Vypis\_rozvrhu()

```
gui.Vypis_rozvrhu (
    rozvrh_oboru,
    canvas_frame,
    obor_name,
    index)

vypisuje rozvrh jednoho oboru
:param rozvrh_oboru: rozvrh jednoho oboru
:param canvas_frame: plátno
:param obor_name: název oboru
:param index: index data co chci vypsát (rozvrh pro Obory - 0 / Místnosti - 1 / Vyučující - 2)
:return: None
```

### 4.1.2 Dokumentace proměnných

#### 4.1.2.1 dny

```
list gui.dny = ["Pondělí", "Úterý", "Středa", "Čtvrtek", "Pátek"]
```

## 4.2 Dokumentace jmenného prostoru hledej\_a\_zapis\_pred\_napevno

### Funkce

- [int, int] [Zjisti\\_KdyZacinaUcit](#) (int vyuc, list[list[int]] vyucujici\_casy)
- int [Zjisti\\_PocetPrednasekVyuc](#) (pppdv, vyuc)
- [Proved\\_ZapisPredmetuNapevno](#) (data\_o\_jiz\_zapsanych\_p, vyucujici\_casy, pppdv, seznam\_delkaPredmetu\_predmet, seznam\_vyuc, pppdo, pref\_mistnosti, seznam\_predmetu, seznam\_vyucujicich, seznam\_mistnosti, mistnosti\_casy)
- bool [Zjisti\\_Uci1ci2PredmetyPro1obor](#) (seznam\_predmety\_pp\_delky, pppdo)

#### 4.2.1 Dokumentace funkcí

##### 4.2.1.1 Proved\_ZapisPredmetuNapevno()

```
hledej_a_zapis_pred_napevno.Proved_ZapisPredmetuNapevno (
    data_o_jiz_zapsanych_p,
    vyucujici_casy,
    pppdv,
    seznam_delkaPredmetu_predmet,
    seznam_vyuc,
    pppdo,
    pref_mistnosti,
    seznam_predmetu,
    seznam_vyucujicich,
    seznam_mistnosti,
    mistnosti_casy)
```

Přidává data o předmětu do seznamu obsahující předměty, co se mají zapsat do šablony, pokud je jasné z dat, že

```
:param data_o_jiz_zapsanych_p: data natvrdo zapsaných předmětů
:param vyucujici_casy: data volných hodin vyučujících
:param pppdv: data přednášek vyučujících
:param seznam_delkaPredmetu_predmet: seznam délek předmětů (index = předmět, hodnota = délka)
:param seznam_vyuc: seznam vyučujících
:param pppdo: data předmětů oborů
:param pref_mistnosti: v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = pořadí
:param seznam_predmetu: seznam předmětů
:param seznam_vyucujicich: seznam vyučujících
:param seznam_mistnosti: seznam místností
:param mistnosti_casy: data volných hodin místností
:return: None
```

#### 4.2.1.2 Zjisti\_KdyZacinaUcit()

```
[int, int] hledej_a_zapis_pred_napevno.Zjisti_KdyZacinaUcit (
    int vyuc,
    list[list[int]] vyucujici_casy)
```

funkce vrátí číslo dne (0-4) a číslo hodiny (0-11)

```
:param vyuc: index vyučujícího
:param vyucujici_casy: data volných hodin vyučujících
:return: index dne od 0, index hodiny od 0
```

#### 4.2.1.3 Zjisti\_PocetPrednasekVyuc()

```
int hledej_a_zapis_pred_napevno.Zjisti_PocetPrednasekVyuc (
    pppdv,
    vyuc)
```

vrátí celkový počet přednášek, co má tento vyučující odučit

```
:param pppdv: data přednášek vyučujících
:param vyuc: index vyučujícího
:return: počet přednášek
```

#### 4.2.1.4 Zjisti\_Uci1ci2PredmetyPro1obor()

```
bool hledej_a_zapis_pred_napevno.Zjisti_Uci1ci2PredmetyPro1obor (
    seznam_predmety_pp_delky,
    pppdo)
```

Zjistí, zda vyučující učí 1 nebo 2 předměty pro jeden obor

```
:param seznam_predmety_pp_delky: seznam obsahující délku přednášky a jejich počet ke každému předmětu
:param pppdo: data předmětů oborů
:return: bool
```

## 4.3 Dokumentace jmenného prostoru hodnoceni\_predmetu

### Funkce

- list[float] [Proved\\_Hodnoceni\\_Predmetu](#) (list[list[int]] pppdv, list[list[list[int]]] vyucujici\_casy, list[str] predmety, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[list[int]] pppdo, list[int] index\_vlivu)
- int [Zjisti\\_JakyDenPrevazneUci](#) (list[list[list[int]]] vyucujici\_casy, list[int] vyucujici)
- list[int] [Zjisti\\_Seznam\\_Vyucujicich](#) (list[list[int]] pppdv, int predmet)
- int [Zjisti\\_Pocet\\_Vyucovanych\\_Prednasek](#) (list[list[int]] pppdv, list[int] seznam\_vyucujicich\_pro\_tento\_predmet, int predmet)
- [list[int], list[int], list[float]] [Vytvor\\_SeznamPoradiPredmetu](#) (list[list[int]] pppdv, list[list[list[int]]] vyucujici\_casy, list[str] seznam\_predmetu, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[list[int]] pppdo)
- list[int] [Uprav\\_Skupiny](#) (dict[list[int]:float] skup)
- list[list[int]] [VytvorVlivy](#) ()

### 4.3.1 Dokumentace funkcí

#### 4.3.1.1 Proved\_Hodnoceni\_Predmetu()

```
list[float] hodnoceni_predmetu.Proved_Hodnoceni_Predmetu (
    list[list[int]] pppdv,
    list[list[list[int]]] vyucujici_casy,
    list[str] predmety,
    list[int] seznam_delkaPredmetu_predmet,
    list[list[int]] preference_mistnosti,
    list[list[int]] pppdo,
    list[int] index_vlivu)
```

funkce, která hodnotí předměty

:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího  
:param vyucujici\_casy: v řádcích jsou vyučující, ve sloupcích dny, 1 -> má volno, 0 -> nemá volno  
:param predmety: seznam předmětů  
:param seznam\_delkaPredmetu\_predmet: index je predmet, hodnota je delka prednasky  
:param preference\_mistnosti: v řádcích jsou místnosti, ve sloupcích předměty, hodnoty = poradi  
:param pppdo: pokud je ve sloupci více nenulových čísel -> pro tyto řádky (obory) je tento sloupec (předmět) sp  
:param index\_vlivu: seznam vlivů na každou složku hodnocení předmětů  
:return: seznam hodnocení předmětů

#### 4.3.1.2 Uprav\_Skupiny()

```
list[int] hodnoceni_predmetu.Uprav_Skupiny (
    dict[list[int]: float] skup)
```

Vytváří seznam s pořadovými čísly předmětů.

:param skup: slovník, [[předmětX, předmětY, ...]: hodnota, [předmětZ, předmětW, ...]: hodnota], ...  
:return: seznam s pořadovými čísly předmětů



#### 4.3.1.3 Vytvor\_SeznamPoradiPredmetu()

```
[list[int], list[int], list[float]] hodnoceni_predmetu.Vytvor_SeznamPoradiPredmetu (
    list[list[int]] pppdv,
    list[list[list[int]]] vyucujici_casy,
    list[str] seznam_predmetu,
    list[int] seznam_delkaPredmetu_predmet,
    list[list[int]] preference_mistnosti,
    list[list[int]] pppdo)
```

Vytváří všechny možné posloupnosti předmětů s použitím vlivů.

```
:param pppdv: řádky jsou vyučující, sloupce jsou předměty a hodnoty jsou počet přednášek vyučujícího k odučení
:param vyucujici_casy: data volen vyučujících
:param seznam_predmetu: seznam s kódy předmětů
:param seznam_delkaPredmetu_predmet: seznam s délkami předmětů
:param preference_mistnosti: v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = poradi
:param pppdo: pokud je ve sloupci více nenulových čísel -> pro tyto řádky (obory) je tento sloupec (předmět) spočítán
:return: seznam s pořadovými čísly předmětů, použitý vektor vlivů, seznam hodnocení předmětů
```

#### 4.3.1.4 VytvorVlivy()

```
list[list[int]] hodnoceni_predmetu.VytvorVlivy ()
```

Vytváří vlivy, kterými násobím složky hodnocení předmětů.

```
:return: seznam vektorů
```

#### 4.3.1.5 Zjisti\_JakyDenPrevazneUci()

```
int hodnoceni_predmetu.Zjisti_JakyDenPrevazneUci (
    list[list[list[int]]] vyucujici_casy,
    list[int] vyucujici)
```

Vrací index dne, kdy vyučující může učit nejdéle.

```
:param vyucujici_casy: data volných hodin vyučujících
:param vyucujici: seznam indexů vyučujících
:return: index dne
```

#### 4.3.1.6 Zjisti\_Pocet\_Vyucovanych\_Prednasek()

```
int hodnoceni_predmetu.Zjisti_Pocet_Vyucovanych_Prednasek (
    list[list[int]] pppdv,
    list[int] seznam_vyucujicich_pro_tento_predmet,
    int predmet)
```

Vrací počet všech vyučovaných přednášek tohoto předmětu.

```
:param pppdv: řádky jsou vyučující, sloupce jsou předměty a hodnoty jsou počet přednášek vyučujícího k odučení
:param seznam_vyucujicich_pro_tento_predmet: seznam indexů vyučujících co tento předmět vyučují
:param predmet: index předmětu v seznamu předmětů
:return: počet přednášek tohoto předmětu k odučení
```

#### 4.3.1.7 Zjisti\_Seznam\_Vyucujicich()

```
list[int] hodnoceni_predmetu.Zjisti_Seznam_Vyucujicich (
    list[list[int]] pppdv,
    int predmet)
```

Vrací seznam vyučujících co udí daný předmět.

:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího

:param predmet: index předmětu v seznamu předmětů

:return: vrací seznam vyučujících tohoto předmětu

## 4.4 Dokumentace jmenného prostoru hodnoceni\_vytvoreneho\_rozvrhu

### Funkce

- `[list[list[int]], list]` [Ohodnot\\_Rozvrh](#) (`list[rozvrh_type]` vytvorene\_rozvrhy, `rozvrh_type` rozvrh, `list[list[int]]` pppdo, `list[str]` seznam\_predmetu, `list[int]` pouzite\_vlivy, `list[list[int]]` seznam\_vlivu\_vytvorených\_rozvrhu, `seznam_vliv_predmety_hodnoty_nezapsaneP`, `list[int]` predmety\_dle\_poradi, `list[float]` hodnoty\_premetu)
- `list[list[int]]` [Najdi\\_Nezapsane\\_Predmety](#) (`list[list[int]]` pppdo, `list[str]` predmety)
- `list[int]` [Zjisti\\_Interval](#) (`rozvrh_type` rozvrh, `int` obor, `int` den)

### Proměnné

- `rozvrh_type` = list

### 4.4.1 Dokumentace funkcí

#### 4.4.1.1 Najdi\_Nezapsane\_Predmety()

```
list[list[int]] hodnoceni_vytvoreneho_rozvrhu.Najdi_Nezapsane_Predmety (
    list[list[int]] pppdo,
    list[str] predmety)
```

Vrací seznam dvojic předmět-počet\_přednášek nezapsaných předmětů

:param pppdo: pokud je ve sloupci více nenulových čísel -> pro tyto řádky (obory) je tento sloupec (předmět) sp

:param predmety: seznam předmětů

:return: seznam dvojic [index\_předmětu, počet\_přednášek]

#### 4.4.1.2 Ohodnot\_Rozvrh()

```
[list[list[int]], list] hodnoceni_vytvoreneho_rozvrhu.Ohodnot_Rozvrh (
    list[rozvrh_type] vytvorene_rozvrhy,
    rozvrh_type rozvrh,
    list[list[int]] pppdo,
    list[str] seznam_predmetu,
    list[int] pouzite_vlivy,
    list[list[int]] seznam_vlivu_vytvorenych_rozvrhu,
    seznam_vliv_predmety_hodnoty_nezapsaneP,
    list[int] predmety_dle_poradi,
    list[float] hodnoty_premetu)

:param vytvorene_rozvrhy: seznam všech vytvořených rozvrhů
:param rozvrh: vytvořený rozvrh
:param pppdo: data předmětů oborů
:param seznam_predmetu:
:param pouzite_vlivy: "vektor" (seznam) vlivů
:param seznam_vlivu_vytvorenych_rozvrhu: seznam "vektorů" (seznamů) vlivů
:param seznam_vliv_predmety_hodnoty_nezapsaneP: seznam s pomocnými daty
:param predmety_dle_poradi: seznam pořadí předmětů
:param hodnoty_premetu: seznam hodnocení předmětů
:return: seznam nezapsaných předmětů, seznam s pomocnými daty
```

#### 4.4.1.3 Zjisti\_Interval()

```
list[int] hodnoceni_vytvoreneho_rozvrhu.Zjisti_Interval (
    rozvrh_type rozvrh,
    int obor,
    int den)

Vrací dvě hodnoty určující interval
:param rozvrh: vytvořený rozvrh
:param obor: index oboru
:param den: index dne (0-4)
:return: seznam, od: první vyučovaná hodina, do: poslední vyučovaná hodina (tento den)
```

### 4.4.2 Dokumentace proměnných

#### 4.4.2.1 rozvrh\_type

```
hodnoceni_vytvoreneho_rozvrhu.rozvrh_type = list
```

## 4.5 Dokumentace jmenného prostoru kontroly

### Funkce

- list[str] [Kontrola\\_Vst\\_Dat](#) (list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_delkaPredmetu, list[list[str]] seznam\_predmet\_mistnost\_poradi, list[list[str]] seznam\_vyucujici\_predmet\_pocetPrednasek, list[list[str]] data\_o\_jiz\_zapsanych\_p, list[int] delky\_predmetu, list[str] seznam\_vyucujicich, list[str] seznam\_mistnosti, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] mistnosti\_casy, list[list[int]] pppdv, list[int] seznam\_delkaPredmetu\_predmet)

- None [Kontrola\\_PredmetuNapevnoZapsanych](#) (list[str] chyby, list[list[str]] data\_o\_jiz\_zapsanych\_p, list[str] seznam\_predmetu, list[str] seznam\_vyucujicich, list[str] seznam\_mistnosti, list[int] delky\_predmetu, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] mistnosti\_casy)
- bool [Kontrola\\_JsouSpravneHodnoty](#) (int cislo, int od, int do)
- bool [Kontrola\\_JeSpravneKod](#) (str kod, list[str] seznam)
- bool [Zjisti\\_JeZdePredmetNavic](#) (list[str] seznam\_predmetu, list kontrolovany\_seznam, int predmet, int index)
- bool [Zjisti\\_ZdaChybiPredmet](#) (list[str] seznam\_predmetu, list kontrolovany\_seznam, int radek, int index)
- None [Kontrola\\_PredmetDelkaPrednasky](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_delkaPredmetu)
- None [Kontrola\\_PredmetMistnostPoradi](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_mistnost\_poradi)
- None [Kontrola\\_VyucujiciPredmetPocetPrednasek](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_vyucujici\_predmet\_pocetPrednasek)
- None [Kontrola\\_CasuVyucujicich](#) (list[str] chyby, list[list[list[int]]] vyucujici\_casy, list[list[int]] pppdv, list[int] seznam\_delkaPredmetu\_predmet, list[str] seznam\_vyuc)
- [Kontrola\\_DelekPrednasek](#) (int vyuc, list[list[list[int]]] vyucujici\_casy, list[list[int]] seznam\_predmety\_pp\_delky, list[str] seznam\_vyuc)
- None [Vypis\\_Chyby](#) (list[str] chyby, list[str] upozorneni)
- int [Zjisti\\_MaxPocetJednicekZaSebou](#) (list[list[list[int]]] vyucujici\_casy, int vyuc, list[str] seznam\_vyuc)

## 4.5.1 Dokumentace funkcí

### 4.5.1.1 Kontrola\_CasuVyucujicich()

```
None kontroly.Kontrola_CasuVyucujicich (
    list[str] chyby,
    list[list[list[int]]] vyucujici_casy,
    list[list[int]] pppdv,
    list[int] seznam_delkaPredmetu_predmet,
    list[str] seznam_vyuc)
```

Kontroluje, zda počet volných hodin odpovídá potřebnému počtu hodin k odučení.  
Kontroluje, zda se do úseku volen vejdou všechny přednášky z hlediska délek.

```
:param chyby: seznam s chybami
:param vyucujici_casy: data volných hodin vyučujících
:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího
:param seznam_delkaPredmetu_predmet: seznam délek předmětů
:param seznam_vyuc: seznam vyučujících
:return: None
```

### 4.5.1.2 Kontrola\_DelekPrednasek()

```
kontroly.Kontrola_DelekPrednasek (
    int vyuc,
    list[list[list[int]]] vyucujici_casy,
    list[list[int]] seznam_predmety_pp_delky,
    list[str] seznam_vyuc)
```

Kontroluje, zda jsou délky přednášek 2, 3 nebo 4.

```
:param vyuc: index vyučujícího
:param vyucujici_casy: data volných hodin vyučujících
:param seznam_predmety_pp_delky: pomocný seznam obsahující délky přednášek a jejich počet
:param seznam_vyuc: seznam vyučujících
:return: bool
```

#### 4.5.1.3 Kontrola\_JeSpravneKod()

```
bool kontroly.Kontrola_JeSpravneKod (  
    str kod,  
    list[str] seznam)
```

Kontroluje, zda kód uvedený v predmety\_napevno.csv místnosti/vyučujícího/oboru/předmětu lze najít v seznamu m/  
:param kod: kód místnosti/vyučujícího/oboru/předmětu  
:param seznam: seznam, ve kterém chci tento kód hledat  
:return: je v seznamu / není v seznamu

#### 4.5.1.4 Kontrola\_JsouSpravneHodnoty()

```
bool kontroly.Kontrola_JsouSpravneHodnoty (  
    int cislo,  
    int od,  
    int do)
```

Kontroluje, zda čísla uvedená v predmety\_napevno.csv jsou ve správném intervalu.  
:param cislo: číslo, které chci zkontrolovat, zda patří do svého intervalu  
:param od: první hodnota intervalu  
:param do: poslední hodnota intervalu (včetně)  
:return: patří/nepatří

#### 4.5.1.5 Kontrola\_PredmetDelkaPrednasky()

```
None kontroly.Kontrola_PredmetDelkaPrednasky (  
    list[str] chyby,  
    list[str] seznam_predmetu,  
    list[list[str]] seznam_predmet_delkaPredmetu)
```

Kontroluje, zda v souboru predmet\_delkaPredmetu.csv jsou přítomny všechny předměty.  
:param chyby: seznam s chybami  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_predmet\_delkaPredmetu: seznam načtených dat z csv  
:return: Noe

#### 4.5.1.6 Kontrola\_PredmetMistnostPoradi()

```
None kontroly.Kontrola_PredmetMistnostPoradi (  
    list[str] chyby,  
    list[str] seznam_predmetu,  
    list[list[str]] seznam_predmet_mistnost_poradi)
```

Kontroluje, zda v souboru predmet\_mistnost\_poradi.csv jsou přítomny všechny předměty.  
:param chyby: seznam s chybami  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_predmet\_mistnost\_poradi: seznam načtených dat z csv  
:return: None

#### 4.5.1.7 Kontrola\_PredmetuNapevnoZapsanych()

```
None kontroly.Kontrola_PredmetuNapevnoZapsanych (
    list[str] chyby,
    list[list[str]] data_o_jiz_zapsanych_p,
    list[str] seznam_predmetu,
    list[str] seznam_vyucujicich,
    list[str] seznam_mistnosti,
    list[int] delky_predmetu,
    list[list[list[int]]] vyucujici_casy,
    list[list[list[int]]] mistnosti_casy)
```

Hledá chyby a ukládá do seznamu chyb.  
:param chyby: seznam chybových hlášek  
:param data\_o\_jiz\_zapsanych\_p: seznam načtených dat z csv  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_vyucujicich: seznam vyučujících  
:param seznam\_mistnosti: seznam místností  
:param delky\_predmetu: seznam délek předmětů  
:param vyucujici\_casy: data volných hodin vyučujících  
:param mistnosti\_casy: data volných hodin místností  
:return: None

#### 4.5.1.8 Kontrola\_Vst\_Dat()

```
list[str] kontroly.Kontrola_Vst_Dat (
    list[str] seznam_predmetu,
    list[list[str]] seznam_predmet_delkaPredmetu,
    list[list[str]] seznam_predmet_mistnost_poradi,
    list[list[str]] seznam_vyucujici_predmet_pocetPrednasek,
    list[list[str]] data_o_jiz_zapsanych_p,
    list[int] delky_predmetu,
    list[str] seznam_vyucujicich,
    list[str] seznam_mistnosti,
    list[list[list[int]]] vyucujici_casy,
    list[list[list[int]]] mistnosti_casy,
    list[list[int]] pppdv,
    list[int] seznam_delkaPredmetu_predmet)
```

:param seznam\_predmetu: seznam předmětů načtených z csv  
:param seznam\_predmet\_delkaPredmetu: seznam délek předmětů  
:param seznam\_predmet\_mistnost\_poradi: seznam načtených dat z csv  
:param seznam\_vyucujici\_predmet\_pocetPrednasek: seznam načtených dat z csv  
:param data\_o\_jiz\_zapsanych\_p: seznam načtených dat z csv  
:param delky\_predmetu: seznam délek předmětů  
:param seznam\_vyucujicich: seznam kódů vyučujících  
:param seznam\_mistnosti: seznam kódů místností  
:param vyucujici\_casy: v řádcích jsou vyučující, ve sloupcích dny, 1 -> má volno, 0 -> nemá volno  
:param mistnosti\_casy: v řádcích jsou místnosti, ve sloupcích dny s hodinami, 1 -> je volno, 0 -> není volno  
:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího  
:param seznam\_delkaPredmetu\_predmet: index je predmet, hodnota je delka prednasky  
:return: seznam nalezených chyb

#### 4.5.1.9 Kontrola\_VyucujiciPredmetPocetPrednasek()

```
None kontroly.Kontrola_VyucujiciPredmetPocetPrednasek (
    list[str] chyby,
    list[str] seznam_predmetu,
    list[list[str]] seznam_vyucujici_predmet_pocetPrednasek)
```

Kontroluje, zda v souboru vyucujici\_predmet\_pocetPrednasek.csv jsou přítomny všechny předměty.  
:param chyby: seznam s chybami  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_vyucujici\_predmet\_pocetPrednasek: seznam načtených dat z csv  
:return: None

#### 4.5.1.10 Vypis\_Chyby()

```
None kontroly.Vypis_Chyby (
    list[str] chyby,
    list[str] upozorneni)
```

Vypisuje chyby do souboru logs/chyby.txt  
:param chyby: seznam nalezených chyb  
:return: None

#### 4.5.1.11 Zjisti\_JeZdePredmetNavic()

```
bool kontroly.Zjisti_JeZdePredmetNavic (
    list[str] seznam_predmetu,
    list[kontrolovany_seznam,
    int predmet,
    int index)
```

Zjistí, zda je v seznamu předmět navíc.  
:param seznam\_predmetu: seznam předmětů  
:param kontrolovany\_seznam: seznam, ve kterém mám předmět hledat  
:param predmet: index předmětu v seznamu předmětů  
:param index:  
:return:

#### 4.5.1.12 Zjisti\_MaxPocetJednicekZaSebou()

```
int kontroly.Zjisti_MaxPocetJednicekZaSebou (
    list[list[list[int]]] vyucujici_casy,
    int vyuc,
    list[str] seznam_vyuc)
```

Hledá maximální počet jedniček za sebou v seznamu vyučující časy -> čím větší úsek tím délejší předmět se může z  
:param vyucujici\_casy: data volných hodin vyučujících  
:param vyuc: index vyučujícího v seznamu vyučujících  
:param seznam\_vyuc: seznam vyučujících  
:return: maximální počet jedniček za sebou jdoucích

#### 4.5.1.13 Zjistí\_ZdaChybiPredmet()

```
bool kontroly.Zjistí_ZdaChybiPredmet (
    list[str] seznam_predmetu,
    list<kontrolovany_seznam>,
    int radek,
    int index)

Zjistí, zda v seznamu předmět chybí.
:param seznam_predmetu: seznam předmětů
:param kontrolovany_seznam: seznam, ve kterém mám předmět hledat
:param radek: první index seznamu
:param index: index hodnoty
:return: chybí / nechybí
```

## 4.6 Dokumentace jmenného prostoru LP2

### Funkce

- [Vytvor\\_Sloty](#) (seznam\_casy, pocet\_radku)
- [LP](#) (pppdo, pppdv, vyucujici\_casy, obory\_casy, mistnosti\_casy, seznam\_oboru, seznam\_predmetu, seznam\_vyucujicich)

### 4.6.1 Dokumentace funkcí

#### 4.6.1.1 LP()

```
LP2.LP (
    pppdo,
    pppdv,
    vyucujici_casy,
    obory_casy,
    mistnosti_casy,
    seznam_oboru,
    seznam_predmetu,
    seznam_vyucujicich)
```

#### 4.6.1.2 Vytvor\_Sloty()

```
LP2.Vytvor_Sloty (
    seznam_casy,
    pocet_radku)
```



## 4.7 Dokumentace jmenného prostoru main

### Proměnné

- `start` = `time.time()`
- `list_dny` = `["Po", "Ut", "St", "Ct", "Pa"]`
- `list_delka_Predmetu` = `[2, 3, 4]`
- `seznam_obor_predmet` = `Nacti_CSV(os.path.join('csv', 'obor_predmet.csv'))`
- `seznam_predmet_mistnost_poradi` = `Nacti_CSV(os.path.join('csv', 'predmet_mistnost_poradi.csv'))`
- `seznam_predmet_delkaPredmetu` = `Nacti_CSV(os.path.join('csv', 'predmet_delkaPredmetu.csv'))`
- `seznam_vyucujici_predmet_pocetPrednasek` = `Nacti_CSV(os.path.join('csv', 'vyucujici_predmet_pocetPrednasek.csv'))`
- `data_o_jiz_zapsanych_p` = `Nacti_CSV(os.path.join('csv', 'predmety_napevno.csv'))`
- `seznam_vyucujici_casy` = `Nacti_JSON(os.path.join('json', 'vyucujici_casy.json'))`
- `seznam_mistnosti_casy` = `Nacti_JSON(os.path.join('json', 'mistnosti_casy.json'))`
- `seznam_oboru` = `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Oboru(seznam_obor_predmet)`
- `seznam_predmetu` = `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Predmetu(seznam_obor_predmet)`
- `seznam_vyucujicich` = `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Vyuc(seznam_vyucujici_predmet_pocetPrednasek)`
- `seznam_mistnosti` = `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Mistnosti(seznam_predmet_mistnost_poradi)`
- `pocet_potrebných_prednasek_dle_oboru` = `tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDO(seznam_obor_predmet, seznam_oboru, seznam_predmetu)`
- `pocet_potrebných_prednasek_dle_vyucujiciho`
- `preference_mistnosti` = `tvorba_seznamu_z_nactenych_dat.Vytvor_PrefMist(seznam_predmet_mistnost_poradi, seznam_mistnosti, seznam_predmetu)`
- `seznam_delkaPredmetu_predmet` = `tvorba_seznamu_z_nactenych_dat.Vytvor_DP(seznam_predmet_delkaPredmetu, seznam_predmetu)`
- `list_upozorneni` = `[]`
- `vyucujici_casy` = `tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_vyucujici_casy, seznam_vyucujicich, upozorneni, 0)`
- `mistnosti_casy` = `tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_mistnosti_casy, seznam_mistnosti, upozorneni, 1)`
- `obory_casy` = `tvorba_seznamu_z_nactenych_dat.Vytvor_Tabulku_Obsazenosti(seznam_oboru)`
- `chyby`
- `seznam_poradi_predmetu` = `pomocne_funkce.Odstran_OpakPrvkySeznamu(seznam_poradi_predmetu)`
- `seznam_vlivu`
- `seznam_hodnoceni`
- `sablona_rozvrhu`
- `list_pocet_prepsani` = `[3, int((len(seznam_predmetu) - (len(seznam_predmetu) % 2))/2)]`
- `list_vytvorene_rozvrhy` = `[]`
- `list_seznam_vlivy_vytv_roz` = `[]`
- `list_seznam_vliv_predmety_hodnoceni_nezapPred` = `[]`
- `list_posl` = `[]`
- `list_vlivy_posl` = `[]`
- `list_hodnoceni_posl` = `[]`
- `rozvrhy` = `pomocne_funkce.Odstran_OpakujiciSeRozvrhy(vytvorene_rozvrhy)`
- `konec` = `time.time()`
- `cas` = `round(konec - start, 2)`
- `str_cesta_k_pom_excel` = `"xlsx/vlivy_predmety_hodnoty_nezapPred.xlsx"`

### 4.7.1 Dokumentace proměnných

#### 4.7.1.1 cas

```
main.cas = round(konec - start, 2)
```

#### 4.7.1.2 cesta\_k\_pom\_excel

```
str main.cesta_k_pom_excel = "xlsx/vlivy_predmety_hodnoty_nezapPred.xlsx"
```

#### 4.7.1.3 chyby

```
main.chyby
```

##### Inicializační hodnota:

```
00001 = kontroly.Kontrola_Vst_Dat(seznam_predmetu,  
00002                                     seznam_predmet_delkaPredmetu,  
00003                                     seznam_predmet_mistnost_poradi,  
00004                                     seznam_vyucujici_predmet_pocetPrednasek,  
00005                                     data_o_jiz_zapsanych_p,  
00006                                     delka_Predmetu,  
00007                                     seznam_vyucujicich,  
00008                                     seznam_mistnosti,  
00009                                     vyucujici_casy,  
00010                                     mistnosti_casy,  
00011                                     pocet_potrebnych_prednasek_dle_vyucujiciho,  
00012                                     seznam_delkaPredmetu_predmet)
```

#### 4.7.1.4 data\_o\_jiz\_zapsanych\_p

```
main.data_o_jiz_zapsanych_p = Nacti_CSV(os.path.join('csv', 'predmety_napevno.csv'))
```

#### 4.7.1.5 delka\_Predmetu

```
list main.delka_Predmetu = [2, 3, 4]
```

#### 4.7.1.6 dny

```
list main.dny = ["Po", "Ut", "St", "Ct", "Pa"]
```

#### 4.7.1.7 hodnoceni\_posl

```
main.hodnoceni_posl = []
```

#### 4.7.1.8 konec

```
main.konec = time.time()
```

#### 4.7.1.9 mistnosti\_casy

```
main.mistnosti_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_mistnosti_casy, seznam_mistnosti,  
upozorneni, 1)
```

#### 4.7.1.10 obory\_casy

```
main.obory_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Tabulku_Obsazenosti(seznam_oboru)
```

#### 4.7.1.11 pocet\_potrebných\_prednasek\_dle\_oboru

```
main.pocet_potrebných_prednasek_dle_oboru = tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDO(seznam_obor_predmet,
seznam_oboru, seznam_predmetu)
```

#### 4.7.1.12 pocet\_potrebných\_prednasek\_dle\_vyucujiciho

```
main.pocet_potrebných_prednasek_dle_vyucujiciho
```

##### Inicializační hodnota:

```
00001 = tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDV(seznam_vyucujici_predmet_pocetPrednasek,
00002                                             seznam_vyucujicich,
00003                                             seznam_predmetu)
```

#### 4.7.1.13 pocet\_prepsani

```
list main.pocet_prepsani = [3, int((len(seznam_predmetu) - (len(seznam_predmetu) % 2))/2)]
```

#### 4.7.1.14 posl

```
main.posl = []
```

#### 4.7.1.15 preference\_mistnosti

```
main.preference_mistnosti = tvorba_seznamu_z_nactenych_dat.Vytvor_PrefMist(seznam_predmet_mistnost_poradi,
seznam_mistnosti, seznam_predmetu)
```

#### 4.7.1.16 rozvrhy

```
main.rozvrhy = pomocne_funkce.Odstran_OpakujiciSeRozvrhy(vytvorene_rozvrhy)
```

#### 4.7.1.17 sablona\_rozvrhu

```
main.sablona_rozvrhu
```

##### Inicializační hodnota:

```
00001 = tvorba_rozvrhu.Vytvor_SablonuRozvrhu(len(seznam_oboru),
00002                                             pocet_potrebných_prednasek_dle_oboru,
00003                                             pocet_potrebných_prednasek_dle_vyucujiciho,
00004                                             obory_casy,
00005                                             vyucujici_casy,
00006                                             mistnosti_casy,
00007                                             seznam_predmetu,
00008                                             seznam_vyucujicich,
00009                                             seznam_mistnosti,
00010                                             data_o_jiz_zapsanych_p,
00011                                             seznam_oboru)
```

#### 4.7.1.18 seznam\_delkaPredmetu\_predmet

```
main.seznam_delkaPredmetu_predmet = tvorba_seznamu_z_nactenych_dat.Vytvor_DP(seznam_predmet_delkaPredmetu, seznam_predmetu)
```

#### 4.7.1.19 seznam\_hodnoceni

```
main.seznam_hodnoceni
```

#### 4.7.1.20 seznam\_mistnosti

```
main.seznam_mistnosti = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Mistnosti(seznam_predmet_mistnost_poradi)
```

#### 4.7.1.21 seznam\_mistnosti\_casy

```
main.seznam_mistnosti_casy = Nacti_JSON(os.path.join('json', 'mistnosti_casy.json'))
```

#### 4.7.1.22 seznam\_obor\_predmet

```
main.seznam_obor_predmet = Nacti_CSV(os.path.join('csv', 'obor_predmet.csv'))
```

#### 4.7.1.23 seznam\_oboru

```
main.seznam_oboru = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Oboru(seznam_obor_predmet)
```

#### 4.7.1.24 seznam\_poradi\_predmetu

```
main.seznam_poradi_predmetu = pomocne_funkce.Odstran_OpakPrvkySeznamu(seznam_poradi_predmetu)
```

#### 4.7.1.25 seznam\_predmet\_delkaPredmetu

```
main.seznam_predmet_delkaPredmetu = Nacti_CSV(os.path.join('csv', 'predmet_delkaPredmetu.↵csv'))
```

#### 4.7.1.26 seznam\_predmet\_mistnost\_poradi

```
main.seznam_predmet_mistnost_poradi = Nacti_CSV(os.path.join('csv', 'predmet_mistnost_poradi.↵csv'))
```

#### 4.7.1.27 seznam\_predmetu

```
main.seznam_predmetu = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Predmetu(seznam_obor_predmet)
```

#### 4.7.1.28 seznam\_vliv\_predmety\_hodnoceni\_nezapPred

```
list main.seznam_vliv_predmety_hodnoceni_nezapPred = []
```

#### 4.7.1.29 seznam\_vlivu

```
main.seznam_vlivu
```

#### 4.7.1.30 seznam\_vlivy\_vytv\_roz

```
list main.seznam_vlivy_vytv_roz = []
```

#### 4.7.1.31 seznam\_vyucujici\_casy

```
main.seznam_vyucujici_casy = Nacti_JSON(os.path.join('json', 'vyucujici_casy.json'))
```

#### 4.7.1.32 seznam\_vyucujici\_predmet\_pocetPrednasek

```
main.seznam_vyucujici_predmet_pocetPrednasek = Nacti_CSV(os.path.join('csv', 'vyucujici_↵  
predmet_pocetPrednasek.csv'))
```

#### 4.7.1.33 seznam\_vyucujicich

```
main.seznam_vyucujicich = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Vyuc(seznam_vyucujici_predmet_pocetPre
```

#### 4.7.1.34 start

```
main.start = time.time()
```

#### 4.7.1.35 upozorneni

```
list main.upozorneni = []
```

#### 4.7.1.36 vlivy\_posl

```
main.vlivy_posl = []
```

#### 4.7.1.37 vytvorene\_rozvrhy

```
main.vytvorene_rozvrhy = []
```

#### 4.7.1.38 vyucujici\_casy

```
main.vyucujici_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_vyucujici_casy, seznam_vyucujicich, upozorneni, 0)
```

## 4.8 Dokumentace jmenného prostoru nacti\_data

### Funkce

- list [Nacti\\_JSON](#) (str cesta)
- list [Nacti\\_CSV](#) (str cesta)

### 4.8.1 Dokumentace funkcí

#### 4.8.1.1 Nacti\_CSV()

```
list nacti_data.Nacti_CSV (
    str cesta)
```

Načítá csv soubory.  
:param cesta: cesta k souboru  
:return: seznam načtených dat jako str

#### 4.8.1.2 Nacti\_JSON()

```
list nacti_data.Nacti_JSON (
    str cesta)
```

Načítá json soubory.  
:param cesta: cesta k souboru  
:return: seznam načtených dat jako str

## 4.9 Dokumentace jmenného prostoru pomocne\_funkce

### Funkce

- int [Zjisti\\_PocetJednicekVyuc](#) (int vyuc, list[list[list[int]]] vyucujici\_casy)
- list[list[int]] [Najdi\\_PredmetyVyucujiciho](#) (list[list[int]] pppdv, int vyuc, list[int] seznam\_delkaPredmetu\_predmet, list[str] seznam\_vyuc)
- list[int] [Najdi\\_Mistnosti\\_Predmetu](#) (list[list[int]] pref\_mistnosti, int predmet)
- list[int] [Najdi\\_Cas\\_Vyucujiciho](#) (list[list[list[int]]] casy\_vyuc, int vyuc, int delka\_prednasky, list[list[int]] jiz\_vybrane)
- list[int] [Najdi\\_Vyucujici\\_Predmetu](#) (list[list[int]] pppdv, int predmet)
- bool [Zjisti\\_Je\\_Mistnost\\_Volna](#) (list[list[list[int]]] obs\_mist, int mistnost, list[int] den\_odH, int delka\_prednasky)
- bool [Zjisti\\_Ma\\_Vyucujici\\_Volno](#) (list[list[list[int]]] obs\_vyuc, int vyuc, int den, int cas, int delka\_prednasky)
- None [Vypis\\_Vlivy](#) (list[list[int]] vlivy)
- int [Zjisti\\_CisloPomociKodu](#) (list[str] seznam, str kod)
- int [Zjisti\\_Pocet\\_Volnych\\_Hodin](#) (list[list[list[int]]] vyucujici\_casy, list[int] seznam\_vyucujicich\_pro\_tento\_predmet)
- int [Zjisti\\_Pocet\\_Mistnosti](#) (list[list[int]] preference\_mistnosti, int predmet)
- list[int] [Zjisti\\_Obory\\_Predmetu](#) (list[list[int]] pppdo, int predmet)
- bool [Zjisti\\_ZdaPredmetJeNapevno](#) (list[list[str]] data\_o\_jiz\_zapsanych\_p, list[str] seznam\_predmetu, int predmet)
- bool [Zjisti\\_JeVolnoVRozvrhu](#) (rozvrh\_type rozvrh, int den, int cas, int delka\_prednasky, list[int] obory\_predmetu)
- list [Odstran\\_OpakPrvkySeznamu](#) (list seznam)
- list[rozvrh\_type] [Odstran\\_OpakujiciSeRozvrhy](#) (list[rozvrh\_type] rozvrhy)

**Proměnné**

- rozvrh\_type = list

**4.9.1 Dokumentace funkcí****4.9.1.1 Najdi\_Cas\_Vyucujiciho()**

```
list[int] pomocne_funkce.Najdi_Cas_Vyucujiciho (
    list[list[list[int]]] casy_vyuc,
    int vyuc,
    int delka_prednasky,
    list[list[int]] jiz_vybrane)
```

Hledá čas vyučujícího, kdy může začít vyučovat předmět.

:param casy\_vyuc: v řádcích jsou vyučující, ve sloupcích dny s hodinami, 1 -> má volno, 0 -> nemá volno

:param vyuc: index vyučujícího v seznamu vyučujících

:param delka\_prednasky: délka přednášky

:param jiz\_vybrane: seznam již vybraných časů

:return: seznam dne a hodiny, od které může vyučovat a dvojice již nebyla použita dříve

**4.9.1.2 Najdi\_Mistnosti\_Predmetu()**

```
list[int] pomocne_funkce.Najdi_Mistnosti_Predmetu (
    list[list[int]] pref_mistnosti,
    int predmet)
```

Hledá povolené místnosti předmětu.

:param pref\_mistnosti: v řádcích jsou místnosti, ve sloupcích předměty, hodnoty = pořadí

:param predmet: index předmětu v seznamu předmětů

:return: seřazený seznam použitelných místností

**4.9.1.3 Najdi\_PredmetyVyucujiciho()**

```
list[list[int]] pomocne_funkce.Najdi_PredmetyVyucujiciho (
    list[list[int]] pppdv,
    int vyuc,
    list[int] seznam_delkaPredmetu_predmet,
    list[str] seznam_vyuc)
```

Hledá indexy předmětů vyučujícího.

:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího

:param vyuc: index vyučujícího v seznamu vyučujících

:param seznam\_delkaPredmetu\_predmet: seznam délek předmětů

:param seznam\_vyuc: seznam vyučujících

:return: seznam indexů předmětů, počtů přednášek a délek

#### 4.9.1.4 Najdi\_Vyucujici\_Predmetu()

```
list[int] pomocne_funkce.Najdi_Vyucujici_Predmetu (  
    list[list[int]] pppdv,  
    int predmet)
```

Hledá vyučující předmětu.

:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího  
:param predmet: index předmětu v seznamu předmětů  
:return: indexy vyučujících co tento předmět vyučují

#### 4.9.1.5 Odstran\_OpakPrvkySeznamu()

```
list pomocne_funkce.Odstran_OpakPrvkySeznamu (  
    list seznam)
```

Odstraňuje opakující se prvky seznamu.

:param seznam: seznam  
:return:

#### 4.9.1.6 Odstran\_OpakujiciSeRozvrhy()

```
list[rozvrh_type] pomocne_funkce.Odstran_OpakujiciSeRozvrhy (  
    list[rozvrh_type] rozvrhy)
```

Odstraňuje opakující se rozvrhy v seznamu vytvořených.

:param rozvrhy: seznam vytvořených rozvrhů, kde se rozvrhy mohou opakovat  
:return: seznam jedinečných rozvrhů

#### 4.9.1.7 Vypis\_Vlivy()

```
None pomocne_funkce.Vypis_Vlivy (  
    list[list[int]] vlivy)
```

Výpis vektorů s vlivy.

:param vlivy: seznam seznamů vlivů ("seznam vektorů")  
:return: None

#### 4.9.1.8 Zjisti\_CisloPomociKodu()

```
int pomocne_funkce.Zjisti_CisloPomociKodu (  
    list[str] seznam,  
    str kod)
```

Zjistí uje index pomocí kódu.

:param seznam: seznam místností/vyučujících/oborů/předmětů  
:param kod: kód místnosti/vyučujícího/oboru/předmětu  
:return: index v seznamu m/v/o/p



#### 4.9.1.9 Zjisti\_Je\_Mistnost\_Volna()

```
bool pomocne_funkce.Zjisti_Je_Mistnost_Volna (  
    list[list[list[int]]] obs_mist,  
    int mistnost,  
    list[int] den_odH,  
    int delka_prednasky)
```

Zjišťuje, zda je místnost volná.

:param obs\_mist: v řádcích jsou místnosti, ve sloupcích dny s hodinami, 1 -> je volno, 0 -> není volno  
:param mistnost: index místnosti v seznamu místností  
:param den\_odH: seznam obsahující den a hodinu, od které má být místnost volná po dobu délky přednášky  
:param delka\_prednasky: délka přednášky  
:return: místnost je / není volná

#### 4.9.1.10 Zjisti\_JeVolnoVRozvrhu()

```
bool pomocne_funkce.Zjisti_JeVolnoVRozvrhu (  
    rozvrh_type rozvrh,  
    int den,  
    int cas,  
    int delka_prednasky,  
    list[int] obory_predmetu)
```

Zjišťuje, zda je v rozvrhu volno.

:param rozvrh: rozvrh  
:param den: index dne (0-4)  
:param cas: index hodiny (0-11)  
:param delka\_prednasky: délka přednášky  
:param obory\_predmetu: seznam indexů oborů co tento předmět mají mít v rozvrhu  
:return: zda, každý z oborů má volno, aby se předmět dal odučit pro všechny obory, pro které je předmět společný

#### 4.9.1.11 Zjisti\_Ma\_Vyucujici\_Volno()

```
bool pomocne_funkce.Zjisti_Ma_Vyucujici_Volno (  
    list[list[list[int]]] obs_vyuc,  
    int vyuc,  
    int den,  
    int cas,  
    int delka_prednasky)
```

Zjišťuje, zda vyučující má volno.

:param obs\_vyuc: v řádcích jsou vyučující, ve sloupcích dny s hodinami, 1 -> má volno, 0 -> nemá volno  
:param vyuc: index vyučujícího v seznamu vyučujících  
:param den: index dne (0-4)  
:param cas: index hodiny  
:param delka\_prednasky: délka přednášky  
:return: má vyučující volno tento den, od této hodiny po dobu délky přednášky -> bool

#### 4.9.1.12 Zjisti\_Obory\_Predmetu()

```
list[int] pomocne_funkce.Zjisti_Obory_Predmetu (  
    list[list[int]] pppdo,  
    int predmet)
```

Zjist'uje, které obory mají tento předmět.

:param pppdo: pokud je ve sloupci více nenulových čísel -> pro všechny řádky (obory) je tento sloupec (předmět)

:param predmet: index předmětu v seznamu předmětů

:return: seznam indexů oborů co tento předmět mají mít v rozvrhu

#### 4.9.1.13 Zjisti\_Pocet\_Mistnosti()

```
int pomocne_funkce.Zjisti_Pocet_Mistnosti (  
    list[list[int]] preference_mistnosti,  
    int predmet)
```

Zjist'uje počet místností pro tento předmět určených.

:param preference\_mistnosti: v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = pořadí

:param predmet: index předmětu v seznamu předmětů

:return: počet povolených místností pro tento předmět

#### 4.9.1.14 Zjisti\_Pocet\_Volnych\_Hodin()

```
int pomocne_funkce.Zjisti_Pocet_Volnych_Hodin (  
    list[list[list[int]] vyucujici_casy,  
    list[int] seznam_vyucujicich_pro_tento_predmet)
```

Zjist'uje počet volných hodin vyučujících zvoleného předmětu.

:param vyucujici\_casy: data volných hodin vyučujících

:param seznam\_vyucujicich\_pro\_tento\_predmet: seznam vyučujících co tento předmět vyučují

:return: počet volných hodin vyučujících tohoto předmětu

#### 4.9.1.15 Zjisti\_PocetJednicekVyuc()

```
int pomocne_funkce.Zjisti_PocetJednicekVyuc (  
    int vyuc,  
    list[list[list[int]] vyucujici_casy)
```

Počítá počet volných hodin vyučujícího.

:param vyuc: index vyučujícího v seznamu vyučujících

:param vyucujici\_casy: v řádcích jsou vyučující, ve sloupcích dny s hodinami, 1 -> má volno, 0 -> nemá volno

:return: počet volných hodin

#### 4.9.1.16 Zjisti\_ZdaPredmetJeNapevno()

```
bool pomocne_funkce.Zjisti_ZdaPredmetJeNapevno (
    list[list[str]] data_o_jiz_zapsanych_p,
    list[str] seznam_predmetu,
    int predmet)
```

Zjistí, zda je tento předmět zapsán do šablony rozvrhů.

:param data\_o\_jiz\_zapsanych\_p: seznam seznamů dat o předem zapsaných předmětech do rozvrhu

:param seznam\_predmetu: seznam předmětů

:param predmet: index předmětu v seznamu předmětů

:return: předmět je/není zapsán předem

## 4.9.2 Dokumentace proměnných

### 4.9.2.1 rozvrh\_type

```
pomocne_funkce.rozvrh_type = list
```

## 4.10 Dokumentace jmenného prostoru tvorba\_rozvrhu

### Funkce

- [TvorRozvrhyProRuzneVlivy](#) (list[list[int]] seznam\_poradi\_predmetu, list[list[int]] seznam\_vlivu, list[list[int]] pocet\_potrebnych\_prednasek\_dle\_oboru, list[list[int]] pocet\_potrebnych\_prednasek\_dle\_vyucujiciho, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] obory\_casy, list[list[list[int]]] mistnosti\_casy, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[str] seznam\_predmetu, [rozvrh\\_type](#) sablona\_rozvrhu, list[list[float]] seznam\_hodnoceni, list[str] seznam\_oboru, list[str] seznam\_mistnosti, list[str] seznam\_vyucujicich, list[list[str]] data\_o\_jiz\_zapsanych\_p, int pocet\_prepsani)
- [Tvor\\_RozvrhyUpravouPosloupnosti](#) (posl\_predmetu, vlivy\_posl, list[list[int]] pocet\_potrebnych\_prednasek\_dle\_oboru, list[list[int]] pocet\_potrebnych\_prednasek\_dle\_vyucujiciho, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] obory\_casy, list[list[list[int]]] mistnosti\_casy, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[str] seznam\_predmetu, [rozvrh\\_type](#) sablona\_rozvrhu, hodnoceni\_posl, list[str] seznam\_oboru, list[str] seznam\_mistnosti, list[str] seznam\_vyucujicich)
- list[int] [Uprav\\_Posloupnost](#) (list[int] posloupnost, int nezap\_predmet, list[list[int]] pouzite\_posl, int pocet\_predmetu)
- [rozvrh\\_type Proved\\_PridejDoSablonyRozvrhu](#) ([rozvrh\\_type](#) rozvrh, list[int] data\_k\_presunuti, list[list[list[int]]] obs\_vyuc, list[list[list[int]]] obs\_mist, list[list[list[int]]] obs\_oboru, list[list[int]] pppdv, list[list[int]] pppdo, list[str] seznam\_vyuc)
- [list[[rozvrh\\_type](#)], list[list[int]], list[list[int]], list[list[list[int]]], list[list[list[int]]], list[list[list[int]]]] [Vytvor\\_Rozvrh\\_dlePredmetu](#) (list[int] posloupnost\_predmetu, list[list[int]] pom\_pppdo, list[list[int]] pom\_pppdv, list[list[list[int]]] pom\_obs\_vyuc, list[list[list[int]]] pom\_obs\_oboru, list[list[list[int]]] pom\_obs\_mist, list[int] delky\_prednasek, list[list[int]] prefer\_mist, [rozvrh\\_type](#) sablona\_rozvrhu, list[str] seznam\_oboru, list[str] seznam\_predmetu, list[str] seznam\_mistnosti, list[str] seznam\_vyucujicich)
- list[int] [Zjisti\\_MohouLiSePremistitPredmety](#) ([rozvrh\\_type](#) rozvrh, list[list[list[int]]] obs\_vyuc, list[list[list[int]]] obs\_oboru, list[list[list[int]]] obs\_mist, list[list[int]] nezapsane\_predmety, list[list[int]] pppdv, list[int] delky\_predmetu, list[list[int]] pppdo, list[list[int]] prefer\_mist, list[list[list[int]]] vyucujici\_casy, list[str] seznam\_predmetu, list[str] seznam\_vyucujicich, list[str] seznam\_mistnosti, list[list[str]] data\_o\_jiz\_zapsanych\_p)
- None [Obsad](#) (list[list[list[int]]] casy, [int, int] den\_odH, int delka\_prednasky, list[int] kdo\_se\_ma\_obsadit)
- None [Zapis\\_Do\\_Rozvrhu](#) (list[int] seznam\_oboru, [int, int] den\_odH, int delka\_prednasky, int vyuc, int mistnost, [rozvrh\\_type](#) rozvrh, int predmet)
- None [Odecti\\_Pocet\\_Prednasek](#) (list[list[int]] pppd, list[int] seznam\_oboru\_nebo\_vyuc, int predmet)
- bool [Zjisti\\_Maji\\_Obory\\_Volno](#) (list[list[list[int]]] obs\_oboru, list[int] obory, [int, int] den\_odH, int delka\_prenasky)
- list[list[int]] [Najdi\\_VyucBlokujiciNezapPredmet](#) ([rozvrh\\_type](#) rozvrh, list[list[list[int]]] vyucujici\_casy, list[int] vyuc\_nezap\_predmetu, list[list[list[int]]] obs\_oboru)
- [rozvrh\\_type Vytvor\\_SablonyRozvrhu](#) (int pocet\_oboru, list[list[int]] pppdo, list[list[int]] pppdv, list[list[list[int]]] obs\_oboru, list[list[list[int]]] obs\_vyuc, list[list[list[int]]] obs\_mist, list[str] seznam\_predmetu, list[str] seznam\_vyuc, list[str] seznam\_mistnosti, list[list[str]] data\_o\_jiz\_zapsanych\_p, list[str] seznam\_oboru)

## Proměnné

- `rozvrh_type` = list

## 4.10.1 Dokumentace funkcí

### 4.10.1.1 Najdi\_VyucBlokujiciNezapPredmet()

```
list[list[int]] tvorba_rozvrhu.Najdi_VyucBlokujiciNezapPredmet (
    rozvrh_type rozvrh,
    list[list[list[int]]] vyucujici_casy,
    list[int] vyuc_nezap_predmetu,
    list[list[list[int]]] obs_oboru)
```

Hledá vyučující, co blokovaly čas nezapsanému předmětu.  
:param rozvrh: rozvrh, kde se nějaký předmět nezapsal  
:param vyucujici\_casy: data vole vyučujících (obsazenost)  
:param vyuc\_nezap\_predmetu: seznam vyučujících předmětu, který se nezapsal  
:param obs\_oboru: data volen oborů (obsazenost)  
:return: seznam dvojic vyučující-předmět co blokovaly časy (všechny) vyučujících blokovaného předmětu

### 4.10.1.2 Obsad()

```
None tvorba_rozvrhu.Obsad (
    list[list[list[int]]] casy,
    [int, int] den_odH,
    int delka_prednasky,
    list[int] kdo_se_ma_obsadit)
```

Mění volné hodiny na obsazené.  
:param casy: data volen vyučujících/místností/oborů  
:param den\_odH: [den, hodina]. (0 hodina - od 8h do 9h, 1 hodina - od 9h do 10h)  
:param delka\_prednasky: délka přednášky  
:param kdo\_se\_ma\_obsadit: seznam indexů vyučujících/místností/oborů  
:return: None

### 4.10.1.3 Odecti\_Pocet\_Prednasek()

```
None tvorba_rozvrhu.Odecti_Pocet_Prednasek (
    list[list[int]] pppd_,
    list[int] seznam_oboru_nebo_vuyc,
    int predmet)
```

Odečítá počet přednášek oborům nebo vyučujícímu  
:param pppd\_: (pocet\_potrebnych\_prednasek\_dle\_oboru/pocet\_potrebnych\_prednasek\_dle\_vyucujiciho)  
:param seznam\_oboru\_nebo\_vuyc: seznam indexů oborů/vyučujících  
:param predmet: index předmětu  
:return: None

#### 4.10.1.4 Proved\_PridejDoSablonyRozvrhu()

```
rozvrh_type tvorba_rozvrhu.Proved_PridejDoSablonyRozvrhu (
    rozvrh_type rozvrh,
    list[int] data_k_presunutí,
    list[list[list[int]]] obs_vyuc,
    list[list[list[int]]] obs_mist,
    list[list[list[int]]] obs_oboru,
    list[list[int]] pppdv,
    list[list[int]] pppdo,
    list[str] seznam_vyuc)
```

Přidává předmět do šablony rozvrhu.

:param rozvrh: šablona rozvrhu

:param data\_k\_presunutí: [index\_předmětu: int, index\_vyučujícího: int, index\_místnosti:int , den\_od\_1: int, ca

:return: šablona s přidáním předmětem

#### 4.10.1.5 Tvor\_RozvrhyUpravouPosloupnosti()

```
tvorba_rozvrhu.Tvor_RozvrhyUpravouPosloupnosti (
    posl_predmetu,
    vlivy_posl,
    list[list[int]] pocet_potrebných_prednasek_dle_oboru,
    list[list[int]] pocet_potrebných_prednasek_dle_vyucujícího,
    list[list[list[int]]] vyucující_casy,
    list[list[list[int]]] obory_casy,
    list[list[list[int]]] místnosti_casy,
    list[int] seznam_delkaPredmetu_predmet,
    list[list[int]] preference_místnosti,
    list[str] seznam_predmetu,
    rozvrh_type sablona_rozvrhu,
    hodnoceni_posl,
    list[str] seznam_oboru,
    list[str] seznam_místnosti,
    list[str] seznam_vyucujících)
```

Přesouvá předměty dopředu v posloupnosti předmětů, pokud se nezapiší. Pokud takováto posloupnost už byla testována (např. se předmět opět nezapsal), tak se předmět posouvá dopředu o 2 pozice, atd. Tento proces se opakuje dokud:

se všechny předměty nezapiší

posloupnost už byla testována a předmět už nejde dále posunout

:param posl\_predmetu: seznam seřazených předmětů podle důležitosti

:param vlivy\_posl: vlivy použité na vytvoření dané posloupnosti

:param pocet\_potrebných\_prednasek\_dle\_oboru: data předmětů oborů

:param pocet\_potrebných\_prednasek\_dle\_vyucujícího: data přednášek vyučujících

:param vyucující\_casy: data volných hodin vyučujících

:param obory\_casy: data volných hodin oborů

:param místnosti\_casy: data volných hodin místností

:param seznam\_delkaPredmetu\_predmet: seznam délek předmětů

:param preference\_místnosti: v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = pořadí

:param seznam\_predmetu: seznam předmětů

:param sablona\_rozvrhu: šablona rozvrhu

:param hodnoceni\_posl: seznam hodnocení předmětů posloupnosti

:param seznam\_oboru: seznam oborů

:param seznam\_místnosti: seznam místností

:param seznam\_vyucujících: seznam vyučujících

:return: vytvořené rozvrhy

#### 4.10.1.6 TvorRozvrhyProRuzneVlivy()

```

tvorba_rozvrhu.TvorRozvrhyProRuzneVlivy (
    list[list[int]] seznam_poradi_predmetu,
    list[list[int]] seznam_vlivu,
    list[list[int]] pocet_potrebnych_prednasek_dle_oboru,
    list[list[int]] pocet_potrebnych_prednasek_dle_vyucujiciho,
    list[list[list[int]]] vyucujici_casy,
    list[list[list[int]]] obory_casy,
    list[list[list[int]]] mistnosti_casy,
    list[int] seznam_delkaPredmetu_predmet,
    list[list[int]] preference_mistnosti,
    list[str] seznam_predmetu,
    rozvrh_type sablona_rozvrhu,
    list[list[float]] seznam_hodnoceni,
    list[str] seznam_oboru,
    list[str] seznam_mistnosti,
    list[str] seznam_vyucujicich,
    list[list[str]] data_o_jiz_zapsanych_p,
    int pocet_prepsani)

```

Hlavní funkce pro tvorbu rozvrhů procházením posloupností předmětů.

```

:param seznam_poradi_predmetu: seznam seznamů pořadí předmětů
:param seznam_vlivu: seznam "vektorů"
:param pocet_potrebnych_prednasek_dle_oboru: pokud je ve sloupci více nenulových čísel -> pro všechny řádky (obor)
:param pocet_potrebnych_prednasek_dle_vyucujiciho: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty
:param vyucujici_casy: data volných hodin vyučujících
:param obory_casy: data volných hodin oborů
:param mistnosti_casy: data volných hodin místností
:param seznam_delkaPredmetu_predmet: seznam délek předmětů (index = předmět, hodnota = délka)
:param preference_mistnosti: v řádcích jsou místnosti, ve sloupcích předměty, hodnoty = pořadí
:param seznam_predmetu: seznam předmětů
:param sablona_rozvrhu: rozvrh s natvrdo zapsanými předměty
:param seznam_hodnoceni: seznam seznamů hodnocení každého předmětu (index podseznamu = seznam s pořadovými čísly)
:param seznam_oboru: seznam oborů
:param seznam_mistnosti: seznam místností
:param seznam_vyucujicich: seznam vyučujících
:param data_o_jiz_zapsanych_p: data natvrdo zapsaných předmětů
:return: vytvorene_rozvrhy: rozvrhy s plným počtem zapsaných předmětů
        seznam_vlivu_vytvorených_rozvrhu: vektory vlivů vytvořených rozvrhů
        seznam_vliv_predmety_hodnoty_nezapsane: seznam se seznamy, které obsahují vektor vlivů, předměty v seznamu

```

#### 4.10.1.7 Uprav\_Posloupnost()

```

list[int] tvorba_rozvrhu.Uprav_Posloupnost (
    list[int] posloupnost,
    int nezap_predmet,
    list[list[int]] pouzite_posl,
    int pocet_predmetu)

```

Nezapsaný předmět se prohodí s tím co je na indexu i-1.

Pokud tato posloupnost již byla použita, tak se předmět prohodí s předmětem co je na indexu i-2 pokud to jde

```

:param posloupnost: posloupnost předmětů
:param nezap_predmet: index nezapsaného předmětu
:param pouzite_posl: seznam použitých posloupností
:param pocet_predmetu: počet předmětů
:return: upravená posloupnost předmětů

```

## 4.10.1.8 Vytvor\_Rozvrh\_dlePredmetu()

```
[list[rozvrh_type], list[list[int]], list[list[int]], list[list[list[int]]], list[list[list[int]]],
list[list[list[int]]]] tvorba_rozvrhu.Vytvor_Rozvrh_dlePredmetu (
    list[int] posloupnost_predmetu,
    list[list[int]] pom_pppdo,
    list[list[int]] pom_pppdv,
    list[list[list[int]]] pom_obs_vyuc,
    list[list[list[int]]] pom_obs_oboru,
    list[list[list[int]]] pom_obs_mist,
    list[int] delky_prednasek,
    list[list[int]] prefer_mist,
    rozvrh_type sablona_rozvrhu,
    list[str] seznam_oboru,
    list[str] seznam_predmetu,
    list[str] seznam_mistnosti,
    list[str] seznam_vyucujicich)
```

Tvoří rozvrhy pomocí posloupnosti předmětů.

```
:param posloupnost_predmetu: seznam pořadí předmětů (předměty jsou v pořadí dle seznamu předmětů, hodnoty jsou
:param pom_pppdo: pokud je ve sloupci více nenulových čísel -> pro všechny řádky(obory) je tento sloupec(předmět)
:param pom_pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího
:param pom_obs_vyuc: data volných hodin vyučujících
:param pom_obs_oboru: data volných hodin místností
:param pom_obs_mist: data volných hodin místností
:param delky_prednasek: seznam délek přednášek (index = předmět, hodnota = délka)
:param prefer_mist: v řádcích jsou místnosti, ve sloupcích předměty, hodnoty = pořadí
:param sablona_rozvrhu: rozvrh s napevno zapsanými předměty
:param seznam_oboru: seznam oborů
:param seznam_predmetu: seznam předmětů
:param seznam_mistnosti: seznam místností
:param seznam_vyucujicich: seznam vyučujících
:return: rozvrh: rozvrh s nějakým počtem zapsaných předmětů
        pppdo: parametr ovlivněný zápisem předmětů
        pppdv: parametr ovlivněný zápisem předmětů
        obs_vyuc: parametr ovlivněný zápisem předmětů
        obs_oboru: parametr ovlivněný zápisem předmětů
        obs_mist: parametr ovlivněný zápisem předmětů
```

## 4.10.1.9 Vytvor\_SablонуRozvrhu()

```
rozvrh_type tvorba_rozvrhu.Vytvor_SablónuRozvrhu (
    int pocet_oboru,
    list[list[int]] pppdo,
    list[list[int]] pppdv,
    list[list[list[int]]] obs_oboru,
    list[list[list[int]]] obs_vyuc,
    list[list[list[int]]] obs_mist,
    list[str] seznam_predmetu,
    list[str] seznam_vyuc,
    list[str] seznam_mistnosti,
    list[list[str]] data_o_jiz_zapsanych_p,
    list[str] seznam_oboru)
```

Vytváří šablonu rozvrhů.

```
:param pocet_oboru: počet oborů
:param pppdo: pokud je ve sloupci více nenulových čísel -> pro všechny řádky(obory) je tento sloupec(předmět)
:param pppdv: v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyučujícího
:param obs_oboru: data volen oborů (obsazenost oborů)
```

```
:param obs_vyuc: data volen vyučujících (obsazenost vyučujících)
:param obs_mist: data volen místností (obsazenost místností)
:param seznam_predmetu: seznam předmětů
:param seznam_vyuc: seznam vyučujících
:param seznam_mistnosti: seznam místností
:param data_o_jiz_zapsanych_p: seznam s daty napevno zapsaných předmětů
:param seznam_oboru: seznam
:return: šablona rozvrhu
```

#### 4.10.1.10 Zapis\_Do\_Rozvrhu()

```
None tvorba_rozvrhu.Zapis_Do_Rozvrhu (
    list[int] seznam_oboru,
    [int, int] den_odH,
    int delka_prednasky,
    int vyuc,
    int mistnost,
    rozvrh_type rozvrh,
    int predmet)
```

Přidává určitým oborům do určitých časových slotů v rozvrhu seznam obsahující vyučujícího, předmět, místnost.

```
:param seznam_oboru: seznam indexů oborů
:param den_odH: [den, hodina], hodina = číslo slotu (0 hodina = od 8h do 9h)
:param delka_prednasky: délka přednášky
:param vyuc: index vyučujícího
:param mistnost: index místnosti
:param rozvrh: rozvrh kam se to mají zapsat data
:param predmet: index předmětu
:return: None
```

#### 4.10.1.11 Zjisti\_Maji\_Obory\_Volno()

```
bool tvorba_rozvrhu.Zjisti_Maji_Obory_Volno (
    list[list[list[int]]] obs_oboru,
    list[int] obory,
    [int, int] den_odH,
    int delka_prenasky)
```

Zjistí, zda mají obory volno po dobu délky přednášky.

```
:param obs_oboru: data volen oborů (obsazenost)
:param obory: seznam indexů oborů
:param den_odH: [den, hodina] (0 hodina = od 8h do 9h)
:param delka_prenasky: délka přednášky
:return: obory mají/nemají volno po dobu délky přednášky
```

#### 4.10.1.12 Zjisti\_MohouLiSePremistitPredmety()

```
list[int] tvorba_rozvrhu.Zjisti_MohouLiSePremistitPredmety (
    rozvrh_type rozvrh,
    list[list[list[int]]] obs_vyuc,
    list[list[list[int]]] obs_oboru,
    list[list[list[int]]] obs_mist,
    list[list[int]] nezapsane_predmety,
    list[list[int]] pppdv,
```



```
list[int] delky_predmetu,
list[list[int]] pppdo,
list[list[int]] prefer_mist,
list[list[list[int]]] vyucujici_casy,
list[str] seznam_predmetu,
list[str] seznam_vyucujicich,
list[str] seznam_mistnosti,
list[list[str]] data_o_jiz_zapsanych_p)
```

Zjist'uje, zda existuje způsob přepsat zapsaný předmět na jiné místo, kdy po nevytvoření rozvrhu zbylo místo. Beru předmět, co se nezapsal, a zjist'uji, zda ten předmět, co ho vyblokoval, mohl být zapsán jinak.

```
:param rozvrh: rozvrh
:param obs_vyuc: ovlivněná data volných hodin vyučujících
:param obs_oboru: ovlivněná data volných hodin oborů
:param obs_mist: ovlivněná data volných hodin místností
:param nezapsane_predmety: seznam nezapsaných předmětů
:param pppdv: data přednášek vyučujících
:param delky_predmetu: seznam délek předmětů
:param pppdo: data předmětů oborů
:param prefer_mist: v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = pořadí
:param vyucujici_casy: neovlivněná data volných hodin vyučujících
:param seznam_predmetu: seznam předmětů
:param seznam_vyucujicich: seznam vyučujících
:param seznam_mistnosti: seznam místností
:param data_o_jiz_zapsanych_p: data natvrdo zapsaných předmětů
:return: data předmětu, co se запиše do šablony rozvrhů
```

## 4.10.2 Dokumentace proměnných

### 4.10.2.1 rozvrh\_type

```
tvorba_rozvrhu.rozvrh_type = list
```

## 4.11 Dokumentace jmenného prostoru tvorba\_seznamu\_z\_nactenych\_dat

### Funkce

- [Priprav\\_Seznam\\_Vyuc](#) (seznam)
- [Priprav\\_Seznam\\_Oboru](#) (seznam)
- [Priprav\\_Seznam\\_Mistnosti](#) (seznam)
- [Priprav\\_Seznam\\_Predmetu](#) (seznam)
- [Vytvor\\_Matici](#) (radky, sloupce)
- [Vytvor\\_DP](#) (seznam\_predmet\_delkaPredmetu, predmety)
- [list\[list\[int\]\] Vytvor\\_PPPDO](#) (list[list[str]] seznam, list[str] obory, list[str] predmety)
- [list\[list\[int\]\] Vytvor\\_PPPDV](#) (list[list[str]] seznam, list[str] vyucujici, list[str] predmety)
- [list\[list\[int\]\] Vytvor\\_PrefMist](#) (list[list[str]] seznam, list[str] mistnosti, list[str] predmety)
- [list\[list\[list\[int\]\]\] Vytvor\\_Casy](#) (list seznam, list[str] pom, list[str] upozorneni, int index\_seznamu)
- [list\[list\[list\[int\]\]\] Vytvor\\_Tabulku\\_Obsazenosti](#) (obory)
- [rozvrh\\_type Vytvor\\_Prazdny\\_Rozvrh](#) (int pocet\_oboru)

### Proměnné

- [rozvrh\\_type](#) = list

### 4.11.1 Dokumentace funkcí

#### 4.11.1.1 `Priprav_Seznam_Mistnosti()`

```
tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Mistnosti (  
    seznam)
```

#### 4.11.1.2 `Priprav_Seznam_Oboru()`

```
tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Oboru (  
    seznam)
```

#### 4.11.1.3 `Priprav_Seznam_Predmetu()`

```
tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Predmetu (  
    seznam)
```

#### 4.11.1.4 `Priprav_Seznam_Vyuc()`

```
tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Vyuc (  
    seznam)
```

#### 4.11.1.5 `Vytvor_Casy()`

```
list[list[list[int]]] tvorba_seznamu_z_nactenych_dat.Vytvor_Casy (  
    list seznam,  
    list[str] pom,  
    list[str] upozorneni,  
    int index_seznamu)
```

Vytváří seznam volných hodin.

Zohledňuje také situaci, kdy v json souborech nejsou data o vyučujícím/místnosti, které jsou v seznamu vyučujících/místnosti -> je volno vždy.

:param seznam: seznam, ze kterého budu čerpat data (seznam\_mistnosti\_casy/seznam\_vyucujici\_casy)

:param pom: seznam určující pořadí (seznam\_mistnosti/seznam\_vyucujicich)

:param upozorneni: seznam, do kterého se ukládají upozornění o automatickému přidání místnosti/vyučujícího

:param index\_seznamu: index seznamu -- vyučující = 0, místnosti = 1

:return: data o dnech a hodinách, kdy je volno

#### 4.11.1.6 `Vytvor_DP()`

```
tvorba_seznamu_z_nactenych_dat.Vytvor_DP (  
    seznam_predmet_delkaPredmetu,  
    predmety)
```

#### 4.11.1.7 Vytvor\_Matici()

```

tvorba_seznamu_z_nactenych_dat.Vytvor_Matici (
    radky,
    sloupce)

```

#### 4.11.1.8 Vytvor\_PPPDO()

```

list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDO (
    list[list[str]] seznam,
    list[str] obory,
    list[str] predmety)

```

vem obor(řádek) a pro každý předmět(sloupec), zapisuj 0 (není u tohoto oboru) nebo číslo udávající počet požadovaných oborem přednášek tohoto předmětu

:param seznam: seznam\_obor\_predmet (data z csv)

:param obory: seznam oborů

:param predmety: seznam předmětů

:return: pokud je ve sloupci více nenulových čísel -> pro všechny řádky(obory) je tento sloupec(předmět) spole

#### 4.11.1.9 Vytvor\_PPPDV()

```

list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDV (
    list[list[str]] seznam,
    list[str] vyucujici,
    list[str] predmety)

```

:param seznam: seznam\_vyucujici\_predmet\_pocetPrednasek (načtená data z csv)

:param vyucujici: seznam vyučujících

:param predmety: seznam předmětů

:return: 2d pole, kde v řádcích jsou vyučující, ve sloupcích jsou předměty a hodnoty jsou počet přednášek vyuč

#### 4.11.1.10 Vytvor\_Prazdny\_Rozvrh()

```

rozvrh_type tvorba_seznamu_z_nactenych_dat.Vytvor_Prazdny_Rozvrh (
    int pocet_oboru)

```

Vytváří prázdný rozvrh.

:param pocet\_oboru: počet oborů

:return: prázdná šablona rozvrhu

#### 4.11.1.11 Vytvor\_PrefMist()

```

list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PrefMist (
    list[list[str]] seznam,
    list[str] mistnosti,
    list[str] predmety)

```

Vytváří seznam povolených místností pro předměty.

:param seznam: seznam\_predmet\_mistnost\_poradi (data načtená z csv)

:param mistnosti:

:param predmety:

:return: 2d pole, kde v řádkách jsou místnosti, ve sloupcích předměty, hodnoty = pořadí

#### 4.11.1.12 Vytvor\_Tabulku\_Obsazenosti()

```
list[list[list[int]]] tvorba_seznamu_z_nactenych_dat.Vytvor_Tabulku_Obsazenosti (
    obory)
```

Vytváří prázdný seznam volných hodin.

:param obory: seznam oborů

:return: v řádcích jsou obory, ve sloupcích dny s hodinami, obs[obor][den][hodina] == 1 -> má volno, == 0 -> n

### 4.11.2 Dokumentace proměnných

#### 4.11.2.1 rozvrh\_type

```
tvorba_seznamu_z_nactenych_dat.rozvrh_type = list
```

## 4.12 Dokumentace jmenného prostoru vypis\_pomocnych\_udaju

### Funkce

- [Vypis\\_PomUdajeDoExcelu](#) (str cesta, pom\_seznam, seznam\_predmetu)

#### 4.12.1 Dokumentace funkcí

##### 4.12.1.1 Vypis\_PomUdajeDoExcelu()

```
vypis_pomocnych_udaju.Vypis_PomUdajeDoExcelu (
    str cesta,
    pom_seznam,
    seznam_predmetu)
```

Vypisuje pomocná data do EXCELu.

:param cesta: cesta k souboru

:param pom\_seznam: seznam pomocných dat

:param seznam\_predmetu: seznam předmětů pro převod indexů zpět na názvy

:return: None

## 4.13 Dokumentace jmenného prostoru vypis\_vytvorených\_rozvrhu

### Funkce

- None [Vypis\\_Rozvrh](#) (list[rozvrh\_type] rozvrhy, list[str] seznam\_oboru, list[str] seznam\_predmetu, list[str] seznam\_mistnosti, list[str] seznam\_vyucujících)

### Proměnné

- [rozvrh\\_type](#) = list

## 4.13.1 Dokumentace funkcí

### 4.13.1.1 Vypis\_Rozvrh()

```
None vypis_vytvorených_rozvrhu.Vypis_Rozvrh (  
    list[rozvrh_type] rozvrhy,  
    list[str] seznam_oboru,  
    list[str] seznam_predmetu,  
    list[str] seznam_mistnosti,  
    list[str] seznam_vyucujících)
```

Výpis rozvrhu do console.  
:param rozvrhy: seznam vytvořených rozvrhů  
:param seznam\_oboru: seznam oborů  
:param seznam\_predmetu: seznam předmětů  
:param seznam\_mistnosti: seznam místností  
:param seznam\_vyucujících: seznam vyučujících  
:return: None

## 4.13.2 Dokumentace proměnných

### 4.13.2.1 rozvrh\_type

```
vypis_vytvorených_rozvrhu.rozvrh_type = list
```



## Kapitola 5

# Dokumentace souborů

### 5.1 Dokumentace souboru gui.py

#### Jmenné prostory

- namespace [gui](#)

#### Funkce

- [gui.Prevod\\_Rozvrhu](#) (rozvrhy, index, seznam\_delkaPredmetu\_predmet)
- [gui.Proved\\_PrevodUpravRozvrhuProOboryNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [gui.Proved\\_PrevodUpravRozvrhuProMistnostiNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [gui.Proved\\_PrevodUpravRozvrhuProVyucujícíNaSlovník](#) (upraveny\_rozvrh, seznam\_oboru, seznam\_predmetu, seznam\_mistnosti, seznam\_vyucujících)
- [gui.Vypis\\_rozvrhu](#) (rozvrh\_oboru, canvas\_frame, obor\_name, index)
- [gui.Vykresli\\_Rozvrh](#) (rozvrh\_combobox, canvas\_frame, canvas\_container, rozvrhy, seznam\_oboru, seznam\_mistnosti, seznam\_vyucujících, seznam\_predmetu, delky\_predmetu, rozvrh\_combobox2)
- [gui.on\\_close](#) (root)
- [gui.on\\_mouse\\_wheel](#) (event, canvas\_container)
- [gui.Vypis\\_Gui](#) (rozvrhy, seznam\_oboru, seznam\_mistnosti, seznam\_vyucujících, seznam\_predmetu, delky\_predmetu)

#### Proměnné

- list [gui.dny](#) = ["Pondělí", "Úterý", "Středa", "Čtvrtek", "Pátek"]

### 5.2 Dokumentace souboru hledej\_a\_zapis\_pred\_napevno.py

#### Jmenné prostory

- namespace [hledej\\_a\\_zapis\\_pred\\_napevno](#)

**Funkce**

- [int, int] [hledej\\_a\\_zapis\\_pred\\_napevno.Zjisti\\_KdyZacinaUcit](#) (int vyuc, list[list[int]] vyucujici\_casy)
- int [hledej\\_a\\_zapis\\_pred\\_napevno.Zjisti\\_PocetPrednasekVyuc](#) (pppdv, vyuc)
- [hledej\\_a\\_zapis\\_pred\\_napevno.Proved\\_ZapisPredmetuNapevno](#) (data\_o\_jiz\_zapsanych\_p, vyucujici\_casy, pppdv, seznam\_delkaPredmetu\_predmet, seznam\_vyuc, pppdo, pref\_mistnosti, seznam\_predmetu, seznam\_vyucujicich, seznam\_mistnosti, mistnosti\_casy)
- bool [hledej\\_a\\_zapis\\_pred\\_napevno.Zjisti\\_Uci1ci2PredmetyPro1obor](#) (seznam\_predmety\_pp\_delky, pppdo)

**5.3 Dokumentace souboru hodnoceni\_predmetu.py****Jmenné prostory**

- namespace [hodnoceni\\_predmetu](#)

**Funkce**

- list[float] [hodnoceni\\_predmetu.Proved\\_Hodnoceni\\_Predmetu](#) (list[list[int]] pppdv, list[list[list[int]]] vyucujici\_casy, list[str] predmety, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[list[int]] pppdo, list[int] index\_vlivu)
- int [hodnoceni\\_predmetu.Zjisti\\_JakyDenPrevazneUci](#) (list[list[list[int]]] vyucujici\_casy, list[int] vyucujici)
- list[int] [hodnoceni\\_predmetu.Zjisti\\_Seznam\\_Vyucujicich](#) (list[list[int]] pppdv, int predmet)
- int [hodnoceni\\_predmetu.Zjisti\\_Pocet\\_Vyucovanych\\_Prednasek](#) (list[list[int]] pppdv, list[int] seznam\_vyucujicich\_pro\_tento\_predmet, int predmet)
- [list[int], list[int], list[float]] [hodnoceni\\_predmetu.Vytvor\\_SeznamPoradiPredmetu](#) (list[list[int]] pppdv, list[list[list[int]]] vyucujici\_casy, list[str] seznam\_predmetu, list[int] seznam\_delkaPredmetu\_predmet, list[list[int]] preference\_mistnosti, list[list[int]] pppdo)
- list[int] [hodnoceni\\_predmetu.Uprav\\_Skupiny](#) (dict[list[int]:float] skup)
- list[list[int]] [hodnoceni\\_predmetu.VytvorVlivy](#) ()

**5.4 Dokumentace souboru hodnoceni\_vytvoreneho\_rozvrhu.py****Jmenné prostory**

- namespace [hodnoceni\\_vytvoreneho\\_rozvrhu](#)

**Funkce**

- [list[list[int]], list] [hodnoceni\\_vytvoreneho\\_rozvrhu.Ohodnot\\_Rozvrh](#) (list[rozvrh\_type] vytvorene\_rozvrhy, rozvrh\_type rozvrh, list[list[int]] pppdo, list[str] seznam\_predmetu, list[int] pouzite\_vlivy, list[list[int]] seznam\_vlivu\_vytvorených\_rozvrhu, seznam\_vliv\_predmety\_hodnoty\_nezapsaneP, list[int] predmety\_dle\_poradi, list[float] hodnoty\_premetu)
- list[list[int]] [hodnoceni\\_vytvoreneho\\_rozvrhu.Najdi\\_Nezapsane\\_Predmety](#) (list[list[int]] pppdo, list[str] predmety)
- list[int] [hodnoceni\\_vytvoreneho\\_rozvrhu.Zjisti\\_Interval](#) (rozvrh\_type rozvrh, int obor, int den)

**Proměnné**

- [hodnoceni\\_vytvoreneho\\_rozvrhu.rozvrh\\_type](#) = list



## 5.5 Dokumentace souboru kontroly.py

### Jmenné prostory

- namespace [kontroly](#)

### Funkce

- list[str] [kontroly.Kontrola\\_Vst\\_Dat](#) (list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_delkaPredmetu, list[list[str]] seznam\_predmet\_mistnost\_poradi, list[list[str]] seznam\_vyucujici\_predmet\_pocetPrednasek, list[list[str]] data\_o\_jiz\_zapsanych\_p, list[int] delky\_predmetu, list[str] seznam\_vyucujicich, list[str] seznam\_mistnosti, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] mistnosti\_casy, list[list[int]] pppdv, list[int] seznam\_delkaPredmetu\_predmet)
- None [kontroly.Kontrola\\_PredmetuNapevnoZapsanych](#) (list[str] chyby, list[list[str]] data\_o\_jiz\_zapsanych\_p, list[str] seznam\_predmetu, list[str] seznam\_vyucujicich, list[str] seznam\_mistnosti, list[int] delky\_predmetu, list[list[list[int]]] vyucujici\_casy, list[list[list[int]]] mistnosti\_casy)
- bool [kontroly.Kontrola\\_JsouSpravneHodnoty](#) (int cislo, int od, int do)
- bool [kontroly.Kontrola\\_JeSpravneKod](#) (str kod, list[str] seznam)
- bool [kontroly.Zjisti\\_JeZdePredmetNavic](#) (list[str] seznam\_predmetu, list kontrolovany\_seznam, int predmet, int index)
- bool [kontroly.Zjisti\\_ZdaChybiPredmet](#) (list[str] seznam\_predmetu, list kontrolovany\_seznam, int radek, int index)
- None [kontroly.Kontrola\\_PredmetDelkaPrednasky](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_delkaPredmetu)
- None [kontroly.Kontrola\\_PredmetMistnostPoradi](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_predmet\_mistnost\_poradi)
- None [kontroly.Kontrola\\_VyucujiciPredmetPocetPrednasek](#) (list[str] chyby, list[str] seznam\_predmetu, list[list[str]] seznam\_vyucujici\_predmet\_pocetPrednasek)
- None [kontroly.Kontrola\\_CasuVyucujicich](#) (list[str] chyby, list[list[list[int]]] vyucujici\_casy, list[list[int]] pppdv, list[int] seznam\_delkaPredmetu\_predmet, list[str] seznam\_vyuc)
- [kontroly.Kontrola\\_DelekPrednasek](#) (int vyuc, list[list[list[int]]] vyucujici\_casy, list[list[int]] seznam\_predmety\_pp\_delky, list[str] seznam\_vyuc)
- None [kontroly.Vypis\\_Chyby](#) (list[str] chyby, list[str] upozorneni)
- int [kontroly.Zjisti\\_MaxPocetJednicekZaSebou](#) (list[list[list[int]]] vyucujici\_casy, int vyuc, list[str] seznam\_vyuc)

## 5.6 Dokumentace souboru LP2.py

### Jmenné prostory

- namespace [LP2](#)

### Funkce

- [LP2.Vytvor\\_Sloty](#) (seznam\_casy, pocet\_radku)
- [LP2.LP](#) (pppdo, pppdv, vyucujici\_casy, obory\_casy, mistnosti\_casy, seznam\_oboru, seznam\_predmetu, seznam\_vyucujicich)

## 5.7 Dokumentace souboru main.py

### Jmenné prostory

- namespace [main](#)

## Proměnné

- `main.start = time.time()`
- list `main.dny = ["Po", "Ut", "St", "Ct", "Pa"]`
- list `main.delka_Predmetu = [2, 3, 4]`
- `main.seznam_obor_predmet = Nacti_CSV(os.path.join('csv', 'obor_predmet.csv'))`
- `main.seznam_predmet_mistnost_poradi = Nacti_CSV(os.path.join('csv', 'predmet_mistnost_poradi.csv'))`
- `main.seznam_predmet_delkaPredmetu = Nacti_CSV(os.path.join('csv', 'predmet_delkaPredmetu.csv'))`
- `main.seznam_vyucujici_predmet_pocetPrednasek = Nacti_CSV(os.path.join('csv', 'vyucujici_predmet_↵  
pocetPrednasek.csv'))`
- `main.data_o_jiz_zapsanych_p = Nacti_CSV(os.path.join('csv', 'predmety_napevno.csv'))`
- `main.seznam_vyucujici_casy = Nacti_JSON(os.path.join('json', 'vyucujici_casy.json'))`
- `main.seznam_mistnosti_casy = Nacti_JSON(os.path.join('json', 'mistnosti_casy.json'))`
- `main.seznam_oboru = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Oboru(seznam_obor_predmet)`
- `main.seznam_predmetu = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Predmetu(seznam_obor_predmet)`
- `main.seznam_vyucujich = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Vyuc(seznam_vyucujici_predmet_pocetPred`
- `main.seznam_mistnosti = tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Mistnosti(seznam_predmet_mistnost_poradi)`
- `main.pocet_potrebných_prednasek_dle_oboru = tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDO(seznam_obor_predmet,  
seznam_oboru, seznam_predmetu)`
- `main.pocet_potrebných_prednasek_dle_vyucujiciho`
- `main.preference_mistnosti = tvorba_seznamu_z_nactenych_dat.Vytvor_PrefMist(seznam_predmet_mistnost_poradi,  
seznam_mistnosti, seznam_predmetu)`
- `main.seznam_delkaPredmetu_predmet = tvorba_seznamu_z_nactenych_dat.Vytvor_DP(seznam_predmet_delkaPredmetu,  
seznam_predmetu)`
- list `main.upozorneni = []`
- `main.vyucujici_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_vyucujici_casy, seznam_vyucujich,  
upozorneni, 0)`
- `main.mistnosti_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Casy(seznam_mistnosti_casy, seznam_mistnosti,  
upozorneni, 1)`
- `main.obory_casy = tvorba_seznamu_z_nactenych_dat.Vytvor_Tabulku_Obsazenosti(seznam_oboru)`
- `main.chyby`
- `main.seznam_poradi_predmetu = pomocne_funkce.Odstran_OpakPrvkySeznamu(seznam_poradi_↵  
predmetu)`
- `main.seznam_vlivu`
- `main.seznam_hodnoceni`
- `main.sablona_rozvrhu`
- list `main.pocet_prepsani = [3, int((len(seznam_predmetu) - (len(seznam_predmetu) % 2))/2)]`
- list `main.vytvorene_rozvrhy = []`
- list `main.seznam_vlivy_vytv_roz = []`
- list `main.seznam_vliv_predmety_hodnoceni_nezapPred = []`
- list `main.posl = []`
- list `main.vlivy_posl = []`
- list `main.hodnoceni_posl = []`
- `main.rozvrhy = pomocne_funkce.Odstran_OpakujiciSeRozvrhy(vytvorene_rozvrhy)`
- `main.konec = time.time()`
- `main.cas = round(konec - start, 2)`
- str `main.cesta_k_pom_excel = "xlsx/vlivy_predmety_hodnoty_nezapPred.xlsx"`

## 5.8 Dokumentace souboru nacti\_data.py

### Jmenné prostory

- namespace `nacti_data`

**Funkce**

- list [nacti\\_data.Nacti\\_JSON](#) (str cesta)
- list [nacti\\_data.Nacti\\_CSV](#) (str cesta)

**5.9 Dokumentace souboru pomocne\_funkce.py****Jmenné prostory**

- namespace [pomocne\\_funkce](#)

**Funkce**

- int [pomocne\\_funkce.Zjisti\\_PocetJednicekVyuc](#) (int vyuc, list[list[list[int]]] vyucujici\_casy)
- list[list[int]] [pomocne\\_funkce.Najdi\\_PredmetyVyucujiciho](#) (list[list[int]] pppdv, int vyuc, list[int] seznam\_delka←  
Predmetu\_predmet, list[str] seznam\_vyuc)
- list[int] [pomocne\\_funkce.Najdi\\_Mistnosti\\_Predmetu](#) (list[list[int]] pref\_mistnosti, int predmet)
- list[int] [pomocne\\_funkce.Najdi\\_Cas\\_Vyucujiciho](#) (list[list[list[int]]] casy\_vyuc, int vyuc, int delka\_prednasky,  
list[list[int]] jiz\_vybrane)
- list[int] [pomocne\\_funkce.Najdi\\_Vyucujici\\_Predmetu](#) (list[list[int]] pppdv, int predmet)
- bool [pomocne\\_funkce.Zjisti\\_Je\\_Mistnost\\_Volna](#) (list[list[list[int]]] obs\_mist, int mistnost, list[int] den\_odH, int  
delka\_prednasky)
- bool [pomocne\\_funkce.Zjisti\\_Ma\\_Vyucujici\\_Volno](#) (list[list[list[int]]] obs\_vyuc, int vyuc, int den, int cas, int  
delka\_prednasky)
- None [pomocne\\_funkce.Vypis\\_Vlivy](#) (list[list[int]] vlivy)
- int [pomocne\\_funkce.Zjisti\\_CisloPomociKodu](#) (list[str] seznam, str kod)
- int [pomocne\\_funkce.Zjisti\\_Pocet\\_Volnych\\_Hodin](#) (list[list[list[int]]] vyucujici\_casy, list[int] seznam←  
vyucujicich\_pro\_tento\_predmet)
- int [pomocne\\_funkce.Zjisti\\_Pocet\\_Mistnosti](#) (list[list[int]] preference\_mistnosti, int predmet)
- list[int] [pomocne\\_funkce.Zjisti\\_Obory\\_Predmetu](#) (list[list[int]] pppdo, int predmet)
- bool [pomocne\\_funkce.Zjisti\\_ZdaPredmetJeNapevno](#) (list[list[str]] data\_o\_jiz\_zapsanych\_p, list[str] seznam←  
\_predmetu, int predmet)
- bool [pomocne\\_funkce.Zjisti\\_JeVolnoVRozvrhu](#) ([rozvrh\\_type](#) rozvrh, int den, int cas, int delka\_prednasky,  
list[int] obory\_predmetu)
- list [pomocne\\_funkce.Odstran\\_OpakPrvkySeznamu](#) (list seznam)
- list[[rozvrh\\_type](#)] [pomocne\\_funkce.Odstran\\_OpakujiciSeRozvrhy](#) (list[[rozvrh\\_type](#)] rozvrhy)

**Proměnné**

- [pomocne\\_funkce.rozvrh\\_type](#) = list

**5.10 Dokumentace souboru REAMDE.md****5.11 Dokumentace souboru tvorba\_rozvrhu.py****Jmenné prostory**

- namespace [tvorba\\_rozvrhu](#)

## Funkce

- `tvorba_rozvrhu.TvorRozvrhyProRuzneVlivy` (`list[list[int]] seznam_poradi_predmetu`, `list[list[int]] seznam_vlivu`, `list[list[int]] pocet_potrebných_prednasek_dle_oboru`, `list[list[int]] pocet_potrebných_prednasek_dle_vyucujiciho`, `list[list[list[int]]] vyucujici_casy`, `list[list[list[int]]] obory_casy`, `list[list[list[int]]] mistnosti_casy`, `list[int] seznam_delkaPredmetu_predmet`, `list[list[int]] preference_mistnosti`, `list[str] seznam_predmetu`, `rozvrh_type` `sablona_rozvrhu`, `list[list[float]] seznam_hodnoceni`, `list[str] seznam_oboru`, `list[str] seznam_mistnosti`, `list[str] seznam_vyucujicich`, `list[list[str]] data_o_jiz_zapsanych_p`, `int pocet_prepsani`)
- `tvorba_rozvrhu.Tvor_RozvrhyUpravouPosloupnosti` (`posl_predmetu`, `vlivy_posl`, `list[list[int]] pocet_potrebných_prednasek_dle_oboru`, `list[list[int]] pocet_potrebných_prednasek_dle_vyucujiciho`, `list[list[list[int]]] vyucujici_casy`, `list[list[list[int]]] obory_casy`, `list[list[list[int]]] mistnosti_casy`, `list[int] seznam_delkaPredmetu_predmet`, `list[list[int]] preference_mistnosti`, `list[str] seznam_predmetu`, `rozvrh_type` `sablona_rozvrhu`, `hodnoceni_posl`, `list[str] seznam_oboru`, `list[str] seznam_mistnosti`, `list[str] seznam_vyucujicich`)
- `list[int] tvorba_rozvrhu.Uprav_Posloupnost` (`list[int] posloupnost`, `int nezap_predmet`, `list[list[int]] pouzite_posl`, `int pocet_predmetu`)
- `rozvrh_type tvorba_rozvrhu.Proved_PridejDoSablonyRozvrhu` (`rozvrh_type` `rozvrh`, `list[int] data_k_presunuti`, `list[list[list[int]]] obs_vyuc`, `list[list[list[int]]] obs_mist`, `list[list[list[int]]] obs_oboru`, `list[list[int]] pppdv`, `list[list[int]] pppdo`, `list[str] seznam_vyuc`)
- `[list[rozvrh_type], list[list[int]], list[list[int]], list[list[list[int]]], list[list[list[int]]], list[list[list[int]]]] tvorba_rozvrhu.Vytvor_Rozvrh_dlePredmetu` (`list[int] posloupnost_predmetu`, `list[list[int]] pom_pppdo`, `list[list[int]] pom_pppdv`, `list[list[list[int]]] pom_obs_vyuc`, `list[list[list[int]]] pom_obs_oboru`, `list[list[list[int]]] pom_obs_mist`, `list[int] delky_prednasek`, `list[list[int]] prefer_mist`, `rozvrh_type` `sablona_rozvrhu`, `list[str] seznam_oboru`, `list[str] seznam_predmetu`, `list[str] seznam_mistnosti`, `list[str] seznam_vyucujicich`)
- `list[int] tvorba_rozvrhu.Zjisti_MohouLiSePremistitPredmety` (`rozvrh_type` `rozvrh`, `list[list[list[int]]] obs_vyuc`, `list[list[list[int]]] obs_oboru`, `list[list[list[int]]] obs_mist`, `list[list[int]] nezapsane_predmety`, `list[list[int]] pppdv`, `list[int] delky_predmetu`, `list[list[int]] pppdo`, `list[list[int]] prefer_mist`, `list[list[list[int]]] vyucujici_casy`, `list[str] seznam_predmetu`, `list[str] seznam_vyucujicich`, `list[str] seznam_mistnosti`, `list[list[str]] data_o_jiz_zapsanych_p`)
- `None tvorba_rozvrhu.Obsad` (`list[list[list[int]]] casy`, `[int, int] den_odH`, `int delka_prednasky`, `list[int] kdo_se_ma_obsadit`)
- `None tvorba_rozvrhu.Zapis_Do_Rozvrhu` (`list[int] seznam_oboru`, `[int, int] den_odH`, `int delka_prednasky`, `int vyuc`, `int mistnost`, `rozvrh_type` `rozvrh`, `int predmet`)
- `None tvorba_rozvrhu.Odecti_Pocet_Prednasek` (`list[list[int]] pppd`, `list[int] seznam_oboru_nebo_vyuc`, `int predmet`)
- `bool tvorba_rozvrhu.Zjisti_Maji_Obory_Volno` (`list[list[list[int]]] obs_oboru`, `list[int] obory`, `[int, int] den_odH`, `int delka_prenasky`)
- `list[list[int]] tvorba_rozvrhu.Najdi_VyucBlokujiciNezapPredmet` (`rozvrh_type` `rozvrh`, `list[list[list[int]]] vyucujici_casy`, `list[int] vyuc_nezap_predmetu`, `list[list[list[int]]] obs_oboru`)
- `rozvrh_type tvorba_rozvrhu.Vytvor_SablonuRozvrhu` (`int pocet_oboru`, `list[list[int]] pppdo`, `list[list[int]] pppdv`, `list[list[list[int]]] obs_oboru`, `list[list[list[int]]] obs_vyuc`, `list[list[list[int]]] obs_mist`, `list[str] seznam_predmetu`, `list[str] seznam_vyuc`, `list[str] seznam_mistnosti`, `list[list[str]] data_o_jiz_zapsanych_p`, `list[str] seznam_oboru`)

## Proměnné

- `tvorba_rozvrhu.rozvrh_type` = `list`

## 5.12 Dokumentace souboru tvorba\_seznamu\_z\_nactenych\_dat.py

### Jmenné prostory

- namespace `tvorba_seznamu_z_nactenych_dat`

**Funkce**

- `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Vyuc` (seznam)
- `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Oboru` (seznam)
- `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Mistnosti` (seznam)
- `tvorba_seznamu_z_nactenych_dat.Priprav_Seznam_Predmetu` (seznam)
- `tvorba_seznamu_z_nactenych_dat.Vytvor_Matici` (radky, sloupce)
- `tvorba_seznamu_z_nactenych_dat.Vytvor_DP` (seznam\_predmet\_delkaPredmetu, predmety)
- `list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDO` (list[list[str]] seznam, list[str] obory, list[str] predmety)
- `list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PPPDV` (list[list[str]] seznam, list[str] vyucujici, list[str] predmety)
- `list[list[int]] tvorba_seznamu_z_nactenych_dat.Vytvor_PrefMist` (list[list[str]] seznam, list[str] mistnosti, list[str] predmety)
- `list[list[list[int]]] tvorba_seznamu_z_nactenych_dat.Vytvor_Casy` (list seznam, list[str] pom, list[str] upozor-neni, int index\_seznamu)
- `list[list[list[int]]] tvorba_seznamu_z_nactenych_dat.Vytvor_Tabulku_Obsazenosti` (obory)
- `rozvrh_type tvorba_seznamu_z_nactenych_dat.Vytvor_Prazdny_Rozvrh` (int pocet\_oboru)

**Proměnné**

- `tvorba_seznamu_z_nactenych_dat.rozvrh_type` = list

## 5.13 Dokumentace souboru vypis\_pomocnych\_udaju.py

**Jmenné prostory**

- namespace `vypis_pomocnych_udaju`

**Funkce**

- `vypis_pomocnych_udaju.Vypis_PomUdajeDoExcelu` (str cesta, pom\_seznam, seznam\_predmetu)

## 5.14 Dokumentace souboru vypis\_vytvorených\_rozvrhu.py

**Jmenné prostory**

- namespace `vypis_vytvorených_rozvrhu`

**Funkce**

- None `vypis_vytvorených_rozvrhu.Vypis_Rozvrh` (list[rozvrh\_type] rozvrhy, list[str] seznam\_oboru, list[str] seznam\_predmetu, list[str] seznam\_mistnosti, list[str] seznam\_vyucujících)

**Proměnné**

- `vypis_vytvorených_rozvrhu.rozvrh_type` = list

