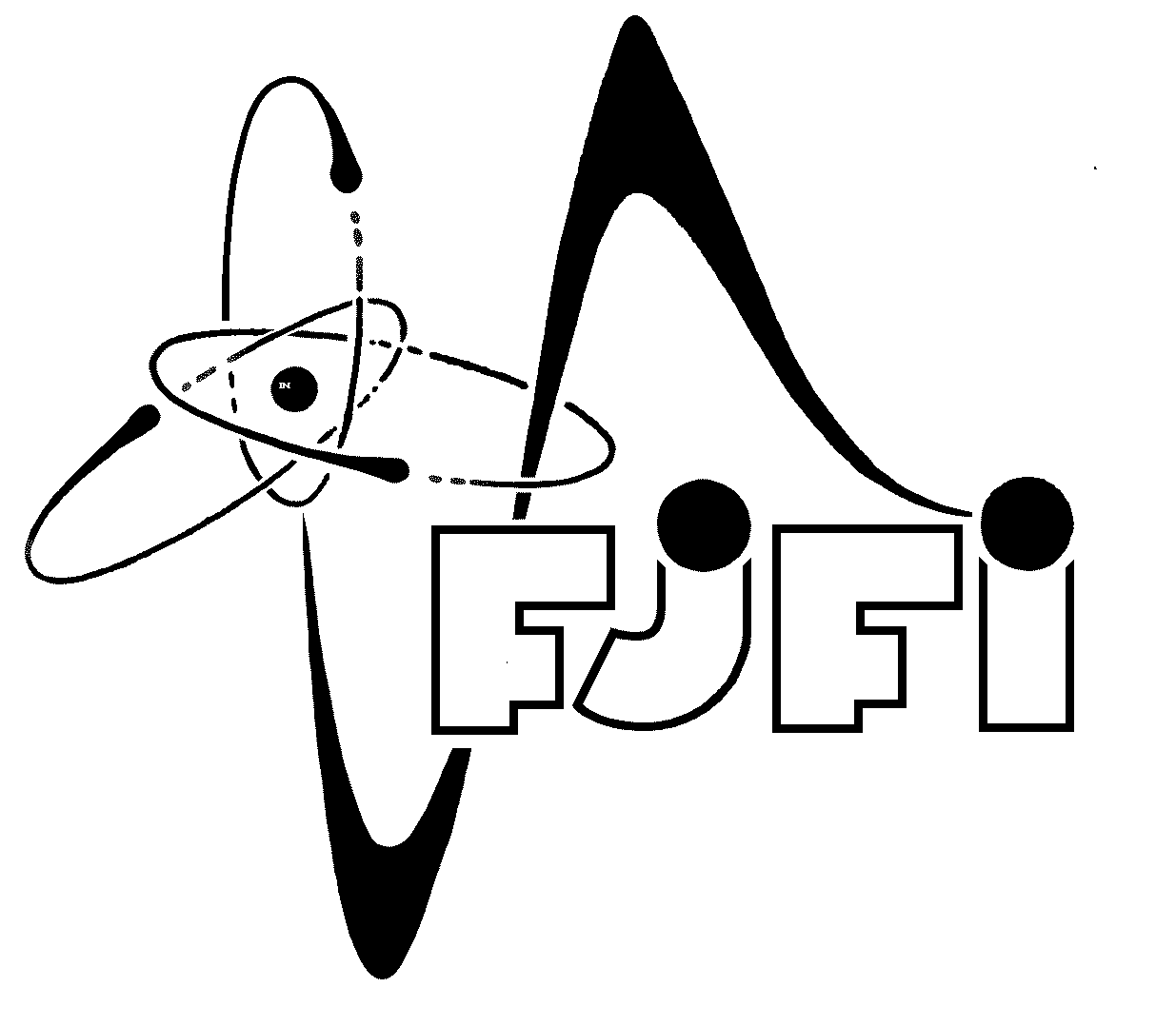
FJFI ČVUT v Praze



**Specifikace filtrování**

**-**

**Aplikace Rozvrh FJFI**

|  |  |  |
| --- | --- | --- |
|  | datum vytvoření: | 11.11.2013 |
|  | verze: | 1.0 |
|  | počet stran: | 6 |
|  | počet příloh: | 0 |
|  | druh dokumentu: | Zadání |

OBSAH

[1. Historie 3](#_Toc371972139)

[2. Specifikace zadání 3](#_Toc371972140)

[2.1 Gui návrh filtrovacího panelu 3](#_Toc371972141)

[2.2 Programátorský popis řešení 4](#_Toc371972142)

[2.3 Seznam veřejných proměnných pro kontejnery GUI 4](#_Toc371972143)

[2.4 Seznam veřejných Filterů 5](#_Toc371972144)

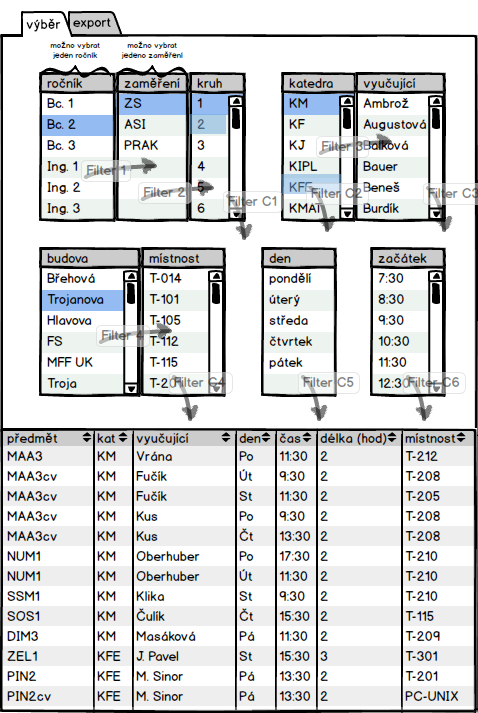
[3. Pozn.: Manuální editece timetablefield 5](#_Toc371972145)

# Historie

|  |  |  |
| --- | --- | --- |
| **Datum** | **Změna** | **Autor** |
| 11.11.2013 | Prvotní návrh | David Blatský |
| 12.11.2013 | Přidány poznámky | Oldřich Štika |

# Specifikace zadání

## Gui návrh filtrovacího panelu



Obrázek : Návrh pravého panelu GUI.

V panelu jsou nahoře seznamy parametrů podle kterých se filtruje a dole výpis odpovídajících hodin.

Na obrázku jsou filtry vyznačeny šipkami .

Seznamy jsou následující:

1. Ročník – z něho se dá vybrat pouze jedna položka, na základě které se vyberou hodnoty do seznamu Zaměření
2. Zaměření - z něho se dá vybrat pouze jedna položka, na základě které se vyberou hodnoty do seznamu Kruh
3. Kruh – je možné vybírat libovolně mnoho položek. Vybrané kruhy jsou vstupem pro filtr, který produkuje seznam hodin.
4. Katedra – je možné vybírat libovolně mnoho položek. Vybrané katedry jsou vstupem pro filtr, který produkuje seznam hodin ale i pro filtr, který produkuje vedlejší seznam učitelů. Tedy vyberou se hodiny vypisované danými katedrami a učitelé spadající pod dané katedry. Pokud se nevybere žádná položka, filtr se nevezme v potaz.
5. Vyučující – Je možno vybrat libovolné množství. Pokud se nevybere žádný, zobrazí se předměty vybraných kateder.
6. Budova – Libovolné množství je možno vybrat. Zobrazí se místnosti ve vybraných budovách.
7. Místnost – Zobrazí se hodiny vedené ve vybraných místnostech.
8. Den – Zobrazí se hodiny vedené v daný den.
9. Čas – Zobrazí se hodiny vedené v daný Čas.

## Programátorský popis řešení

Pro každý seznam (ročník, zaměření, kruh…) jsou dostupné (z view componenty projektu) dvě instance objektu které je datově reprezentují příslušnou entitu (List<DegreeYear>, List<Zamereni>,…). Jeden seznam představuje zobrazené prvky seznamu, ve druhé seznamu jsou vybrané prvky. Například zobrazená (visible) zaměření jsou:

public List<Zamereni> v\_zamerenis { get; set; } // v\_ zmanená visible

Tento objekt je readonly, ve View se pouze jednoduše přepíšou hodnoty do příslušné komponenty GUI.

Potom co uživatel vybere nějaké zaměření, tak se ve View přiřadí do následující instance vybrané zaměření (zaměření se vybírá jen jedno, proto to není list).

public Zamereni s\_zamereni { get; set; } // s\_ zmanená selected

Po přiřazení do selected proměnné stačí zavolat příslušný filtr, který aktualizuje hodnoty ve visible proměnných. V tomto případě to bude FilterZamereni2Kruhy().

Podle mě je zbytečné držet zvlášť seznam vybraných polí a podle něj filtrovat. Filtry se stejně musí volat z View a tudíž jim lze rovnou vybrané položky předávat jako parametry a není třeba je speciálně plnit do dalších objektů.

## Seznam veřejných proměnných pro kontejnery GUI

public List<TimetableField> v\_timetablefields { get; private set; }

//data zobrazená v příslučných objektech gui - visible

public List<DegreeYear> v\_degreeyears { get; set; }

public List<Zamereni> v\_zamerenis { get; set; }

public List<Kruh> v\_kruhy { get; set; }

public List<Department> v\_departments { get; set; }

public List<Lecturer> v\_lecturers { get; set; }

public List<Building> v\_buildings { get; set; }

public List<Classroom> v\_classrooms { get; set; }

public List<Day> v\_days { get; set; }

public List<Time> v\_times { get; set; }

//označené kolonky v příslučných objektech view - selected

public DegreeYear s\_degreeyear { get; set; }

public Zamereni s\_zamereni { get; set; }

public List<Kruh> s\_kruhy { get; set; }

public List<Department> s\_departments { get; set; }

public List<Lecturer> s\_lecturers { get; set; }

public List<Building> s\_buildings { get; set; }

public List<Classroom> s\_classrooms { get; set; }

public List<Day> s\_days { get; set; }

public List<Time> s\_times { get; set; }

## Seznam veřejných Filterů

Šipky na obrázku znamenají filtery, první 4 pracují vždy jen s jedním „vstupem“ a jedním „výstupem“. Filter 5 má za „vstupy všechny položky, ze kterých míří šipky směrem dolů, „výstupem“ je pak aktualizovaný seznam v\_timetablefields.

public void FilterDegreeYear2Zamereni() – Filter 1

public void FilterZamereni2Kruhy() – Filter 2

public void FilterDepartments2Lecturers() – Filter 3

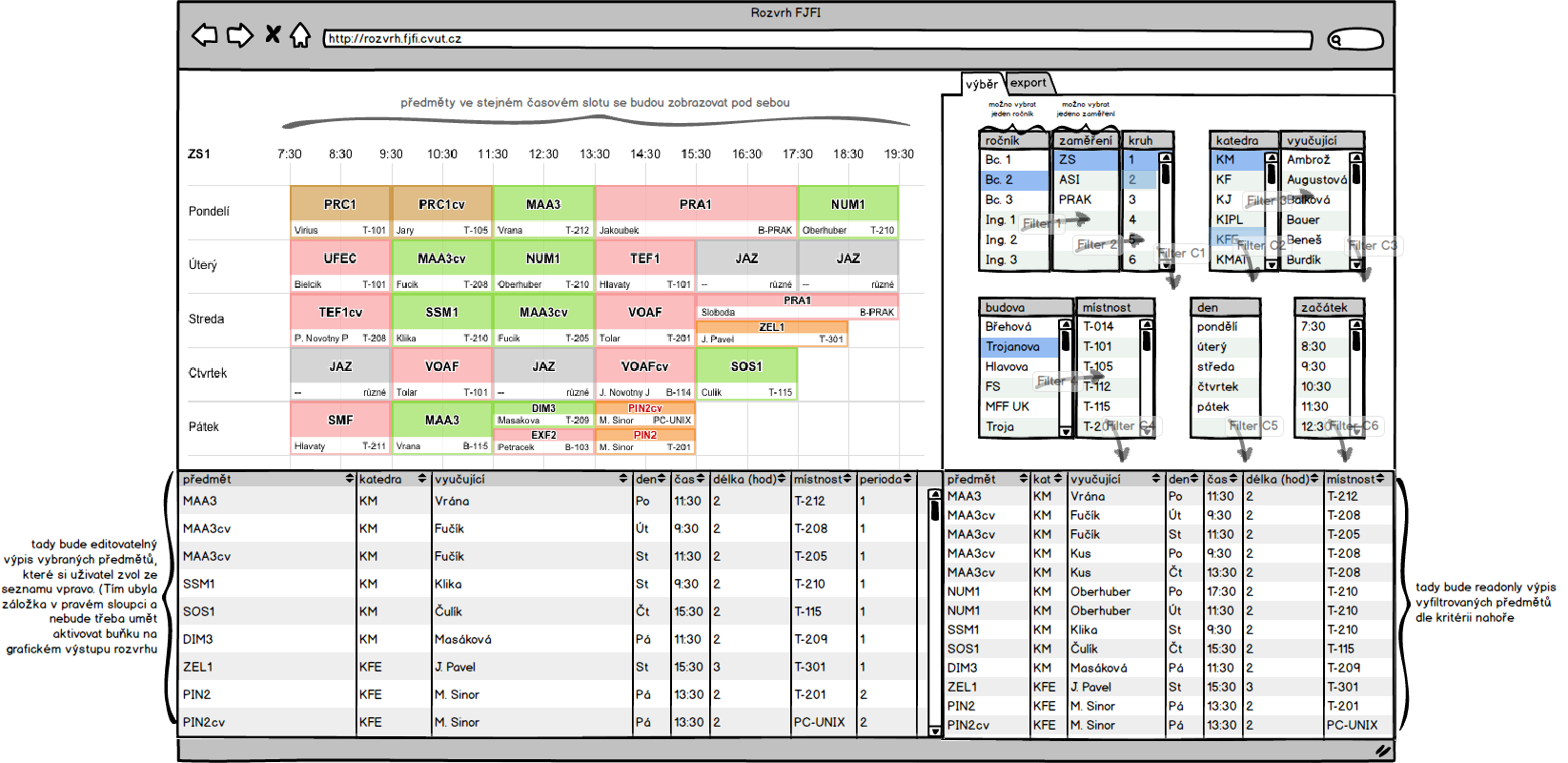
public void FilterBuildings2Classrooms() – Filter 4

public void FilterAll2TimetableFields() – Filter C

# Pozn.: Manuální editAce timetablefield

Jak bylo popsáno v prvotním zadání, aplikace bude umět ručně editovat zobrazené TimeTableFields. Pokud by se měly myší vybírat vykreslená políčka na ploše, bylo by to složité naprogramovat. Napadlo mě to vyřešit následovně. Udělat ještě jeden seznam hodin. Těch, které jsou vykreslené na ploše s možnosti editace hodnot v seznamu. Nevypadá to tak hezky, ale bylo by to jednodušší. Co na to říkáte?

A další věc, jak budeme řešit hodiny spolu „spojené“. Klasicky cvika z MAB, která jsou 2x týdně a standardně se předpokládá, že člověk bude chodit na v rozvrhu uveřejněné dvojice a nebude to nějak kombinovat…



Tohle vypadá hezky, ale aby se to vše vešlo na obrazovku a aby šly texty přečíst, tak to bude chtít celkem velký rozlišení, který ale nemusí mít každý k dispozici. A s posuvníky to už nebude tak dobře ovladatelné. Podle mě záložka Editace spojená s klikacím kontextovým menu nebyla špatný nápad.