

Week 06 – Design Patterns 2

422551 Václav Hála

422450 Michaela Bocánová

Design Patterns aplikované na plánovač rozvrhu semestru

Vzor Adapter

- a) rozvrh je vytváraný pomocou Genetic Algorithms (<http://kataklinger.com/index.php/genetic-algorithm-class-schedule/>)
adaptee: externá knihovňa, ktorá obsahuje tento algoritmus
adapter: prispôsobí knihovňu rozvrhu fakulty/MU
target: rozvrh fakulty/MU
- b) rozvrh je zobraziteľný v rôznych formátoch
adaptee: knižnice obsahujúce algoritmy pre konverziu formátov (pdf, tex, html...)
adapter: prispôsobí knižnice formátom rozvrhu

Vzor Composite

- rozvrh predmetu sa skladá z jednotlivých dátumov, času a miestnosti (a z rozvrhov jeho seminárnych skupín?)

component: rozvrh

composite: predmet/seminárna skupina (rozvrhnem si predmet alebo jeho seminárne skupiny, zmením všetky listy alebo ich podmnožinu, napr. ak dôjde k zmene po začiatku obdobia výuky, nemusí to byť jednorázová zmena/1 list)

leaf: čas, dátum a miesto výuky (napr. prednáška každý týždeň → má list pre každý týždeň, seminár každé dva týždne → list pre každé dva týždne)

Vzor Observer

- a) po zapísaní/zrušení predmetu dôjde k automatickej aktualizácii rozvrhu
subject: zoznam zaregistrovaných predmetov a prihlásených seminárnych skupín
observer: plánovač rozvrhu reaguje na zmeny v zozname tým, že pridá/odoberie predmety/semináre do/z rozvrhu
- b) po prihlásení/odhlásení skupiny dôjde k automatickej aktualizácii voľnej kapacity seminárnych skupín
subject: zoznam seminárnych skupín a ich aktuálna voľná kapacita (realtime)
observer: plánovač rozvrhu reaguje na zmeny v zozname tým, že znemožní/umožní prihlásenie do skupiny podľa aktuálnej voľnej kapacity
- c) sledovanie obsadenosti učební v čase
subject: rozvrh predmetu
observer: plánovač rozvrhu reaguje na zmeny v rozvrhu (ak bola zmena úspešná, aktualizuje sa zoznam voľných učební v čase)

Vzor Decorator

concrete component: základný rozvrh

concrete decorator: jeho rozšírenia

- detaily nižšie

Vzor Strategy

context: vytváranie rozvrhu, zobrazenie, vykreslenie, plánovanie

concrete strategy: implementuje konkrétnu stratégiu/algoritmus

- detaily nižšie

Vzor Builder

- a) typická aplikácia vzoru builder
builder: konvertor rozvrhu
concrete builder: tex, pdf, postscript, XML, iCalendar, HTML... konvertor
product: rozvrh vo formáte pdf, tex...
- b) TimetableBuilder – zostaví rozvrh, validný stav je bezkonfliktný
- detaily nižšie

Vzor Flyweight

-pre vykreslenie rozvrhu sú potrebné parametry rámcikov/lokácie
concrete flyweight: predmet/seminárna skupina
unshared concrete flyweight: deň v týždni (row), časové intervaly (column)

Vzor Factory Method

- rôzne typy rozvrhov
product: rozvrh
concrete product: rozvrh pravidelnej výuky (1 týždeň), rozvrh blokovej výuky (celý semester)
concrete creator: factory pre rozvrh pravidelnej výuky, factory pre rozvrh blokovej výuky

Detailnejší popis plánovania, vytvorenia a zobrazenia rozvrhu:

Pro *planovani* jsou v systemu dva hlavní rozhraní:

TimeSchedulable - nastavení času události v rámci dne, start a délka v hodinách a minutách.

DateSchedulable - nastavení kalendářního data kdy událost proběhne, daný je den a měsíc.

TimeSchedulable má dvě implementace:

SimpleTimeSchedulable - jedna událost které může být nastaven čas.

CompositeGroupTimeSchedulable - seznam události kterým může být dávkově nastaven čas, například je tedy možné mít instanci pro všechny výukové hodiny jedné seminární skupiny / přednášky, díky tomu je možné naraz upravit čas události v prvním, druhém, třetím... týdnu.

DateSchedulable má dvě implementace:

SimpleDateSchedulable - jedna událost které může být nastaven datum konání.

CompositeOffsetTimeSchedulable - seznam události s pevně daným rozestupem, nastavení datumu se propaguje na všechny události tak, že 1. události je nastaven přesně daný datum, druhé daný datum + 1x offset, k třetí je přidán offset 2x atd. Díky tomu je možné posouvat seminární skupiny / přednášky v rámci týdne, změna první je propagována na všechny další skupiny / přednášky ve zbytku semestru.

Tento model zároveň umožňuje změnit datum / čas jen jedné události, například pokud je přednášející nemocný a výjimečně se jedna přednáška posouvá o dva dny.

Pro přidání dodatečných informací do rozvrhu se používá vzor **Decorator**.

Vykreslení základního rozvrhu pro studenta (kód předmětu, čas) může být rozšířeno například pomocí TutorDecorator který ke každé přednášce / skupině přidá kolonku s jménem vyučujícího, RoomDecorator který přidá informaci o místnosti v které výuka probíhá nebo BlockCoursesDecorator který při výpisu pro konkrétní týden do příslušných dní a času přidá políčko blokove výuky.

Existuje mnoho způsobů vykreslení/výpisu kalendáře, každý z těchto způsobů je jedna **Strategy**. Tato Strategy pracuje s interní reprezentací rozvrhu, která je vytvořena za použití Decoratoru

popsaných výše.

Pro grafické vykreslení přímo ve internetovém prohlížeči existuje `StaticHtmlTimetableFormat` který vykresluje pouze výuku probíhající v pravidelných intervalech, dny jsou označeny jen jako pondělí, úterý... nikoli konkrétními daty.

Dále je k dispozici `CalendarHtmlTimetableFormat`, jehož výstup je také grafický, rozvrh je ale vykreslen jednotlivě pro vybrané týdny a obsahuje i jednorázové události jako je bloková výuka. Rozvrh je také možno vypsat do textového souboru který si uživatel může stáhnout pomocí `FileTimetableFormat`.

Tato *Strategy* interně používá další *Strategy* pro jednotlivé podporované formáty jako je PDF, Postscript, XML...

Například pro stažení rozvrhu ve formátu PDF bude použit `FileTimetableFormat` interně používající `PdfFormat`.

Pro *planování* rozložení hodin do místnosti při *sestavování* nového rozvrhu jsou použity vzory **Strategy** a **Composite**.

V systému je několik plánovacích algoritmů používajících různé heuristiky a přístupy, každý z nich je jednou implementací *Strategy*.

Všechny tyto algoritmy lze použít současně pomocí speciální `CompositePlanningStrategy`.

Tato třída implementuje stejné rozhraní pro *Strategy* jako jednotlivé konkrétní algoritmy, zároveň je to *Composite*.

Obsahuje pole konkrétních algoritmů (*Leaves*) a hodnotící funkci porovnávající výsledky spočítané jednotlivými algoritmy.

Výpočet `CompositePlanningStrategy` probíhá tak, že nad zadaným vstupem spustí všechny dostupné algoritmy, objektivní funkci porovná jejich výsledky a vrátí nejlepší nebo několik nejlepších výsledků.

Jednotlivé **Strategy** sestavující rozvrh používají *TimetableBuilder*.

Tento umí manipulovat s jednotlivými *Time* a *DateSchedulable*, validuje například že se žádné dvě naplánované události nepřekrývají.

Jednotlivé *Strategy* tedy jen vytváří a přeskládávají události pomocí API Builderu a jsou upozorněny pokud nastane nevalidní stav.