

Objektovo orientované programovanie

(Model/View/Controller)

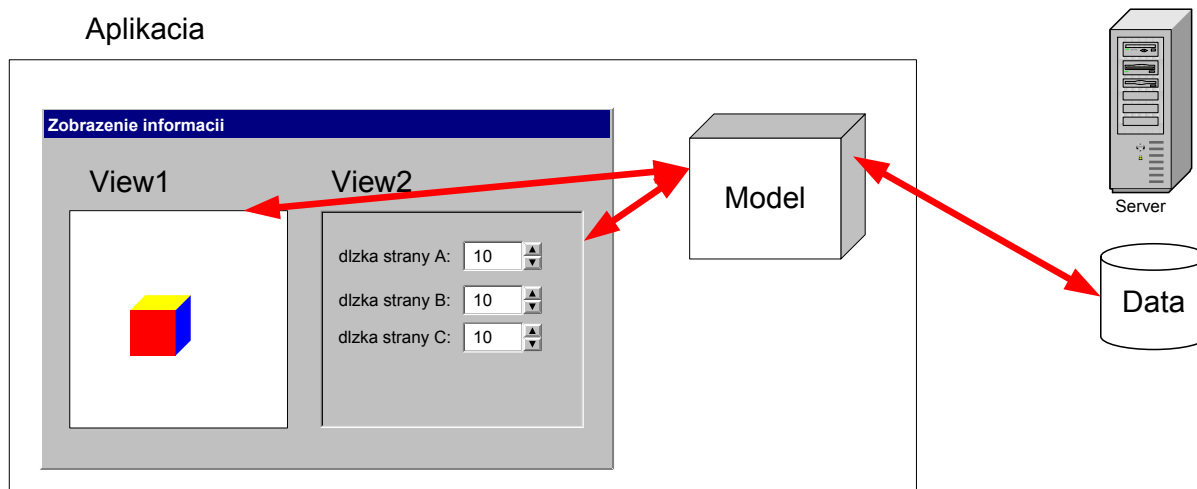
11. prednáška (3. časť)

Vladislav Novák
FEI STU v Bratislave
25.11.2014

Obsah

Architektonický vzor Model-View-Controller.....	1
Príklad (farba)	3

Architektonický vzor Model-View-Controller



Model-View-Controller – oddeľuje údaje a ich zobrazenie

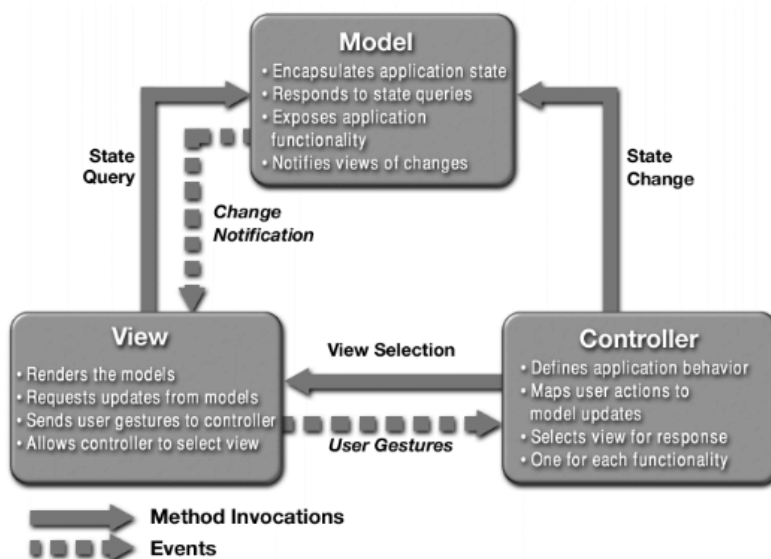
Model-View-Controller – je architektonický vzor. Definuje architektúru aplikácie tak, aby boli prípadné zmeny jednoduchšie realizovateľné.

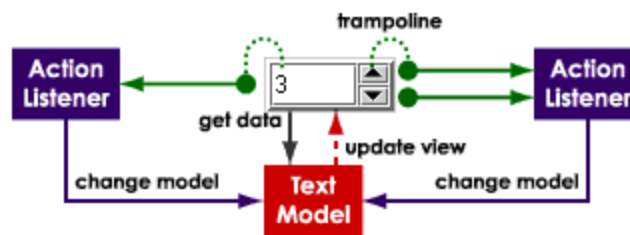
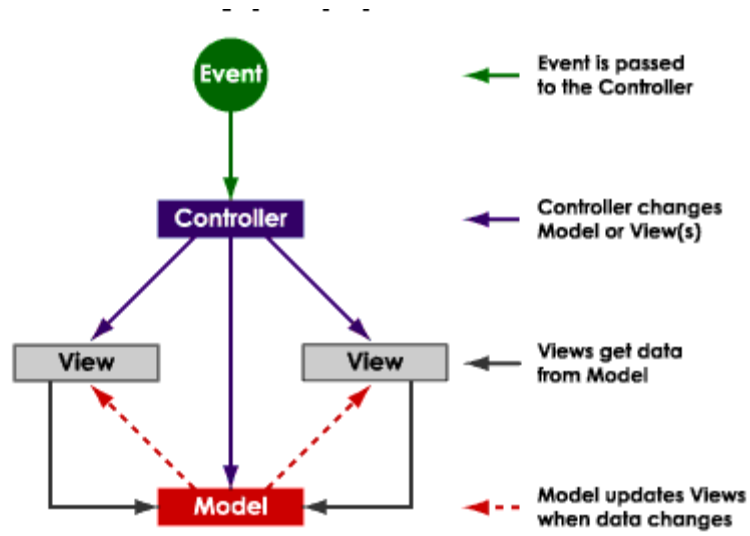
Model – reprezentuje údaje a pravidlá pre zmenu údajov. Nemusí priamo obsahovať údaje, môže len zabezpečovať sprístupnenie a modifikáciu údajov, ktoré sú napr. v databáze.

View – zobrazenie údajov

Controller – interpretuje akcie používateľa vykonané na view, transformuje ich na zmeny v modely (prípadne aj vo view)

Vzor nešpecifikuje ako presne má komunikácia medzi jeho jednotlivými časťami prebiehať. Nešpecifikuje ktoré časti medzi sebou komunikujú, nešpecifikuje či view ma byť listenerom modelu, alebo či má view kontrolovať zmeny v modely.





<http://www.enode.com/x/markup/tutorial/mvc.html>

Príklad (farba)

Zdrojové súbory na stránke predmetu.

V príklade je vzor Model-View-Controller použitý dva krát:

- pre číselnú hodnotu
niektoré súvisiace triedy:
 - o BoundedIntegerModel (model)
 - o Slider (view)
 - o Label (view)
- pre farbu
niektoré súvisiace triedy:
 - o ColorModel (model)
 - o ColorPanel (view)
 - o RgbSliders (view)
 - o NegateColorButton(view)

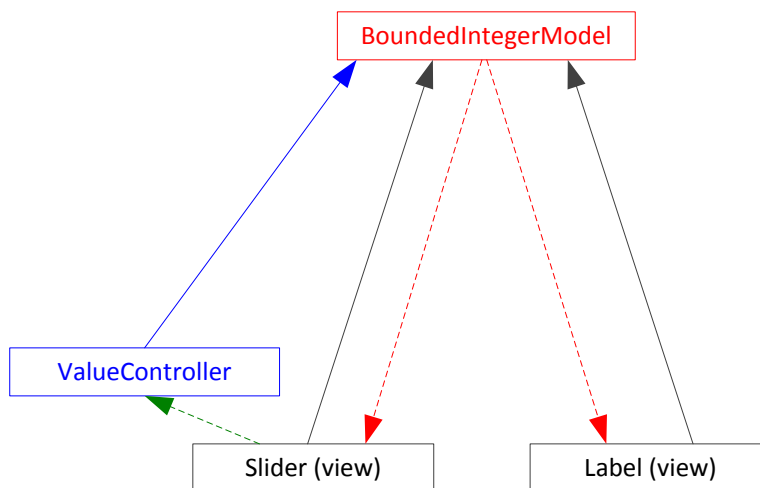
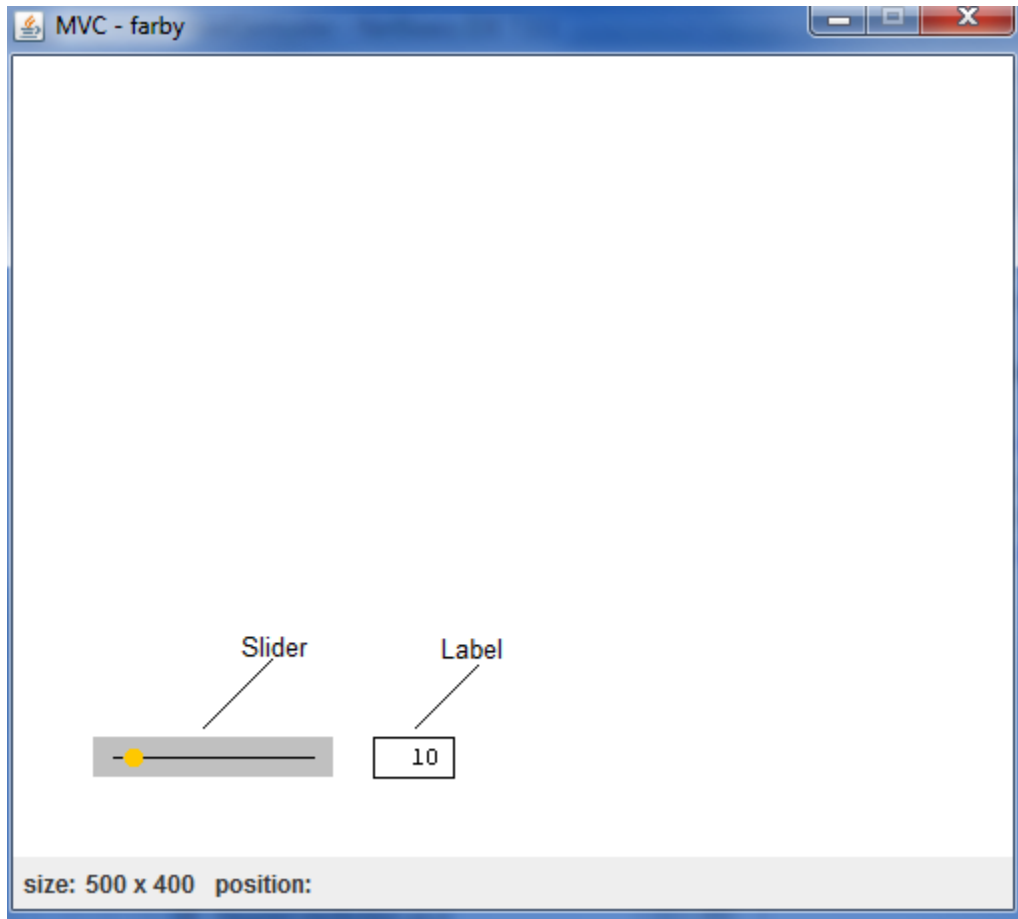
Popis Model-View-Controller pre číslo

Trieda `BoundedIntegerModel` je modelom reprezentujúcim hodnotu celého čísla v určitom rozsahu. O zmene hodnoty informuje všetky view generovaním udalostí (využitie návrhového vzoru `Observer`). Model-View-Controller nepredpisuje, že spôsob komunikácie musí byť cez udalosti, ale v tomto prípade je tento spôsob komunikácie výhodnejší.

Sú vytvorené 2 druhy view:

- `Label` – zobrazuje hodnotu čísla pomocou znakov. V konštruktore zistí aktuálnu hodnotu čísla reprezentovaného modelom.
- `Slider` – zobrazuje hodnotu čísla pozíciou bežca. Umožňuje aj meniť hodnotu čísla pri potiahnutí bežca používateľom. V konštruktore zistí aktuálnu hodnotu čísla reprezentovaného modelom a interval hodnôt v ktorom sa môže číslo meniť.

`Slider` obsahuje kontroler `ValueController`, ktorý interpretuje ťahanie bežca používateľom na zmenu hodnoty čísla.



- model upozorní view na zmenu údajov (udalosť)
- view získava údaje z modelu
- kontroler mení údaje v modeli (interpretuje význam akcie používateľ)
- view informuje o akcii používateľa (stlačenie tlačidla, posun myši)

Popis Model-View-Controller pre farbu

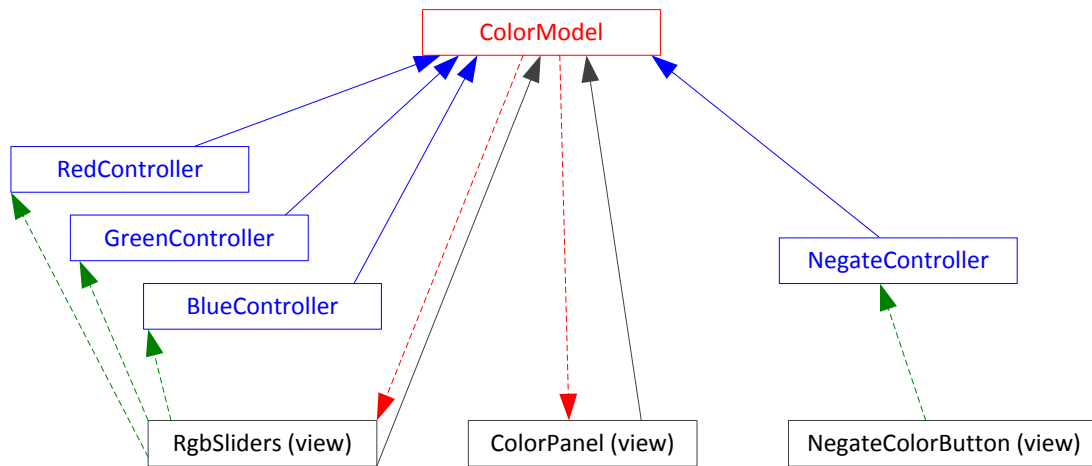
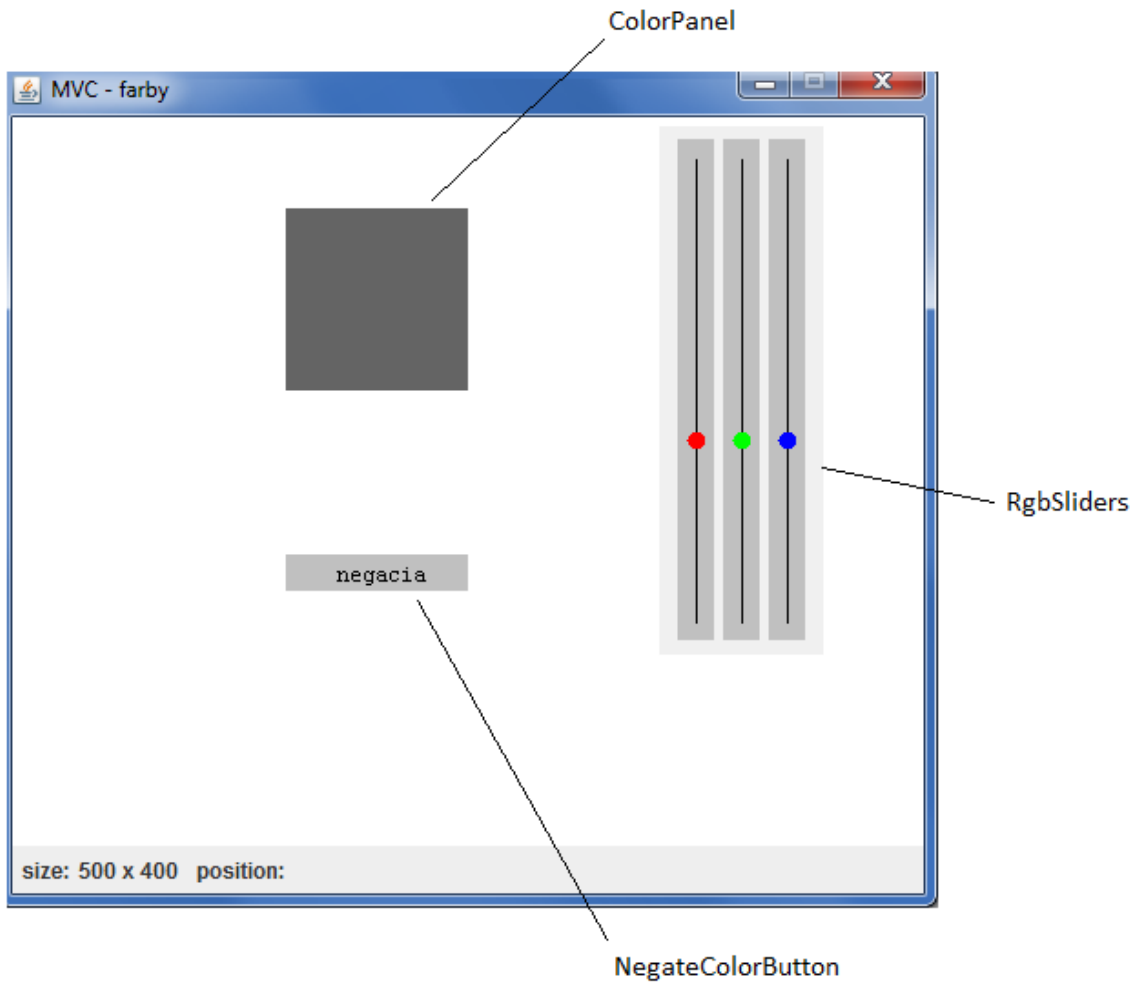
Trieda `ColorModel` je modelom reprezentujúcim údaje o farbe. O zmene farby informuje view generovaním udalostí (využitie návrhového vzoru `Observer`). Model-View-Controller nepredpisuje, že spôsob komunikácie musí byť cez udalosti, ale v tomto prípade je tento spôsob komunikácie výhodnejší.

Sú vytvorené 3 druhy view:

- `ColorPanel` – zobrazuje farbu tak, že svoju plochu vyfarbí podľa farby reprezentovanej modelom.
- `RgbSliders` – trojica slajderov, ktorá zobrazuje informácie o farbe aj umožňuje meniť farbu. Informácie o farbe zobrazuje tak, že podľa hodnôt zložiek farby sú nastavené pozície bežcov na jednotlivých slajderoch. Keď používateľ zmení pozíciu bežcov na slajderoch, zmení farbu reprezentovanú v modeli.
- `NegateColorButton` – tlačidlo, ktoré nezobrazuje informáciu o farbe, umožňuje len meniť jej hodnotu. Inštancia triedy `NegateColorButton` zmení farbu po kliknutí myšou (zneguje farbu).

`RgbSliders` obsahuje tri kontrolery. Tieto kontrolery interpretujú posúvanie bežcov na slajderoch. Posuv bežca transformujú na zmenu hodnoty niektorej zložky farby.

`NegateColorButton` obsahuje jeden kontroler, ktorý interpretuje kliknutie. Kliknutie transformuje na akciu znegovania farby.



- ◀------ model upozorní view na zmenu údajov (udalosť)
- ◀--- view získava údaje z modelu
- ▶--- kontroler mení údaje v modeli (na základe interpretácie významu akcie používateľa)
- ▶--- view informuje o akcii používateľa (napr. zatlačenie tlačidla myši, posun myši, kliknutie)