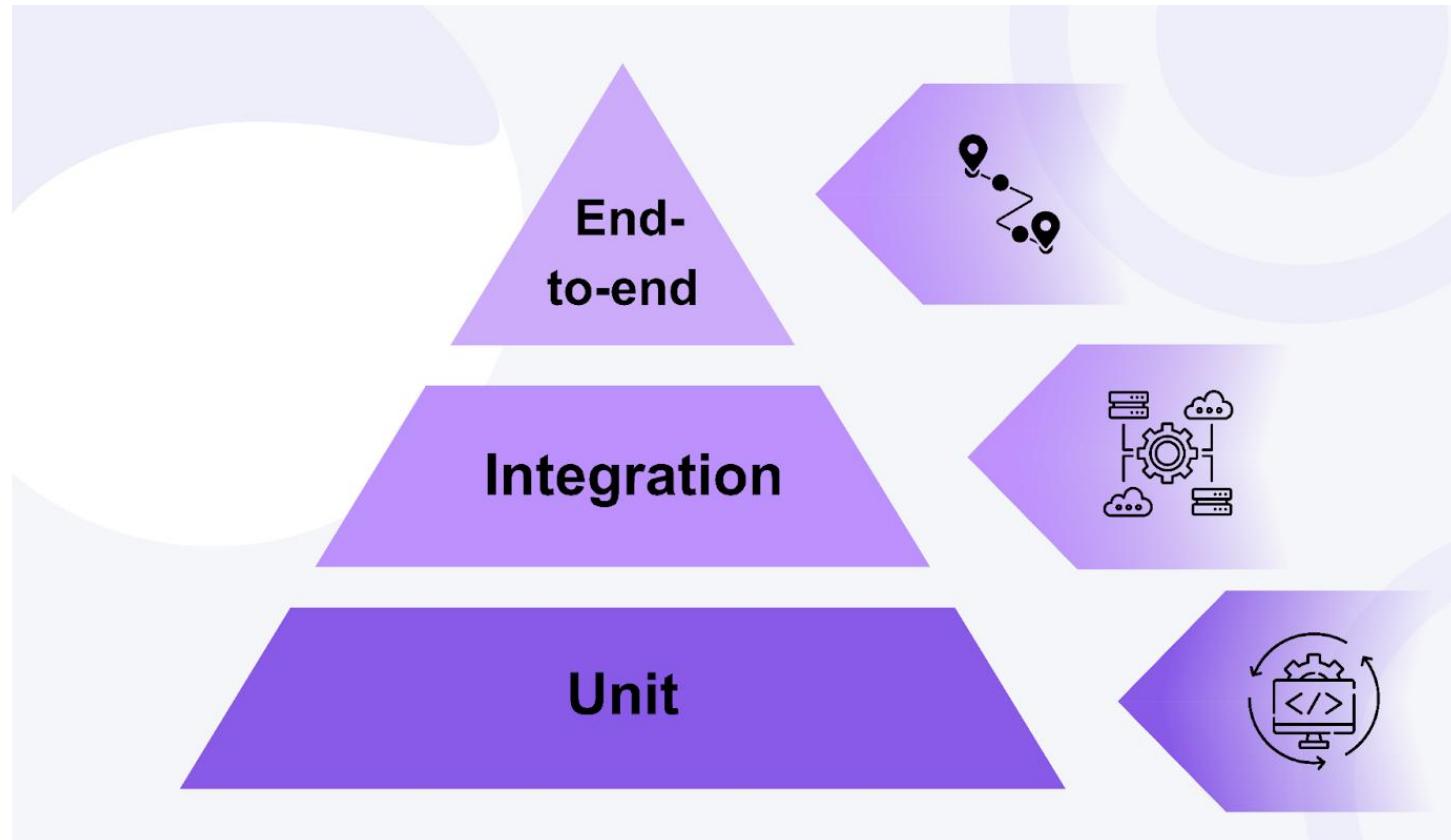


Objektovo Orientované programovanie

5. Prednáška
LS 2024/2025
Juraj Petrík

Testovanie

- Jednotkové
- Integračné
- Funkčné
- Akceptačné
- Výkonnostné
- E2E
- Bezpečnostné



Jednotkové testovanie (JUnit)

- Testujeme jednotky (najmenšie funkčné celky)
- Integrálna súčasť vývoja - píše ich programátor čo implementuje danú funkciu
- Kód musí byť napísaný tak, aby bol testovateľný
- Napr. pri TDD (test driven development) najprv píšem testy, až potom implementujem
- Testujeme logiku, hraničné prípady, exception handling, atď
- Assert once

Základné princípy jednotkového testovania

- Testy majú byť jednoduché
 - Testujte správanie
 - AAA – Arrange, Act, Assert
 - Testy sú deterministické, izolované, nezávislé
 - Dáta majú byť čo najbližšie k realite
-
- 100%-line coverage vs e.g. branch coverage
 - Testovať settery a gettery?

Jednotkové testovanie

- <https://junit.org/junit5/docs/current/user-guide/>
- <https://github.com/DiUS/java-faker>
- <https://site.mockito.org/>
- https://github.com/Jur1cek/jUnit_examples/

GUI

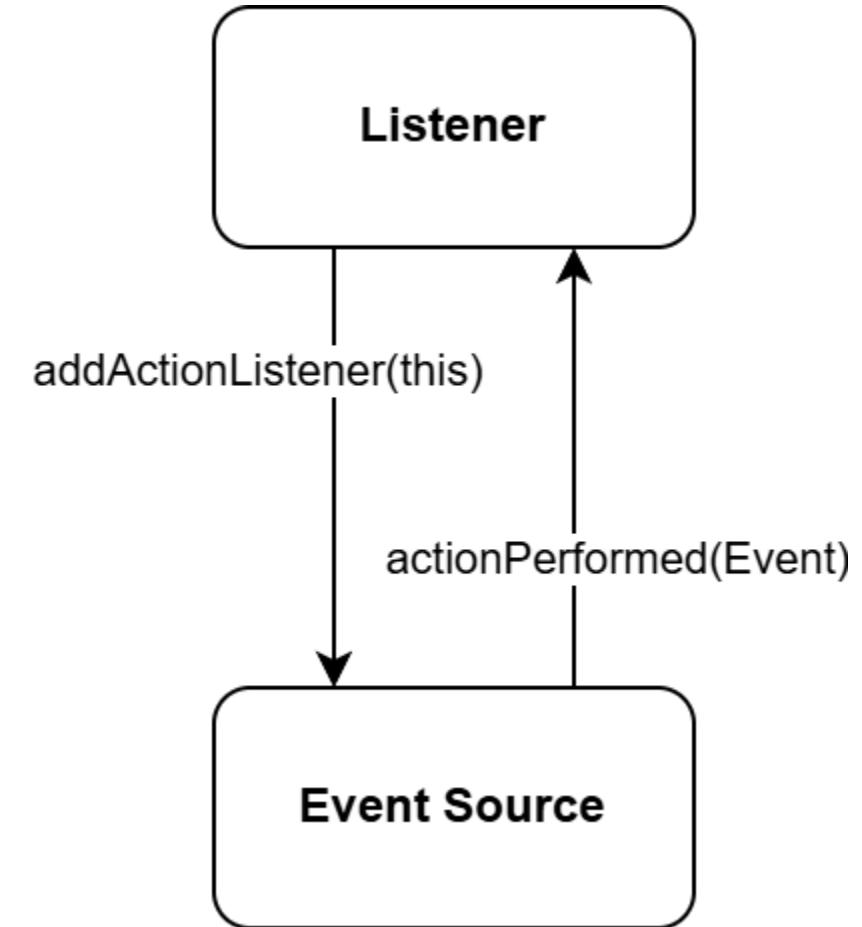
- GUI je vždy oddelené od aplikačnej logiky
- Swing, JavaFX, SWT, atď

Swing

- Priamo v Java
- Lightweight
- Komponentový framework
- MVC pattern:
 - Model – dáda komponentu
 - View – vizuálna reprezentácia dát
 - Controller – používateľský input z view reflektuje do dát modelu

Event Listeners

- ActionListener
- ComponentListener
- ItemListener
- KeyListener
- MouseListener
- WindowListener
- AdjustmentListener
- ContainerListener
- MouseMotionListener
- FocusListener
- ...



Event

- Obsahuje dta o udalosti
- ActionEvent
- ComponentEvent
- ItemEvent
- ...
- getActionCommand()
- getSource()
- getWhen()

Event handling

- V samotnej triede
- V inej triede
- Anonymnou triedou
- Lambda výrazom

Vnútorné triedy

- Trieda v triede
 - Môže používať atribúty „vonkajšej“ triedy, aj privátne
 - „Vnútorný“ objekt je viazaný na „vonkajší“
-
- MyOuter outerObj = new MyOuter();
 - MyOuter.MyInner innerObj = outerObj.new MyInner();

Lambda výraz

- `label.addActionListener(event -> label.setText(„Hello!“));`
- Iba keď používame functional interface (SAM interface) – obsahuje jednu abstraktnú metódu (môže mať aj iné typy)

Layout

- Pridávanie widgetov (komponentov) do framu
 - Kreslenie 2D grafiky
 - Vloženie obrázku
-
- <https://docs.oracle.com/javase/tutorial/uiswing/layout/visual.html>

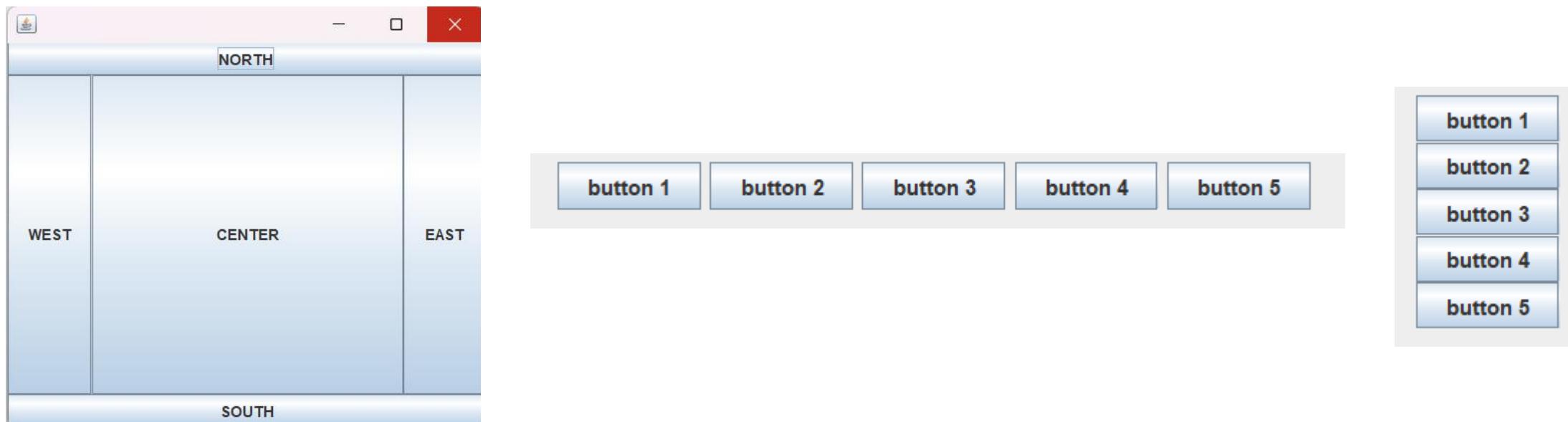


Layout managers

- Komponenty je štandardné možné vnoríť do seba
- BorderLayout – jeden komponent na región, zvyčajne nerešpektuje rozmery komponentov (default pre frame)
- FlowLayout – zľava doprava, rešpektuje rozmery komponentov, wrapping (default pre panel)
- BoxLayout – podobne ako FlowLayout ale zhora nadol, nema automaticky component wrapping
- GridBagLayout -

Layouts

- Pozor na dalšie špecifiká, toto je “default” správanie



BorderLayout

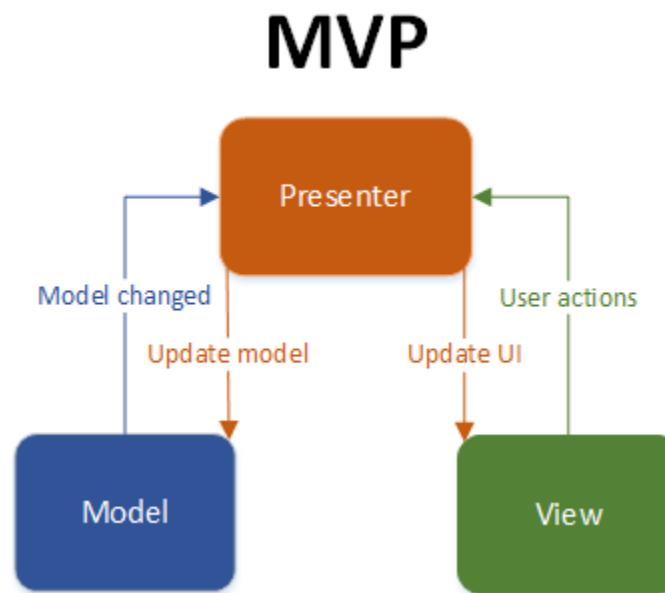
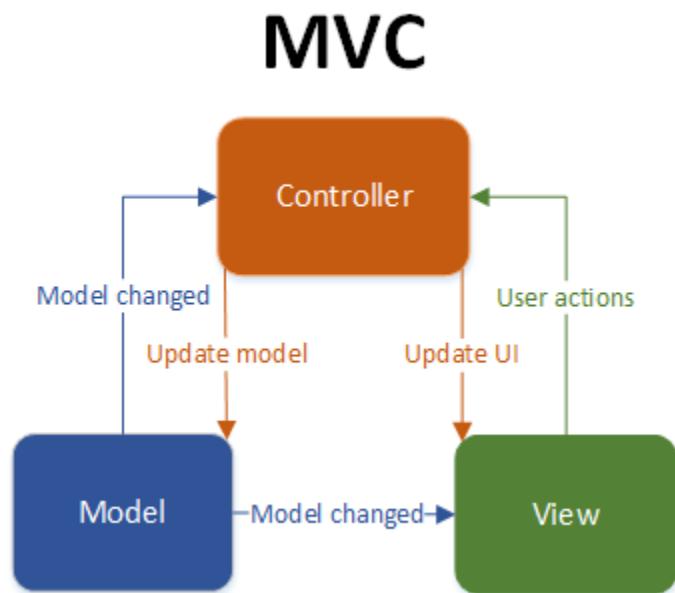
- Ako sa správajú regióny?
- 1 komponent na region
- Ako je to s veľkosťou komponentov?
- It depends..

Komponenty vo Swingu

- Background (JFrame, JPanel)
- Interactive components:
 - JButton
 - JLabel
 - JTextField
 - JTextArea
 - JList

GUI všeobecne

- Čo sa stane keď niečo dlhšie trvá?
- Oddelujte dátu, aplikačnú logiku a prezentáčnu vrstvu
- Používajte vhodný vzor: MVC vs MVP vs MVVM vs ...



Quiz time