

Cvičenia č. 11, úlohy na JavaFX

Nasledujúce úlohy možno využiť na ďalšie precvičenie JavaFX. *Nie sú* určené na odovzdávanie na testovači.

1. Vytvorte JavaFX aplikáciu, ktorá načíta z konzoly prirodzené číslo n a následne zobrazí okno so scénou obsahujúcou n tlačidiel s nápismi zodpovedajúcimi číslam od 1 po n .
Ako koreňovú oblasť scény môžete použiť napríklad `FlowPane`. Načítanie vstupu z konzoly možno realizovať v metóde `start` pred zobrazením hlavného okna aplikácie. Kód je v takom prípade prakticky rovnaký ako pri konzolových aplikáciách.
2. Oživte jednotlivé tlačidlá v predchádzajúcej aplikácii tak, aby každé z nich na konzolu (alebo do textového popisku na scéne) vypisovalo svoje číslo. Implementujte túto funkcionality dvoma spôsobmi:
 - a) So samostatným spracúvateľom udalostí typu `EventHandler` pre každé z tlačidiel.
Tu sa môže zísť cyklus `for` obsahujúci definíciu lambda výrazu prístupujúceho k iteračnej premennej. Niečo také nejde napísať priamo, pretože z lambda výrazov – rovnako ako aj z lokálnych a anonymných tried – definovaných v metóde `f` možno pristupovať iba k tým lokálnym premenným metódy `f`, ktoré sú finálne alebo „v podstate finálne“ (t. j. nie sú nutne označené ako finálne, ale po svojej inicializácii sa už ich hodnoty nemenia). Tento problém je možné obísť viacerými rôznymi spôsobmi. Napríklad možno v tele cyklu definovať novú finálnu premennú inicializovanú na hodnotu iteračnej premennej. Inou možnosťou je použiť namiesto lambda výrazu inštanciu „bežnej“ triedy implementujúcej rozhranie `EventHandler<ActionEvent>`, ktorá dostane hodnotu iteračnej premennej ako argument konštruktora.
 - b) S jediným spracúvateľom udalostí pre všetky tlačidlá. K ovládaciemu prvku, na ktorom udalosť `event` vznikla, možno pristupovať pomocou metódy `event.getSource()`.
3. Vytvorte aplikáciu s tlačidlom `Pridaj`, ktoré bude po stlačení na scéne vytvárať ďalšie tlačidlá s postupne rastúcimi číslami (každé z nich navyše môže svoje číslo vypisovať na konzolu). Skúste na rozloženie tlačidiel použiť rôzne podtriedy triedy `Pane`.
4. Upravte predchádzajúcu aplikáciu tak, aby sa namiesto tlačidiel vytvárali štvorčeky nejakej farby, v ktorých strede bude text s daným číslom (na vytváranie takýchto štvorčekov použite oblasti typu `StackPane`).
5. Vytvorte aplikáciu s jediným tlačidlom štvorcového tvaru umiestneným v oblasti základného typu `Pane`. Po stlačení tlačidla sa jeho veľkosť zväčší o nejaký konštantný faktor. Po stlačení niektorej zo šípok na klávesnici sa zmení poloha tlačidla v rámci scény. Na menenie veľkosti tlačidla použite metódy `setPrefWidth` resp. `setPrefHeight`. Jeho polohu môžete upravovať metódami `setLayoutX` a `setLayoutY`.
6. Napíšte aplikáciu, ktorej okno obsahuje obdĺžnik a tlačidlo, po stlačení ktorého sa vo vnútri obdĺžnika na náhodnej pozícii vykreslí kruh s nejakým pevným polomerom a náhodnou farbou výplne. Kliknutím na ľubovoľný kruh sa navyše tento zo scény odstráni. Pridajte napokon do aplikácie ešte jedno tlačidlo, ktoré náhodne zmení farby všetkých momentálne vykreslených kruhov.

Úlohy k pokročilejšiemu JavaFX

Nasledujúce úlohy sú zamerané na látku spadajúcu do dobrovoľného samoštúdia.

1. Vytvorte aplikáciu, v ktorej sa v textovom popisku zobrazujú rozmery hlavného okna aktualizované kedykoľvek dôjde k ich zmene. Použite vlastnosti `widthProperty()` a `heightProperty()` inštancie triedy `Stage`, ktorým vhodným spôsobom nastavíte „spracúvateľov zmeny“.
2. S použitím previazania vlastností napíšte aplikáciu zobrazujúcu v strede scény kruh, ktorého polomer bude rovný tretine menšieho z rozmerov okna – a to aj po prípadnom menení týchto rozmerov používateľom. Môže sa tu zísť trieda `Bindings` z balíka `javafx.beans.binding`.¹

¹Dajte si pozor na to, aby nedošlo k zámene s rovnomennou triedou z balíka `javax.script`.

3. Napíšte aplikáciu s jednou hlavnou ponukou, do ktorej bude možné stlačením nejakého tlačidla pridávať postupne ďalšie a ďalšie položky, pričom textom každej položky bude jej poradové číslo (ponuka teda bude vždy obsahovať položky od 1 po nejaké prirodzené číslo n). Po kliknutí na každú položku sa táto položka stane neaktívnou.
4. Napíšte aplikáciu s ovládacím prvkom typu `Spinner<Integer>`, v ktorom môže používateľ zvoliť nejaké prirodzené číslo n . Po požiadavke na zavretie okna bude následne používateľ nútený – ak má naozaj dôjsť k zatvoreniu okna a ukončeniu aplikácie – presne n -krát potvrdiť dialóg typu `Alert`. V prípade, že používateľ čo i len raz tento dialóg ukončí inak, než jeho potvrdením, bude aplikácia bežať ďalej rovnako, ako keby k požiadavke na zatvorenie okna vôbec nedošlo.
5. Napíšte aplikáciu s tlačidlom postupne pridávajúcim ďalšie a ďalšie textové polia typu `TextField` spolu s tlačidlom na odobranie daného textového poľa. Pridajte do aplikácie hlavnú ponuku s možnosťami „Otvoriť...“ a „Uložiť ako...“. Po zvolení druhej z týchto možností sa do súboru vybraného v ukladacom dialógu typu `FileChooser` vo vhodnom formáte uloží celkový počet zobrazovaných textových polí a obsah každého z nich. Po zvolení prvej z možností sa tieto informácie zo súboru vybraného v otváracom dialógu typu `FileChooser` načítajú a ovládacie prvky na scéne sa na základe nich aktualizujú.
6. Vytvorte aplikáciu s dvoma tlačidlami a dvoma skupinami typu `ToggleGroup`. Každé z tlačidiel bude do svojej skupiny pridávať nové ovládacie prvky typu `RadioButton` označené napríklad svojimi poradovými číslami. Pre obidve skupiny ďalej do aplikácie pridajte textový popisok typu `Label`, v ktorom sa bude vypisovať text zvoleného tlačidla typu `RadioButton` v danej skupine. Implementujte túto funkcionálnosť dvoma spôsobmi: pridaním spracúvateľov udalostí typu `EventHandler` jednotlivým ovládacím prvkom typu `RadioButton` a nastavením spracúvateľa zmeny typu `ChangeListener` pre vlastnosť `selectedToggleProperty()` inštancií triedy `ToggleGroup`.
7. Vytvorte aplikáciu, ktorá po stlačení tlačidla `btn1` zobrazí nové okno obsahujúce textové pole a tlačidlo `btn2`. Po kliknutí na tlačidlo `btn2` sa nové okno zavrie a nápis tlačidla `btn1` sa zmení na text zadaný používateľom do textového poľa.