Cvičení z předmětu VAW: Úkoly na cvičení

Vypracoval: Leoš Kukačka
březen 2022
leos.kukacka@tul.cz

1. úkol – **Modus**
Prvky ve formuláři: `Button`, `Label`
Napište program, který po stisknutí tlačítka sestaví pole 10–20 integerů (prozatím nastavte čísla ručně přímo v kódu, ale je možné je i náhodně vygenerovat) a v poli najde modus, tj. nejčastěji se vyskytující číslo. Výsledek se vypíše do labelu.

2. úkol – **Trojúhelníková nerovnost**
Prvky ve formuláři: `Button`, 3× `NumericUpDown`
Napište metodu, která ověří trojúhelníkovou nerovnost a další metodu, která ze tří stran spočítá obsah trojúhelníka. Po stisknutí tlačítka program ověří nerovnost a spočítá obsah, je-li splněna.
Zvláštní požadavky: Trojúhelníková nerovnost a obsah mají být počítány ve zvláštních metodách.
Pozn.: Při výpočtech v plovoucí desetinné čárce hrozí při počítání rozdílu dvou hodně blízkých čísel ztráta přesnosti (rozdíl má méně platných číslic, než vstupní hodnoty). Klasický Heronův vzorec tak může být pro extrémně ploché trojúhelníky nepřesný. Lépe použijte:
$$S=\frac{1}{4}\sqrt{\left[a+\left(b+c\right)\right]\left[c-\left(a-b\right)\right]\left[c+\left(a-b\right)\right]\left[a+\left(b-c\right)\right]},a\geq b\geq c.$$
Rozdíl oproti Heronovu vzorci si můžete vyzkoušet (např. $a = b = 10^8$, $c = 10^{-8}$).

3. úkol – **Herní postava**
Prvky ve formuláři: 2× `Button`, `TextBox`, `Labely`
Napište třídu, která reprezentuje herního hrdinu s těmito vlastnostmi: jméno, zdraví a umění boje. Stiskem prvního tlačítka se vygeneruje postava se jménem z TextBoxu a číselnými vlastnosmi:
zdraví: dvakrát hod šestistěnnou kostkou + 12
umění boje: jeden hod šestistěnnou kostkou + 6
Vlastnosti se vypíšou do labelů. Zároveň se vygeneruje protivník s fixními vlastnostmi: umění boje 8 a zdraví 14. Stiskem druhého tlačítka dojde k boji mezi hrdinou a protivníkem – každá strana hodí dvěma kostkami a k výsledku přičte své umění boje. Kdo má vyšší číslo, vyhrál kolo boje a poraženému se odečtou dva životy (zapíše se do příslušného labelu). Opakovaným stiskem druhého tlačítka probíhají další kola souboje, dokud neumře jedna z postav (zdraví klesne na nulu).
Zvláštní požadavky: Hrdina i soupeř jsou reprezentováni instancemi stejné třídy, generování vlastností probíhá v konstruktoru. Měly by být dva konstruktory – jeden pracující s náhodou, a druhý, jenž nastaví fixní hodnoty.
Pozn.: Generátor náhody (instance třídy `Random`) je vhodné mít v celém programu pouze jeden. Je-li potřeba, aby se generovala náhoda uvnitř nějaké třídy, předejte referenci na instanci `Random` jako parametr konstruktoru.
Pozn.: Dva hody šestistěnnou kostkou (náhodné číslo od 1 do 6) není to samé jako náhodné číslo od 2 do 12. Proč?

4. úkol – **Přidávání grafických prvků kódem**
Prvky ve formuláři: `Button`
Po stisknutí tlačítka se na formuláři zobrazí nové tlačítko s číslem 1. Opakovaným stiskem se pak vedle něj přidávají další a další tlačítka se stále vyššími čísly. Stiskem kteréhokoliv z těchto nových tlačítek se zobrazí `MessageBox` s číslem tlačítka, na které bylo kliknuto.

Zvláštní požadavky: Tlačítka se opravdu generují kódem a všechna tlačítka obsluhuje jediná metoda.

5. úkol – **Dědění**
Prvky ve formuláři: Button, 4x RadioButton, Label, NumericUpDown
Navrhněte abstraktní třídu `Utvar`, která představuje obecně jakýkoliv pravidelný n-stěn v trojrozměrném prostoru. Tato obecná třída má v sobě uloženu délku hrany n-stěnu. Třída implementuje rozhraní IUtvar, který předepisuje, že ona i její potomci musí implementovat:
a) protected metodu na výpočet objemu
b) protected getter na výpočet povrchu (read-only)
Dále navrhněte třídy, které reprezentují:
1. krychli
2. dvanáctistěn
3. dvacetistěn
4. kouli (při výpočtech je délka hrany interpretována jako poloměr)
Tyto třídy dědí ze třídy `Utvar`, a proto musí implementovat výše uvedenou metodu a getter, avšak každá jinak. Třídy také přetěžují metodu ToString(), kde bude uveden typ útvaru, objem a povrch.
Dále navrhněte statickou třídu UtvarFactory se statickou veřejnou metodou:
public static IUtvar UtvarFactory(string tvar, double delkahrany) { ... }
Tato metoda vygeneruje jeden z N-stěnů podle RadioButtonu a vrátí jej jako návratovou hodnotu.

Stiskem tlačítka se založí instance jedné ze čtyř tříd podle útvaru vybraného RadioButtonem. Konstruktorem se předá velikost hrany (z NumericUpDownu). V labelu se pak zobrazí typ útvaru, objem a povrch.
Pozn.: třída `Utvar` je abstraktní třídou, která metodu `Objem()` ani getter `Povrch` přímo neimplementuje, avšak vynutí jejich implementaci v potomcích.

6. úkol – **Kolekce a listboxy**
Prvky na formuláři: 3× `Button`, `Textbox`, `NumericUpDown`, 2× `RadioButton`, `Label`, `ListBox`
Navrhněte třídu, která bude reprezentovat osobu (má své jméno), a dále třídu, která reprezentuje zaměstnance nějakého podniku (dědí z osoby, ale navíc má ještě plat). Po stisknutí tlačítka se – podle toho, jaký RadioButton je zaškrtnutý – vytvoří buď instance osoby, nebo instance zaměstnance. V obou případech se pak tato instance uloží do společné kolekce (kolekce je vhodného typu tak, aby šla nabidnovat do ListBoxu). Pomocí dvou tlačítek („vlevo“, „vpravo“) lze kolekci procházet a v Labelu zobrazovat jednotlivé položky (v každém případě jméno, v případě zaměstnance i plat). Paralelně k tomu je celá kolekce nabindovaná v ListBoxu (ListBox je na tuto kolekci přesměrovaný pomocí vlastnosti DataSource).
Pozn.: Pro správné zobrazení položek kolekce v ListBoxu je vhodné přetížit metodu `ToString()` v příslušné třídě.

7. úkol – **Pexeso**
Prvky na formuláři: `NumericUpDown`, `Button, Timer`
Naprogramujte hru pexeso s čísly. Prvkem `NumericUpDown` se vybere velikost hracího pole (v rozumných mezích). Stiskem tlačítka se vygeneruje hrací pole (matice tlačítek, „kartiček“). Po stisknutí první kartičky se zobrazí jí příslušná číslice; po stisknutí druhé kartičky se zobrazí i její číslice, po uplynutí 3 sekund se však obě otočí zpět rubem nahoru (jsou-li číslice rozdílné), případně zmizí (jsou-li stejné).
Zvláštní požadavky: Rozdávání čísel je náhodné. Ošetřete, aby nešly otočit tři kartičky najednou.
Navrhněte a použijte třídu `karticka`, která dědí z Buttonu, avšak lze do ní navíc ještě uložit číslo.

8. úkol – **E-shop**

Prvky na základním formuláři: 4× `Button`, 3× `ListBox`, `Label`
Prvky na druhém formuláři: 2× `Button`, 2× `TextBox`
Naprogramujte jednoduché rozhraní pomyslného e-shopu. V prvním ListBoxu jsou zobrazeny položky zboží které je na skladě, počet dostupných kusů a cena. Ve druhém ListBoxu jsou zobrazení uživatelé e-shopu (jejich jméno). Ve třetím ListBoxu je zobrazený nákupní košík uživatele, který je zrovna rozkliknutý ve druhém ListBoxu. Tento uživatel si může tlačítky přidávat jednotlivé kusy zboží do svého košíku (ze skladu se odečte jedna položka, v košíku se přičte jedna položka), případně zboží z košíku vracet zpět do skladu. Výběrem jiného uživatele se v ListBoxu 3 zobrazí jeho košík. Stiskem třetího tlačítka se zobrazí formulář na zadávání nového uživatele: TextBoxy na jméno a email, Buttony OK a Zrušit. Stiskem čtvrtého tlačítka se zobrazí v Labelu celková hodnota položek v košíku.
Zvláštní požadavky: Každý uživatel si pamatuje svůj košík formou vhodné kolekce. Zboží je řešeno vhodně navrženou třídou. Formulář na zadávání nových uživatelů nejde tlačítkem OK odkliknout pokud jsou TextBoxy prázdné.

9. úkol – **Míček**
Jednoduchá grafická úloha, kdy se malý míček pohybuje přiměřenou rychlostí po formuláři a odráží se od jeho stěn.

10. úkol - **Background worker**
1x Button
Vytvořte program, který bude v backgroundwrokeru počítat číslo pí. Implementujte metodou Monte Carlo:
http://petrfaltus.net/petr-faltus-ludolfovo-cislo-vypocet-pi-na-201-desetinnych-mist.php
Background worker běží donekonečna, jednou za 1e4 iterací udělá update výpočtu v GUI (sleduje se také počet iterací). Tlačítkem lze worker spustit a pak workeru sdělit požadavek na ukončení.
Zajímavost: Jaká je přesnost výpočtu? Co vše ji ovlivňuje? Jaký generátor náhodných čísel se používá v .NET a lze jej změnit?