

Odpovědi píše na zvláštní odpovědní list s vaším jménem a fotografií. Pokud budete odevzdávat více než jeden list s řešením, tak se na 2. a další listy nezapomeňte podepsat. Do zápatí všech listů vždy napište i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

## Otázka č. 1

- ✓ Do DataSetu máme (mimo jiné) načtený řádek X z tabulky Y. Při načítání DataSetu byla hodnota sloupce Z na řádku X rovná 1. Hodnotu sloupce Z na řádku X v DataSetu změníme na hodnotu 2. Než změny v DataSetu uložíme zpět, jiná aplikace modifikuje zdrojovou databázi a do sloupce Z na řádku X uloží hodnotu 3. Co se stane po zavolání metody Update na IDataAdapteru s implicitně vygenerovaným UPDATE dotazem? Detailně vysvětlete, co se bude dít, jaký bude výsledný obsah DataSetu, a jaký bude obsah tabulky v databázi.

[1,5 bodu]

## Otázka č. 2

- ✓ V kontextu technologie ASP.NET popište, co je *viewstate*, a vysvětlete, proč/k čemu se *viewstate* používá.

[1 bod]

## Otázka č. 3

Předpokládejme, že máme metodu a, která pracuje s daty „chráněnými“ resource managerem implementujícím *IEnlistmentNotification*. Dále máme metodu b, která používá *SqlConnection/SqlCommand* pro modifikaci dat v databázi. Obě metody pracují s daty v kontextu aktuální implicitní transakce (ambient transaction) a případnou chybu ve zpracování vracejí ve formě vyhozené výjimky. Doplňte metodu *Process* tak, aby volala metody a a b v kontextu jedné transakce (tj. aby v případě úspěšného provedení a i b došlo ke commitu, v opačném případě k abortu celé transakce [všech side effectů a i b]) – doplňovat smíte pouze do vyznačených míst:

```
void Process(int id) {
```

```
    a(id);
```

```
    b(id);
```

```
}
```

[1,5 bodu]

## Otázka č. 4

V kontextu WPF nebo Silverlight vysvětlete všechny **nevýhody** implementace nějaké vlastnosti jako běžné .NET vlastnosti oproti její implementaci jako *dependency property*. Co to tzv. *dependency property* vlastně je?

[1,5 bodu]

## Otázka č. 5

Následující XAML kód:

```
<Window x:Class="WpfTest.Window1"
        xmlns="http://.../xaml/presentation"
        xmlns:x="http://.../xaml"
        Title="Window1" Height="101" Width="281">
    <Button Margin="10"
```

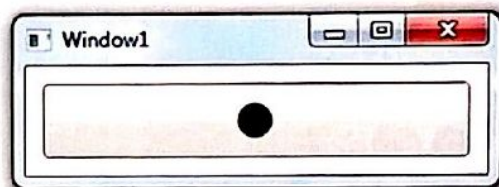
&gt;

```
        <Ellipse Width="20" Height="20"
                Fill="Black" />
```

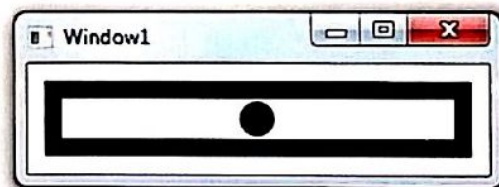
&lt;/Button&gt;

&lt;/Window&gt;

popisuje okno s následujícím vzhledem:



Na volném místě ho doplňte tak, aby výsledný vzhled okna byl tento ("šedá" výplň musí být, min. při spuštění aplikace, stejná jako v systému uživatelem nastavená barva výplně ovládacích prvků - tj. ze C# kódu dostupná jako *SystemColors.ControlBrush*, resp. jako *SystemColors.ControlColor*):



[2 body]

## Otázka č. 6

Vysvětlete všechny výhody zapsání nějakého algoritmu v datových strukturách tvořících workflow z WF (Workflow Foundation), oproti běžnému zapsání stejného algoritmu přímo v jazyce C#.

[1 bod]

## Otázka č. 7

Předpokládejte, že chceme pomocí technologie ASP.NET MVC vyrobit jednoduchou webovou aplikaci, která bude mít dvě stránky: 1) zobrazující seznam všech souborů v nějakém adresáři webového serveru (jméno adresáře je pevně zabudováno v naší aplikaci), 2) po kliknutí na nějaký soubor se zobrazí stránka, kde je pro každý řádek souboru napsána jeho délka ve znacích (počty pro jednotlivé řádky mají být zobrazené na zvláštních řádcích). Navrhněte potřebné *controllery*, *modely*, a *views*.

[1,5 bod]