

## Лабораториска вежба бр.1

### 1. Конзолна апликација во C#

Туристичката агенција „World“ им нуди на своите клиенти патувања до следниве дестинации: Рим, Лондон, Токио, Софија, Истанбул, Торонто, Мајами, Загреб, Белград и Москва. Цената за една карта до одредена зависи од бројот на буквите во името на дестинацијата, помножено со 2000. Пример: карта до Рим би чинела 6000 денари (3 букви \* 2000 денари = 6000 денари).

Бидејќи цените на картите се многу поволни, оваа агенција е многу позната и секојдневно ја посетуваат голем број на клиенти. Па затоа агенцијата се преместила во друга, голема зграда. Во зградата постојат Х (вие одредете колку) шалтери на кои може да се купи карта. Во секој шалтер има по еден вработен кој ги услужува клиентите кои ќе дојдат кај него.

Но, одеднаш почнал да се намалува профитот на агенцијата. Сопственикот забележал дека бројот на клиенти од ден на ден се повеќе се зголемува, но многу малку пари добива на крајот од денот (кога секој вработен го предава профитот од неговиот шалтер). Тогаш на сопственикот му текнало дека неговите вработени не работат чесно.

Ваша задача е да направите софтвер, со кој ќе му помогнете на газдата. Сценариото е: клиент доаѓа на шалтерот Y, со желба да патува до одредена дестинација. Вработениот проверува дали вршат превоз до таа дестинација. Ако да, тогаш го услужува клиентот т.е. му продава карта (со одредена цена). Газдата во секое време може да ги провери картите или заработката на било кој од шалтерите.

Направете мени со следниве опции:

1. Услужи го клиентот (внеси име, презиме, години и пожелна дестинација за клиентот; и број на шалтер)
2. Продадени карти на шалтерот (внеси број на шалтер) (во даден опсег на време)
3. Вкупен промет на шалтерот (внеси број на шалтер) (во даден опсег на време)
4. Сите продадени карти на агенцијата, организирани по шалтер (во даден опсег на време)
5. Вкупен промет на агенцијата
6. Успешност на агенцијата (однос на: бројот на услужени клиенти од сите шалтери и бројот на сите кленти кои дошле во агенцијата)
7. Излез

### **2. Серверски контроли 1**

1. Отворете VisualStudio.
2. Креирајте нов веб сајт кој ќе биде сместен на локалниот податочен систем со програмски код во програмскиот јазик C#.
3. Од менито ToolBox извлекете контрола од типот Label и поставете ја во дизајнерскиот прозорец на страницата Default.aspx
4. Селектирајте ја компонентата и преку менито Properties поставете ги следните својства:  
Име: lblVreme1  
  
Фонт: Arial  
  
Боја на текст: Navy  
  
Големина: Medium  
  
Текст: “Контрола за приказ на време”
5. Извршете ја апликацијата (бидејќи сајтот се компајлира за прв пат, ќе се појави прозорец кој ќе ве праша дали сакате да се компајлира сајтот со дебагирање. Одоговорете потврдно)
6. Откако ќе го добиете резултатот, прекинете го компајлирањето.
7. Во датотеката Default.aspx.cs испрограмирајте го настанот Page\_Load, така што ќе се прикаже времето:  
`lblVreme1.Text = DateTime.Now.ToString();`
8. Извршете ја апликацијата, а потоа неколку пати освежете ја страницата. Што забележувате?
9. Од менито ToolBox извлекете контроли од типот Label и Button и поставете ги во дизајнерскиот прозорец на страницата Default.aspx. Именувајте ги како lblVreme2 и btnVreme, соодветно и додадете им ги истите својства како и на компонентата lblVreme1.
10. Селектирајте ја компонентата btnVreme1, и во подменито Events од менито Properties, изберете го настанот Click, и двојно кликнете во празното поле, десно од името на настанот.

11. Испрограмирајте го настанот Click на копчето btnVreme, така што lblVreme2 ќе го прикаже моменталното време во моментот кога ќе се кликне на копчето.
12. Како се одразува промената на својствата и настаните на компонентата во .aspx кодот?
13. Кодот во Page\_Load заменете го со:  
if (!Page.IsPostBack)  
  
    lblVreme1.Text = DateTime.Now.ToString();
14. Извршете ја апликацијата, и кликнете неколку пати на копчето. Разгледајте го поведението на апликацијата.
15. Прекинете ја апликацијата.
16. Вметнете точки на прекин (Break Points) кај првата наредба од Page\_Load и btnVreme\_Click, и обидете се да го објасните значењето на својството IsPostBack на објектот Page.
17. Креирајте нова страница со име Default2.aspx
18. На страницата Default1.aspx внесете контрола од типот HiperLink.
19. Именувајте ја како hISledno и во својството NavigateUrl изберете ја како локација, штотуку креираната страница Default2.aspx.
20. Во страницата Default2.aspx внесете табела (Layot->Table) и преку изворниот приказ на дизајнерскиот прозорец, центрирајте ја табелата во средина на страницата:  
<table align=center>
21. Во табелата внесете компонента pnlPanela1 од типот Panel и променете и го предефинираниот фонт и позадинска боја.
22. Во компонентата pnlPanela1 внесете две компоненти од типот TextBox со име txtOperand1 и txtOperand2 и по една од типот Label и Button со имиња lblRezultat и btnSoberi, соодветно.
23. Испрограмирајте го копчето btnSoberi со следниот код:  
int op1 = Convert.ToInt32(txtOperand1.Text);

```
int op2 = Convert.ToInt32(txtOperand2.Text);  
lblRezultat.Text = Convert.ToString(op1 + op2);
```

24. Извршете ја апликацијата.
25. За да следното извршување, апликацијата првично ја прикаже страницата Default2.aspx, наместо Default.aspx, во Solution Explorer, лоцирајте ја страницата Default.aspx и со десен клик врз нејзиното име, изберете ја ставката Set as Start Page од менито кое се појавува.
26. Во табелата, надвор од pnlPanela, внесете компонента од типот CheckBox. Именувајте ја како chbVidljivo и на својството Text доделете му вредност “Видливо”.
27. Двојно кликнете врз chbVidljivo и внесете го следниот код:  
if (chbVidljivo.Checked)  
  
    pnlPanela.Visible = true;  
  
else  
  
    pnlPanela.Visible = false;
28. Извршете ја апликацијата. Дали апликацијата го има очекуваното поведење?
29. Поставете го својството AutoPostBack на chbVidljivo на вредност true, и повторно извршете ја апликацијата.
30. Поставете го својството AutoPostBack на txtOperand2 на вредност true, двојно кликнете врз компонентата и во изворното код внесете го кодот од точка 24. Повторно извршете ја апликацијата.
31. Кое е значењето на својството AutoPostBack?
32. Променете го својството EnableViewState на компонентите txtOperand1 и lblRezultat на вредност false.
33. Извршете ја апликацијата, внесете целобројни вредности во текстуалните полиња и неколку пати променете ја селекцијата на chbVidljivo.
34. Кое е значењето на својството EnableViewState?

35. Креирајте страница со име Default3.aspx.
36. Внесете 2 компоненти од типот TextBox со име txtLozinka и txtPoraka, две компоненти од типот Button со име btnProveri и btnPrvaStrana, од кои последната треба да има својство Enabled поставено на вредност false.
37. Поставете го својството TextMode на txtLozinka на вредност Password, а на txtPoraka на вредност MultiLine при што ќе бидат прикажани 5 линии. Поставете го својството ReadOnly на вредност True.
38. Испрограмирајте го настанот Click на копјето btnProveri така што корисникот ќе може да внесе текст во txtPoraka само ако лозинката има вредност “мрежно програмирање”. Ако лозинките се совпаѓаат, потребно е да се постави фокусот во компонентата txtPoraka. Ако лозинките не се совпаѓаат, потребно е да се испише порака за грешка без користење на веб компоненти.
39. Испрограмирајте ја компонентата txtPoraka така што кога истата ќе се напушти, ќе се овозможи употреба на копчето btnPrvaStrana.
40. Копчето btnPrvaStrana да се испрограмира така што кога ќе се притисне, ќе го пренасочи корисникот на страницата Default.aspx.