

# XML

---

12/13

# XML

## ❑ XML - eXtensible Markup Language

- Jezik koji omogućuje logičko strukturiranje podataka
- Općenita specifikacija jezika za opisivanje podataka
  - sintaksa i gramatika za izradu Document Type Definitions (DTD)
  - standard definiran od strane World Wide Web Consortium (W3C)
  - <http://www.w3.org/TR/2000/REC-xml-20001006>
- Struktura i podaci pohranjeni u obliku teksta

## ❑ Metapodaci

- podaci koji opisuju druge podatke, podaci sa semantikom
- za definiranje podatkovnih struktura koriste se oznake (*tags*) slično HTML
  - XML izgleda slično HTML ali nije HTML !
- opisuju se samo podaci ali ne i način njihovog prikaza

## ❑ Primjer: ADO \ DokumentStavka – DokumentStavka.xsd

- pogled u razvojnoj okolini i pogled u uređivaču teksta

# XML oznake

## ❑ Standardne oznake

- početak <oznaka>
- završetak </oznaka> ili />
- prazni element <oznaka/>

## ❑ Elementi

- mogu gnijezditi druge elemente, ali smije postojati samo jedan korijen
- mogu imati pridružene atribute
- mogu, a ne moraju, imati vrijednosti

## ❑ Atributi

- svojstva elemenata

## ❑ Posebne oznake

- Deklaracija dokumenta: <?xml ... ?>
- Komentar: <!-- opaska -->
- Document type declaration (DTD):
  - <!DOCTYPE [ ... ]>
- Definition of document elements in an Internal DTD:
  - <!ELEMENT >, <!ATTLIST>, etc.

```
<?xml version="1.0" encoding="utf-8" ?>
<korijen>
  <element>
    atribut="vrijednost" ...
    <atribut>vrijednost</atribut>
  </element>
</korijen>
```

## ❑ XML razlikuje velika i mala slova u nazivima elemenata!

## ❑ Kako nastane XML (datoteka) ?

# Primjena XML

- Definiranje .NET skupova podataka (*DataSet*)
- Rezultati web servisa, npr. <http://www.webservice.net/globalweather.asmx>
- Prijenos podataka, npr. <http://www.pbz.hr/default.asp?akcija=TL>
- Dokumentiranje programa
- ...

## ☐ SQL naredba koja vraća rezultat u XML formatu

- `SELECT ... FOR XML mode [, XMLDATA] [, ELEMENTS][, BINARY BASE64]`
- mode:
  - RAW (generički redak),
  - AUTO (redak tablice),
  - EXPLICIT (specifikacija za ugniježdene zavisne podatke)
- XMLDATA – rezultat sadrži shemu podataka i podatke
- ELEMENTS – stupci se tretiraju kao podelementi, a ne kao atributi XML čvorova, može se koristiti samo uz AUTO
- BINARY BASE64 – binarni podaci su BASE64 kodirani

## ☐ FOR XML se ne može koristiti u podupitima !

# Jednostavni SQL upiti s FOR XML modifikatorom

## ❑ **SELECT \* FROM Mjesto FOR XML AUTO**

```
<mjesto IdMjesta="132" OznDrzave="HR" NazMjesta="Bast"  
PostBrMjesta="21320" PostNazMjesta="Baška Voda" />  
<mjesto IdMjesta="133" OznDrzave="HR" NazMjesta="Bastajski  
Brđani" PostBrMjesta="43531" PostNazMjesta="Veliki Bastaji" />  
...
```

## ❑ **SELECT \* FROM Mjesto FOR XML RAW**

```
<row IdMjesta="132" OznDrzave="HR" NazMjesta="Bast"  
PostBrMjesta="21320" PostNazMjesta="Baška Voda" />  
<row IdMjesta="133" OznDrzave="HR" NazMjesta="Bastajski Brđani"  
PostBrMjesta="43531" PostNazMjesta="Veliki Bastaji" />  
...
```

# Ugniježdeni elementi u rezultatu spojnog upita

- ❑ `SELECT * FROM Drzava INNER JOIN Mjesto`
- ❑ `ON Drzava.OznDrzave=Mjesto.OznDrzave FOR XML AUTO`

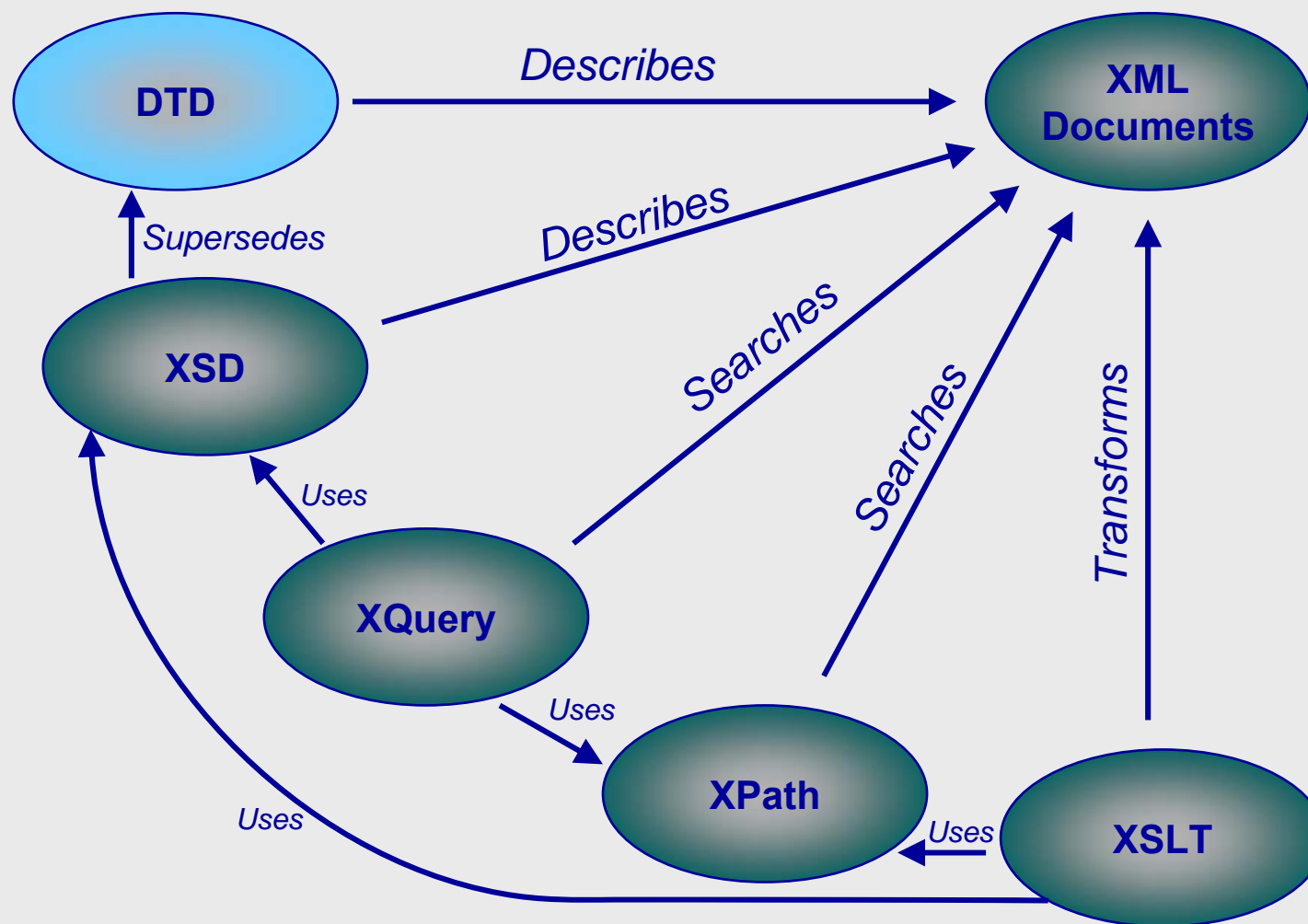
```
<drzava OznDrzave="HR" NazDrzave="Croatia" ISO3Drzave="HRV"  
SifDrzave="191">  
  <mjesto IdMjesta="132" OznDrzave="HR" NazMjesta="Bast"  
PostBrMjesta="21320" PostNazMjesta="Baška Voda" />  
  <mjesto IdMjesta="133" OznDrzave="HR" NazMjesta="Bastajski  
Brđani" PostBrMjesta="43531" PostNazMjesta="Veliki Bastaji" />  
  <mjesto IdMjesta="134" OznDrzave="HR" NazMjesta="Bašana"  
PostBrMjesta="23250" PostNazMjesta="Pag" />  
  ...  
</drzava>
```

# Primjer rezultata sa shemom podataka i podacima

❑ **SELECT \* FROM Mjesto FOR XML AUTO, XMLDATA, ELEMENTS**

```
<Schema name="Schema2" xmlns="urn:schemas-microsoft-com:xml-  
data" xmlns:dt="urn:schemas-microsoft-com:datatypes">  
  <ElementType name="mjesto" content="eltOnly" model="closed"  
order="many">  
    <element type="IdMjesta" />  
    <element type="OznDrzave" />  
    ...  
  </ElementType>  
  <ElementType name="IdMjesta" ... dt:type="i4" />  
  <ElementType name="OznDrzave" ... dt:type="string" />  
</Schema>  
<mjesto xmlns="x-schema:#Schema2">  
  <IdMjesta>132</IdMjesta>  
  <OznDrzave>HR</OznDrzave>  
  <NazMjesta>Bast</NazMjesta>  
</mjesto>  
...
```

# Vrste XML dokumenata





# Obrada XML dokumenata

## ❑ XSD - XML schema

- jezik za opisivanje podataka s vlastitom sintaksom i gramatikom
- za opisivanje podataka koristi se XML → alternativa DTD

## ❑ XPath – jezik za adresiranje dijelova XML dokumenta

- omogućuje selekciju čvorova u XML dokumentu
- koristi se notacija slična onoj za definiranje URL

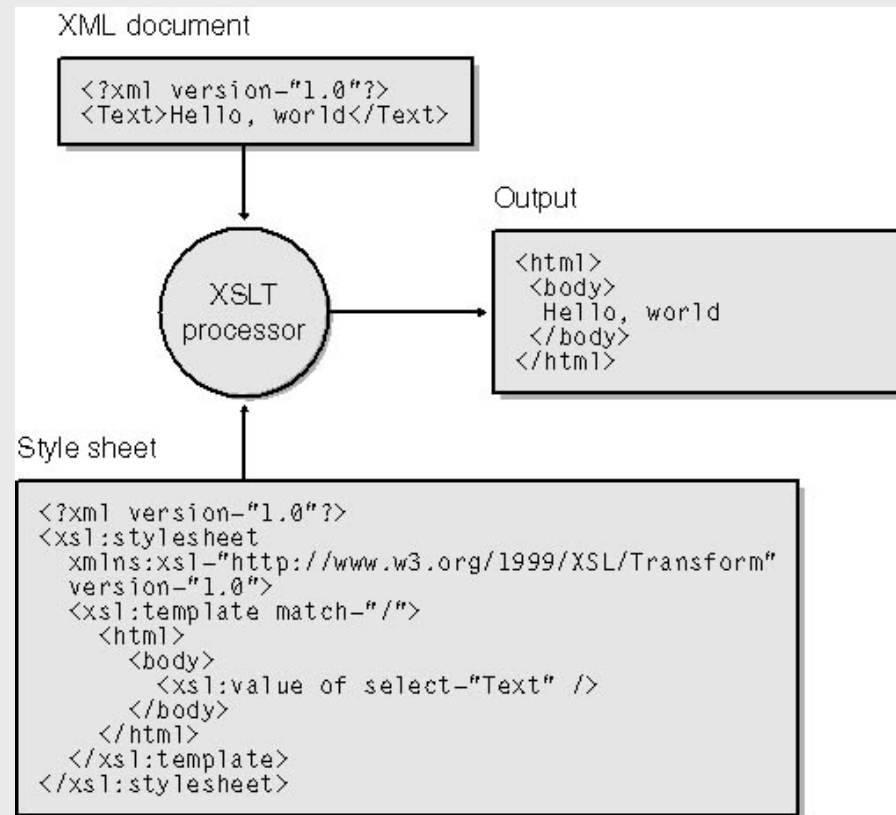
## ❑ XSL - Extensible Stylesheet Language

- jezik za definiranje popisa stilova

## ❑ XSLT - XSL Transformations

- jezik za transformaciju XML dokumenata
- XML programski jezik – ima pravila, izračunava izraze itd.

❑ ...



# Jednostavni tipovi podataka

## ❑ Jednostavni : <simpleType>

### ■ deklaracija

- <simpleType name = "FirstName" type = "string"/>

### ■ ne mogu sadržavati pod-elemente ili attribute

### ■ mogu deklarirati ograničenja

- minLength, maxLength, length, etc

### ■ mogu se koristiti kao osnovni tipovi za složene tipove

## ❑ Primjer

```
<simpleType name = "Ime">  
  <restriction base = "string">  
    <minLength value = "0" />  
    <maxLength value = "25" />  
  </restriction>  
</simpleType>
```

# Složeni tipovi podataka

## ❑ Složeni: <complexType>

- zasnivaju se na jednostavnim tipovima ili na postojećim složenim tipovima
- mogu deklarirati elemente ili reference elemenata
- mogu deklarirati atribut ili reference na grupe podataka

```
<complexType name= "Partner">
  <sequence>
    <element name= "Naziv" type="Name" />
    <element name= "Adresa" type="Address" />
  </sequence>
</complexType>

<complexType name="Address">
  <sequence>
    <element name="Ulica" type="string" />
    <element name="Mjesto" type="string" />
    <element name="Posta" type="string" />
  </sequence>
</complexType>
```

# Kreiranje XML sheme u Visual Studio .NET

## ☐ Kreiranje XML sheme u Visual Studio .NET

- dodavanje sheme: Project / Add New Item / XML Schema
- uređivanje sheme: desni klik, Open ili View
- dodavanje čvorova (Element, Key, Relation, ..)
  - Server Explorer / Data Connections / Connection / Tables / drag
  - ToolBox / XML schema
  - Schema Designer / Add ...

## ☐ Primjer: XML \ TecajnaLista – XMLSchema1

## ☐ Podsjetnik – shema skupa podataka bude generirana automatski

## ☐ Što kad ne postoji generirani skup podataka ?

# Prostor imena *System.Xml*

System. Xml		
. Xsl	. XPath	. Serializati on . Schema
EntityHandling	XmlCharType	XmlNode
Formatting	XmlComment	XmlNodeReader
NameTable	XmlConvert	XmlNodeType
ReadState	XmlDataDocument	XmlNotation
TreePosition	XmlDeclaration	XmlReader
Validation	XmlDocument	XmlSpace
WriteState	XmlDocumentFragment	XmlText
XmlAttribute	XmlDocumentType	XmlTextReader
XmlAttributeCollection	XmlElement	XmlTextWriter
XmlCDataSection	XmlEntity	XmlUrlResolver
XmlCharacterData	XmlEntityReference	XmlWhitespace
	XmlNamedNodeMap	XmlWriter
		...

# Primjer: Tečajna lista

## ❏ Primjer: XML \ TecajnaLista

- ulazna datoteka TecajnaLista.xml
- podaci preuzeti s <http://www.pbz.hr/default.asp?akcija=TL>

Naziv valute	Jedinica	Kupovni za devize	Srednji	Prodajni za devize	Kupovni za efektivu	Prodajni za efektivu
AUD	1	4,351127	4,395078	4,439029	4,320669	4,470102
CAD	1	4,988227	5,038613	5,088999	4,953309	5,124622
CZK	1	0,275401	0,281021	0,286641	0,273473	0,287214
DKK	1	0,975258	0,985109	0,994960	0,968431	1,001925
HUF	100	2,835429	2,893295	2,951161	2,815581	2,957063

Name	Unit	BuyRateCache	BuyRateForeign	MeanRate	SellRateForeign	SellRateCache	Code
AUD	1	4,320669	4,351127	4,395078	4,439029	4,470102	036
CAD	1	4,953309	4,988227	5,038613	5,088999	5,124622	124
CZK	1	0,273473	0,275401	0,281021	0,286641	0,287214	203
DKK	1	0,968431	0,975258	0,985109	0,994960	1,001925	208
HUF	100	2,815581	2,835429	2,893295	2,951161	2,957063	348
JPY	100	4,462883	4,494343	4,558157	4,631088	4,663506	392
NOK	1	0,915993	0,922450	0,931768	0,941086	0,947674	578

# Razredi *XMLDocument* i *XMLNode*

## ❑ Neki postupci *XMLDocument*

- `AppendChild`, `RemoveChild`, `ReplaceChild` – rukovanje čvorovima
- `CreateAttribute` – kreira atribut
- `CreateElement` – kreira element
- `CreateComment` – kreira komentar
- `CreateNode` – kreira instancu razreda *XMLNode*
- `Load` – učitava XML tekst iz navedenog *Stream*, *TextReader* ili *XMLReader*
- `Save` – sprema sadržaj na navedenu lokaciju
- `GetElementsByTagName` – vraća listu čvorova po zadanoj oznaci
- `SelectNodes`, `SelectSingleNode` – traženje čvorova XPath izrazom

## ❑ Neki postupci *XMLNode*

- `AppendChild`, `InsertAfter`, `InsertBefore` – rukovanje čvorovima

## ❑ Neka svojstva *XMLNode*

- `FirstChild`, `LastChild` – prvo i zadnje dijete
- `NextSibling`, `PreviousSibling` – slijedeći i prethodni čvor
- `InnerText` – vrijednost čvora i njegove djece

# Primjer čitanja XML dokumenta

## ❏ Primjer: XML\TecajnaLista

```
listView1.Items.Clear();

XmlDocument tecajnaLista = new XmlDocument();
tecajnaLista.Load(imeDatoteke);
XmlNodeList listaValuta =
    tecajnaLista.GetElementsByTagName("Currency");

foreach (XmlNode cvor in listaValuta)
{
    XmlElement valuta = (XmlElement)cvor;

    XmlNode nazivValute = valuta.GetElementsByTagName("Name")[0];
    XmlNode jedinicaValute = valuta.GetElementsByTagName("Unit")[0];
    ...
    ListViewItem item = new ListViewItem(nazivValute.InnerText);
    item.SubItems.Add(jedinicaValute.InnerText);
    ...
    listView1.Items.Add(item);
}
```



# Kreiranje XML elementa

```
private XmlElement NewElement(  
    XmlDocument xmlDokument, string nazivElementa, string text)  
{  
    XmlElement xmlElement = xmlDokument.CreateElement(nazivElementa);  
    XmlText xmlText = xmlDokument.CreateTextNode(text);  
    xmlElement.AppendChild(xmlText);  
    return xmlElement;  
}
```

Dodavanje nove valute

Naziv valute:

Jedinica:

Kupovni za efektivu:

Kupovni za devize:

Srednji:

Prodajni za devize:

Prodajni za efektivu:

# Rukovanje strukturom XML dokumenta

## ❏ Primjer: XML\TecajnaLista - FormDodajValutu

```
private void buttonDodaj_Click(object sender, EventArgs e)
{
    XmlDocument tecajnaLista = new XmlDocument();
    tecajnaLista.Load(imeDatoteke);

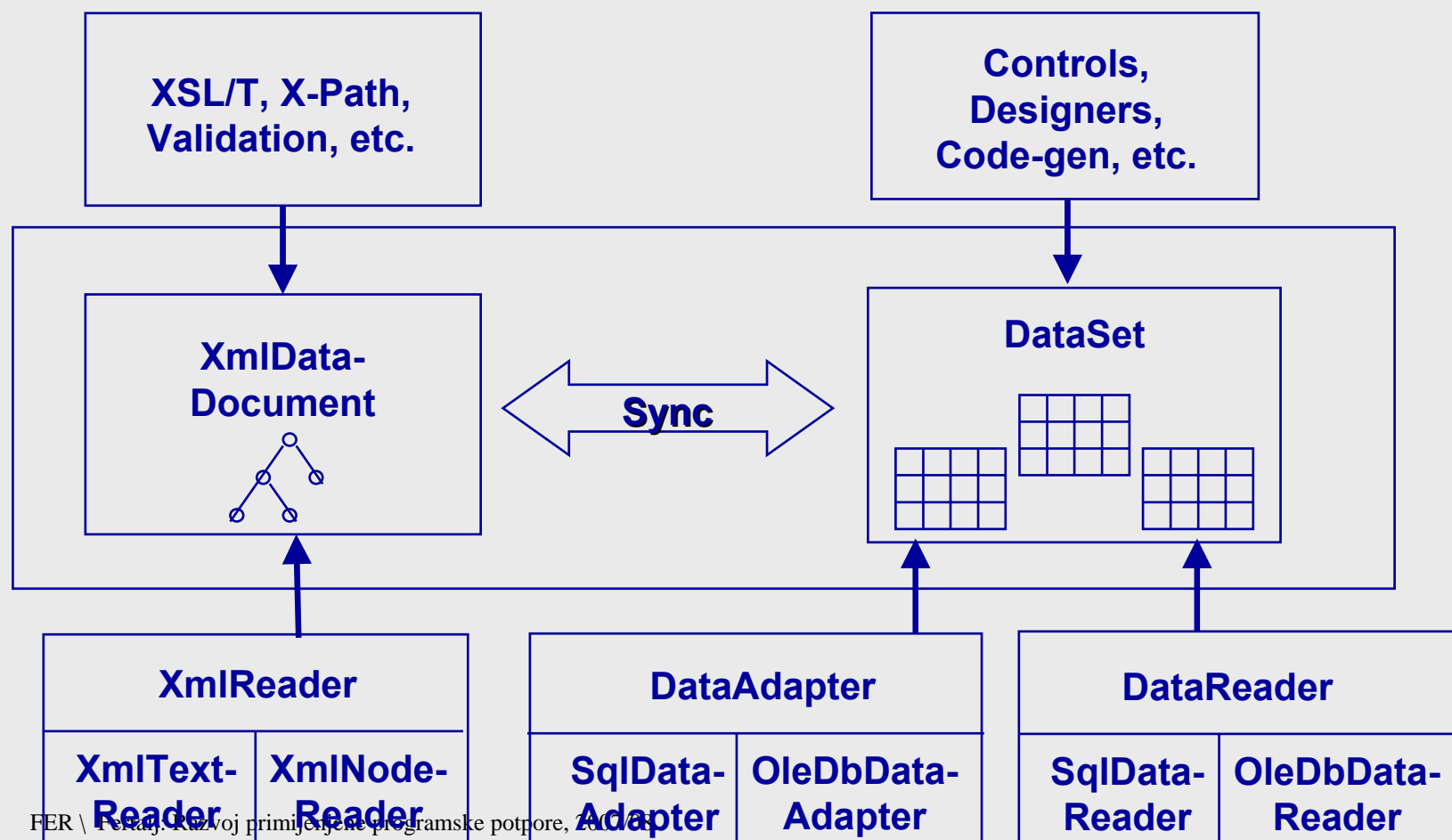
    XmlElement novaValuta = tecajnaLista.CreateElement("Currency");

    XmlElement nazivValute
        = NewElement(tecajnaLista, "Name", textBoxNaziv.Text);
    XmlElement jedinica
        = NewElement(tecajnaLista, "Unit", textBoxJedinica.Text);
    ...
    novaValuta.AppendChild(nazivValute);
    novaValuta.AppendChild(jedinica);
    ...
    tecajnaLista.DocumentElement.FirstChild.AppendChild(novaValuta);
    tecajnaLista.Save(imeDatoteke);
}
```

# XML i DataSet

❑ Razred *XmlDataDocument* – omogućuje čitav XML dokument u memoriji računala koji se može koristiti kao DataSet

- Izveden iz razreda *XmlNode*,
- Svojstvo `DataSet` – relacijska reprezentacija podataka u dokumentu



# ***DataSet* postupci za rad s XML-om**

## **❑ ReadXMLSchema - učitava definiciju iz XSD sheme ili iz XML-a**

- `ReadXMLSchema(stream) // string, TextReader, XmlReader`

## **❑ ReadXML - učitava XML podatke u DataSet**

- `ReadXML(stream, XmlReadMode)`
  - `XmlReadMode` - `Auto`, `ReadSchema`, `IgnoreSchema`, `InferSchema`, `DiffGram`, `Fragment`

## **❑ WriteXMLSchema - zapisuje XML shemu**

- `WriteXMLSchema(stream) // string, TextReader, XmlReader...`

## **❑ WriteXML - zapisuje podatke**

- `WriteXML(stream, XmlWriteMode)`
- `XmlWriteMode` - `IgnoreSchema`, `WriteSchema`, `DiffGram`
  - `IgnoreSchema=False` za veze s postavljenim atributom `Nested=true` podatke zapisuje hijerarhijski

## **❑ InferXMLSchema - izlučuje shemu iz strukture XML podataka**

- `InferXMLSchema(stream) // file, TextReader ili XmlReader`
- korijen bez atributa i podelemenata → `DataSet`, inače `DataTable`
- elementi s atributima i podelementima te ponavljajući → `DataTable`
- atributi i elementi bez atributa i podelemenata → stupci
- za ugniježdene elemente-tablice, automatski se kreira `DataRelation`

## **❑ GetXMLSchema, GetXML - string sheme odnosno podataka**

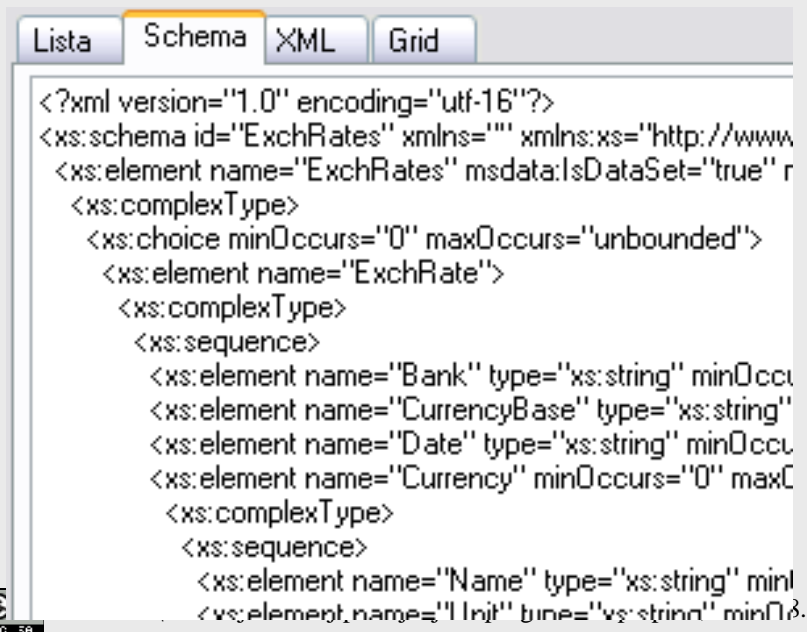
# Učitavanje XML sheme i podataka

## ❏ Primjer: XML \ TecajnaLista

```
XmlDataDocument xmlLista = new XmlDataDocument();
xmlLista.DataSet.ReadXml(
    new StreamReader(imeDatoteke), XmlReadMode.InferSchema);

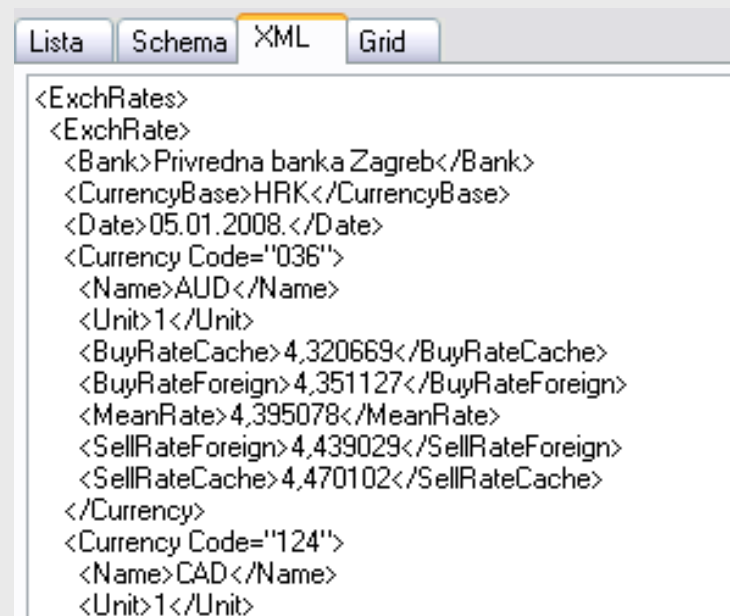
textBoxXMLSchema.Text = xmlLista.DataSet.GetXmlSchema();
textBoxXML.Text = xmlLista.DataSet.GetXml();

dataGridView1.DataSource = xmlLista.DataSet;
dataGridView1.DataMember = xmlLista.DataSet.Tables[1].TableName;
```



The screenshot shows the 'XML' tab selected in a window with tabs 'Lista', 'Schema', 'XML', and 'Grid'. The XML Schema (XSD) is displayed, defining an 'ExchRates' element with a choice of 'ExchRate' elements. The schema includes namespaces for 'ExchRates' and 'xs', and defines various data types like 'string' and 'float'.

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema id="ExchRates" xmlns="" xmlns:xs="http://www
<xs:element name="ExchRates" msdata:IsDataSet="true" r
<xs:complexType>
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="ExchRate">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Bank" type="xs:string" minOccurs="0" maxOccurs="1" />
          <xs:element name="CurrencyBase" type="xs:string" minOccurs="0" maxOccurs="1" />
          <xs:element name="Date" type="xs:string" minOccurs="0" maxOccurs="1" />
          <xs:element name="Currency" minOccurs="0" maxOccurs="1" />
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name" type="xs:string" minOccurs="0" maxOccurs="1" />
              <xs:element name="Unit" type="xs:string" minOccurs="0" maxOccurs="1" />
            </xs:sequence>
          </xs:complexType>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```



The screenshot shows the 'XML' tab selected in a window with tabs 'Lista', 'Schema', 'XML', and 'Grid'. The XML data is displayed, showing two 'ExchRate' elements. The first element is for AUD (Australian Dollar) with a buy rate of 4.320669 and a sell rate of 4.439029. The second element is for CAD (Canadian Dollar) with a buy rate of 4.351127 and a sell rate of 4.470102.

```
<ExchRates>
  <ExchRate>
    <Bank>Privredna banka Zagreb</Bank>
    <CurrencyBase>HRK</CurrencyBase>
    <Date>05.01.2008.</Date>
    <Currency Code="036">
      <Name>AUD</Name>
      <Unit>1</Unit>
      <BuyRateCache>4,320669</BuyRateCache>
      <BuyRateForeign>4,351127</BuyRateForeign>
      <MeanRate>4,395078</MeanRate>
      <SellRateForeign>4,439029</SellRateForeign>
      <SellRateCache>4,470102</SellRateCache>
    </Currency>
    <Currency Code="124">
      <Name>CAD</Name>
      <Unit>1</Unit>
    </Currency>
  </ExchRate>
</ExchRates>
```

# Zapisivanje XML sheme i podataka

## ❑ Primjer: XML \ TecajnaLista

```
private void buttonSpremi_Click(object sender, EventArgs e)
{
    ((DataSet)dataGridView1.DataSource).WriteXml(
        @"c:\projects\tecajna.xsd", XmlWriteMode.WriteSchema);
    ((DataSet)dataGridView1.DataSource).WriteXml(
        @"c:\projects\tecajna.xml", XmlWriteMode.IgnoreSchema);
    MessageBox.Show(@"c:\projects\tecajna.* Ok");
}
```

# Razredi *XmlReader* i *XmlWriter*

## ❑ *XmlReader*

- apstraktni razred za čitanje XML,
- forward-only, non-cached XML stream reader
- svojstva
  - Value: vrijednost čvora
  - NodeType: tip čvora
  - HasValue: oznaka da čvor ima vrijednost
  - LocalName: naziv čvora bez prefiksa
  - ReadState: Closed, EndOfFile, Error, ...

## ❑ *XMLWriter*

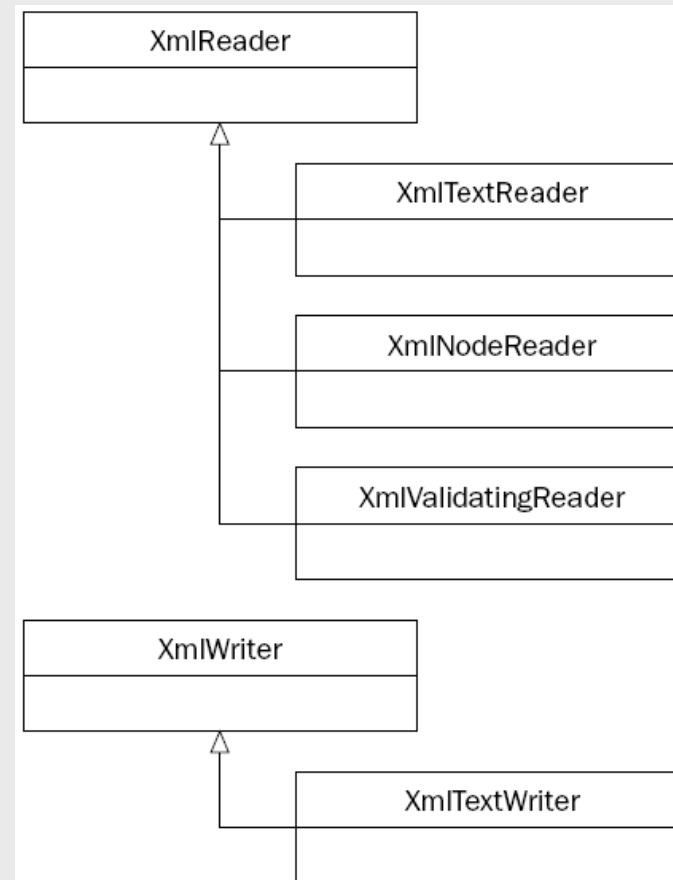
- apstraktni razred za pisanje XML
- forward-only, non-cached XML stream writer
- svojstva
  - WriteState: Attribute, Content, Element, ...
  - XmlLang: vraća aktualni xml:lang
  - XmlSpace: vraća aktualni xml:space

## ❑ *XmlTextReader* – izveden iz *XMLReader*

- postupci: Read, MoveToElement, ReadString

## ❑ *XmlTextWriter* – izveden iz *XMLReader*

- postupci: WriteDocType, WriteComment, WriteName



# Primjer zasebnog učitavanja sheme i podataka

## ❑ Primjer za *XMLReader* i *XMLDataDocument.Load*

```
// Create an XmlDataDocument.  
XmlDataDocument doc = new XmlDataDocument();  
  
// Load the schema file.  
doc.DataSet.ReadXmlSchema("tecajna.xsd");  
  
// Load the XML data.  
XmlTextReader reader = new XmlTextReader("tecajna.xml");  
reader.MoveToContent(); // Moves the reader to the root node.  
doc.Load(reader);
```



# Reference

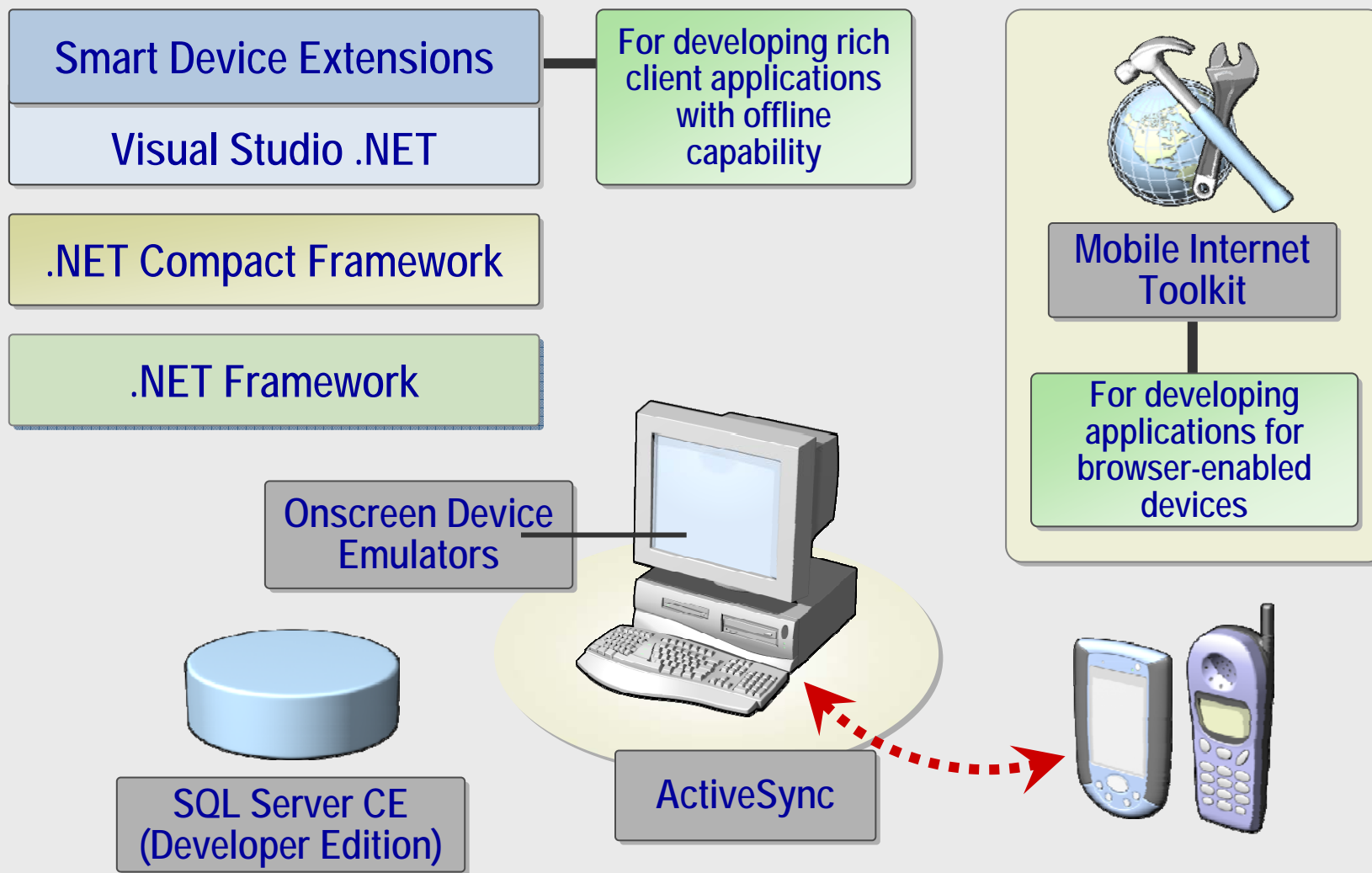
- ❑ <http://www.w3.org/XML/>
- ❑ <http://www.w3schools.com/xml/>
- ❑ <http://msdn.microsoft.com/xml/>
- ❑ <http://www.xml.com/>
- ❑ <http://www.xmlfiles.com/xml/>

# Mobilne aplikacije

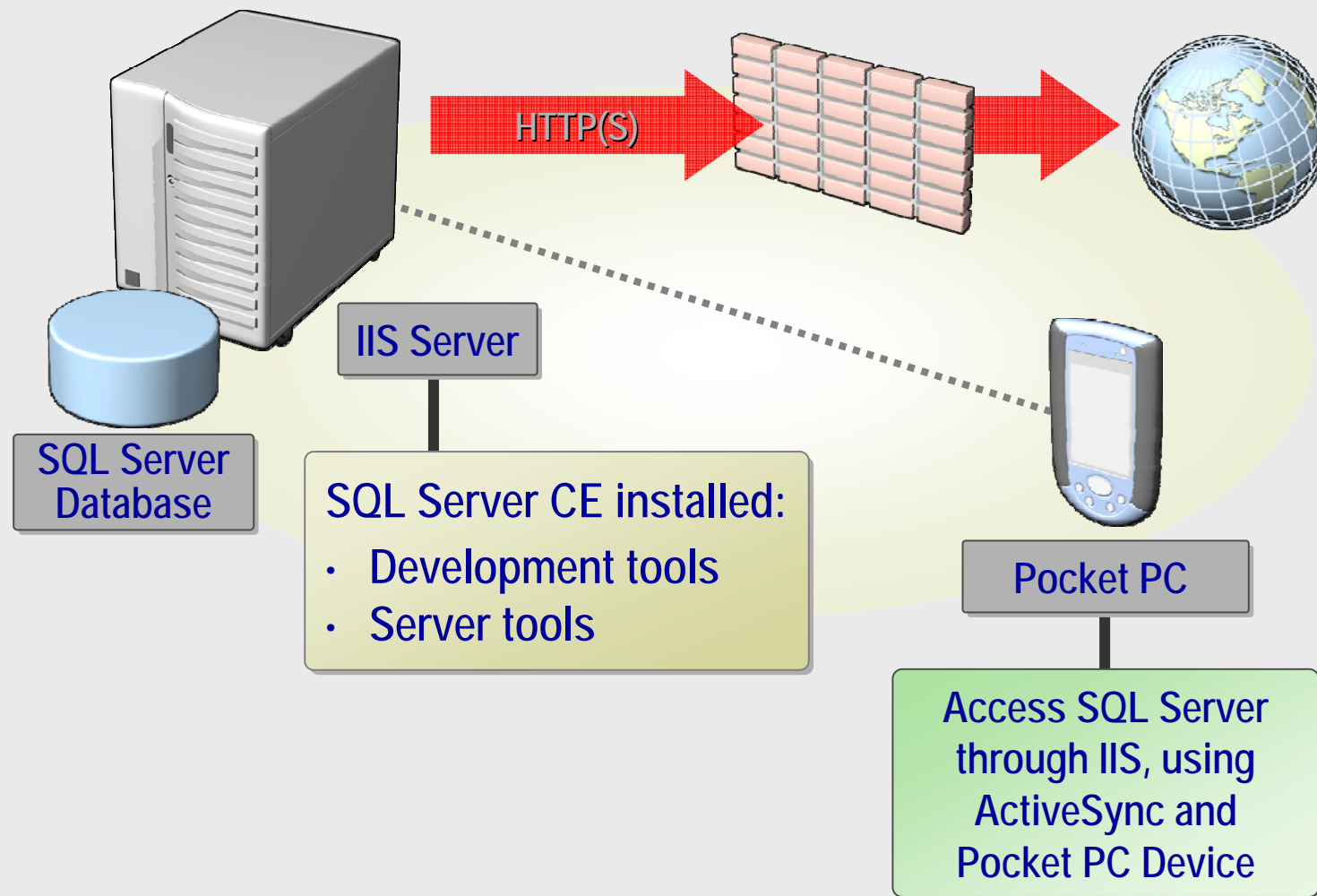
---

12/13

# Alati za razvoj mobilnih aplikacija



# Kontekst rada mobilnih aplikacija



# Ključne karakteristike mobilnih aplikacija

## ☐ Vrijeme pokretanja

- Mobilne aplikacije koriste se učestalo, te u kratkim vremenskim razmacima
- Važno da je aplikacija što brže raspoloživa za uporabu

## ☐ Vrijeme odziva

- Pri interakciji aplikacija treba brzo reagirati na korisnikov zahtjev.
- Važno je korisniku dati do znanja da se njegov zahtjev obrađuje (promjenom ekrana, ispisivanjem poruke, postavljanjem kursora u čekanje)

## ☐ Jasno određena usluga

- Usluga koju aplikacija pruža mora biti što jasnije definirana i lako pristupačna

## ☐ Podešena komunikacija s izvorom informacija

- Smanjena brzina prijenosa podataka na mobilnim uređajima
- Prijenos samo bitnih informacija, podešavanjem filtara i načina komunikacije

## ☐ Konzistentnost i prikladnost

- Uobičajeno zahtijevana konzistentnost korisničkog sučelja mora dodatno biti provedena sukladno mogućnostima uređaja.

## ☐ Ograničeni kapacitet uređaja – prostorni, memorijski, procesni

# Oblikovanje korisničkog sučelja mobilne aplikacije

## ❑ Ključne pretpostavke

- Kreativnost - potrebna zbog limitirane veličine zaslona ručnih računala.
- Disciplina - potrebna za održanje konzistentnosti odabranom modelu, odnosno dizajnu korisničkog sučelja.

## ❑ Ekran osjetljiv na dodir

- važno je minimizirati broj promjena rasporeda – ukoliko treba proći nekoliko koraka unosa, treba nastojati omogućiti da se unos obavi uzastopnim pritiskanjem tipke na istom mjestu, odgovarajući broj puta.

## ❑ Veličina zaslona i navigacija

- zbog veličine uređaja te načina držanja i unosa, dio informacija ostaje prekriven, pa treba posebno pripaziti na raspored kontrola i navigaciju

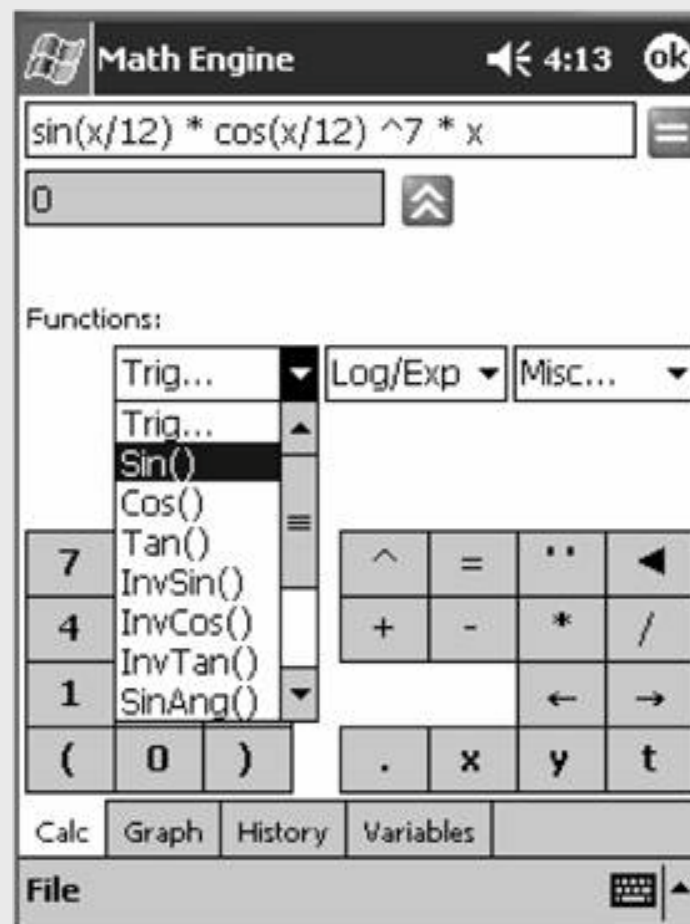
## ❑ Važnost velikih kontrola

- male i neprecizne kontrole frustriraju korisnika
- pri testiranju se koristi emulator pri čemu se programer smije služiti mišem, pa se stječe pogrešan dojam o lakoći korištenja
- realna interakcija mora biti moguća pod različitim kutevima pristupa uređaju

# Oslonac na ugrađene kontrole

- ❑ Treba se osloniti na promišljene i dokazane kontrole
- ❑ Ugrađene kontrole su univerzalne pa zato ne nužno i najbolje za sve primjene

January 1900						
S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10
Today: 1/18/07						



# Grafičko sučelje mobilnih uređaja

## ❑ Sustav direktorija na mobilnom uređaju

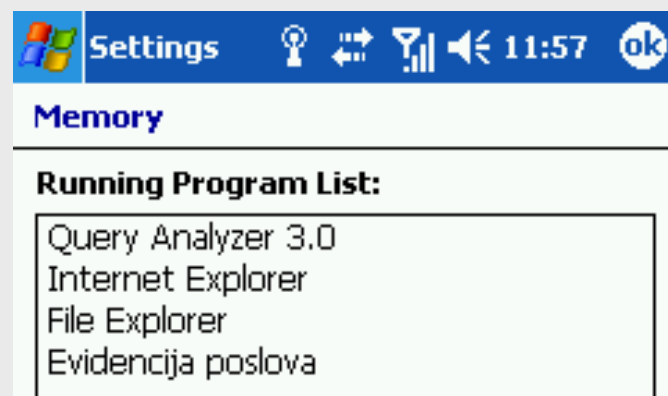
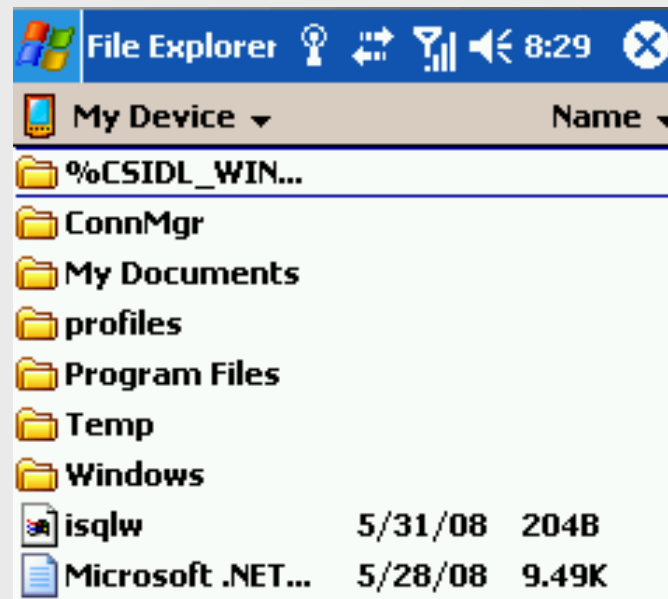
- *My Device* – korijen (kao *My Computer*)
- *Program Files*
  - Direktorij u koji se podrazumijevano kopira aplikacija prilikom objavljivanja

## ❑ Aktivne aplikacije

- Ako se aplikacija nije zatvorila, ona ostaje aktivna
- Pregled aktivnih aplikacija: *Settings* → *System* → *Memory* → *Running Programs*

## ❑ Za unos se najčešće koristi posebna pisaljka (stylus) za pisanje po zaslonu osjetljivom na dodir

- klik = tap
- desni klik = tap & hold (zadržani klik)





# Model pohrane podataka

## ❑ Datoteke

- Ako aplikacija koristi mali broj podatka vjerojatno je pametnije podatke spremati u datoteku.
- Ako su podaci namijenjeni za komunikaciju i razmjenu informacija s drugim uređajima XML je vjerojatno najbolja odluka.

## ❑ Baze podataka

- Ako aplikacija koristi velik broj podataka nad kojima je potrebno postavljanje upita, obavljanje transakcija i sinkronizacija podataka, baza podataka je najbolje rješenje.
- Glavni nedostatak baza podataka je njihova dodatna veličina jer sadrže podatke o podacima i dodatne mehanizme implementacije.

## ❑ Lokalna ili udaljena baza podataka

- Ako aplikacija koristi veću količinu podataka vjerojatno je pametnije bazu imati na samom uređaju.
- Ukoliko baza podataka nije na uređaju treba ugraditi i komunikaciju s poslužiteljem za razmjenu podataka (web servis ili replikacija).

# Compact Framework

## ❑ .NET Compact Framework (.NET CF)

- Verzija *.NET Frameworka* prilagođena za mobilne uređaje
- Uz knjižnice *.NET Frameworka*, definira i specifične za mobilne uređaje
- Mobilni uređaj koji želi pokrenuti aplikaciju napisanu u *Compact Frameworku* mora naravno imati instaliran *.NET Compact Framework*

## ❑ Posebnosti CF

- Prostor imena za interaktivne aplikacije je *Microsoft.WindowsCE.Forms*
- Alfanički unos obavlja se posebnom kontrolom - *InputPanel*

## ❑ Razlike između *.NET* i *.NET CF*

- CF podržava većinu kontrola osim *CheckListBox*, *PageSetupDialog*, *PrintDialog*, *PrintDocument*, *PrintPreviewControl*, *PrintPreviewDialog*, *RichTextBox*, ...
- CF ne podržava izmjene u *GridView* kontroli, MDI, Drag&Drop, GDI+
- CF ne podržava ASP.NET, OleDb

# Ostale komponente

## ❑ MS ActiveSync ili Device Center

- omogućuje sinkronizaciju između stolnog i ručnog računala
- nije obavezan ali ga se preporuča imati (radi povezivanja s mobilnim uređajem iz VS)
- <http://www.microsoft.com/windowsmobile/activesync/activesync45.mspix>
- <http://www.microsoft.com/windowsmobile/devicecenter.mspix>

## ❑ Visual Studio i emulatori mobilnih uređaja

- Tools \ Connect to Device
- Tools \ Device Emulator Manager – Connect, Craddle

## ❑ Emulator

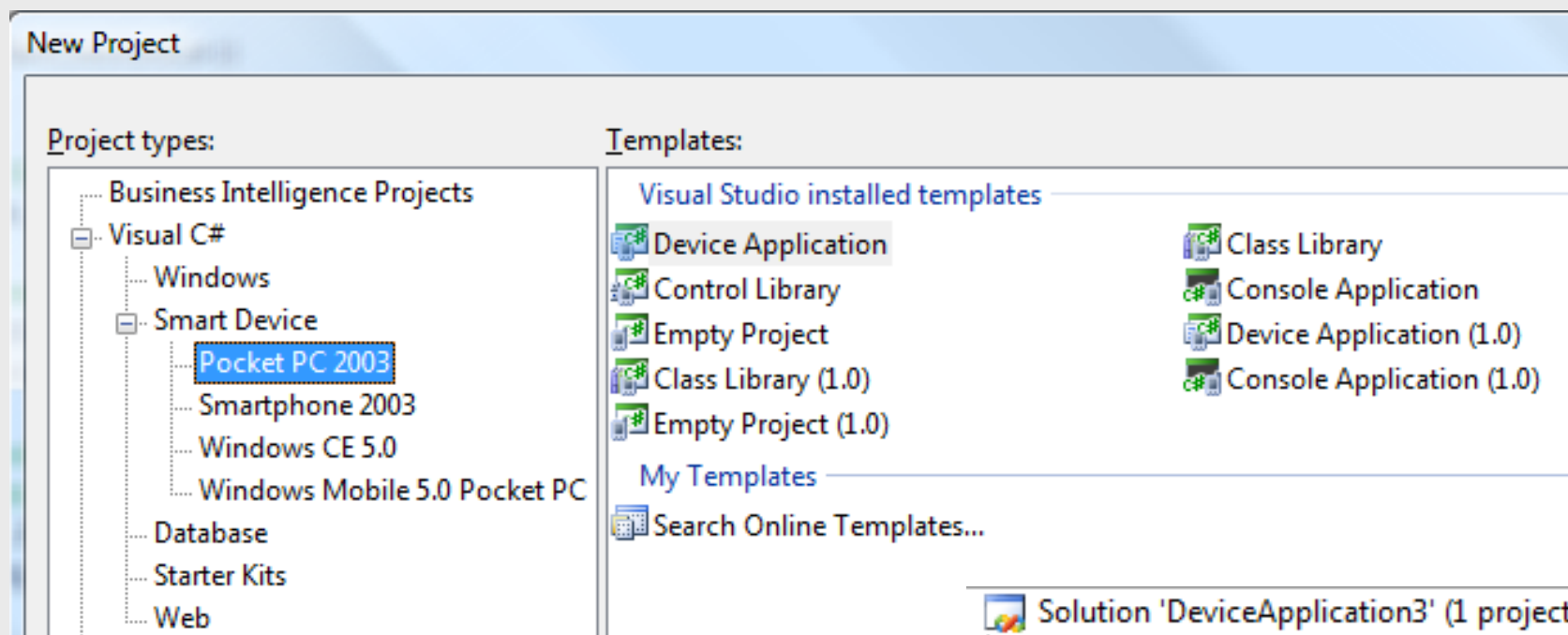
- Traka s alatima \ Device Options \ (emulator) \ Emulator Options - bind to Connected network Card
- Prethodno instalirati Virtual PC 2007
  - Prilagodnik (zapravo *driver*) za virtualno mrežno okruženje

## ❑ Loopback adapter – konekcija neumreženog računala

- <http://support.microsoft.com/kb/839013>

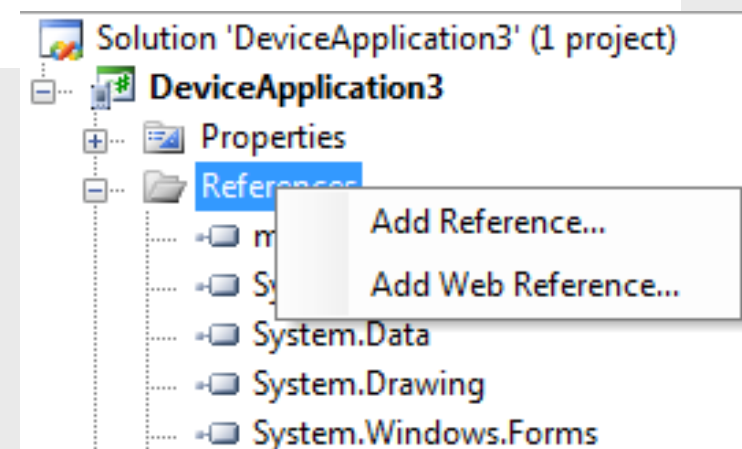
# Stvaranje projekta za mobilne uređaje

## ❑ *File->New...->Project – Smart Device*



## ❑ U reference stvorenog projekta potrebno je dodati

- `Microsoft.WindowsCE.Forms`
- `System.Data.SqlServerCe`
  - Za rad s bazom na mobilnom uređaju



# Standardna mobilna aplikacija

## □ Primjer

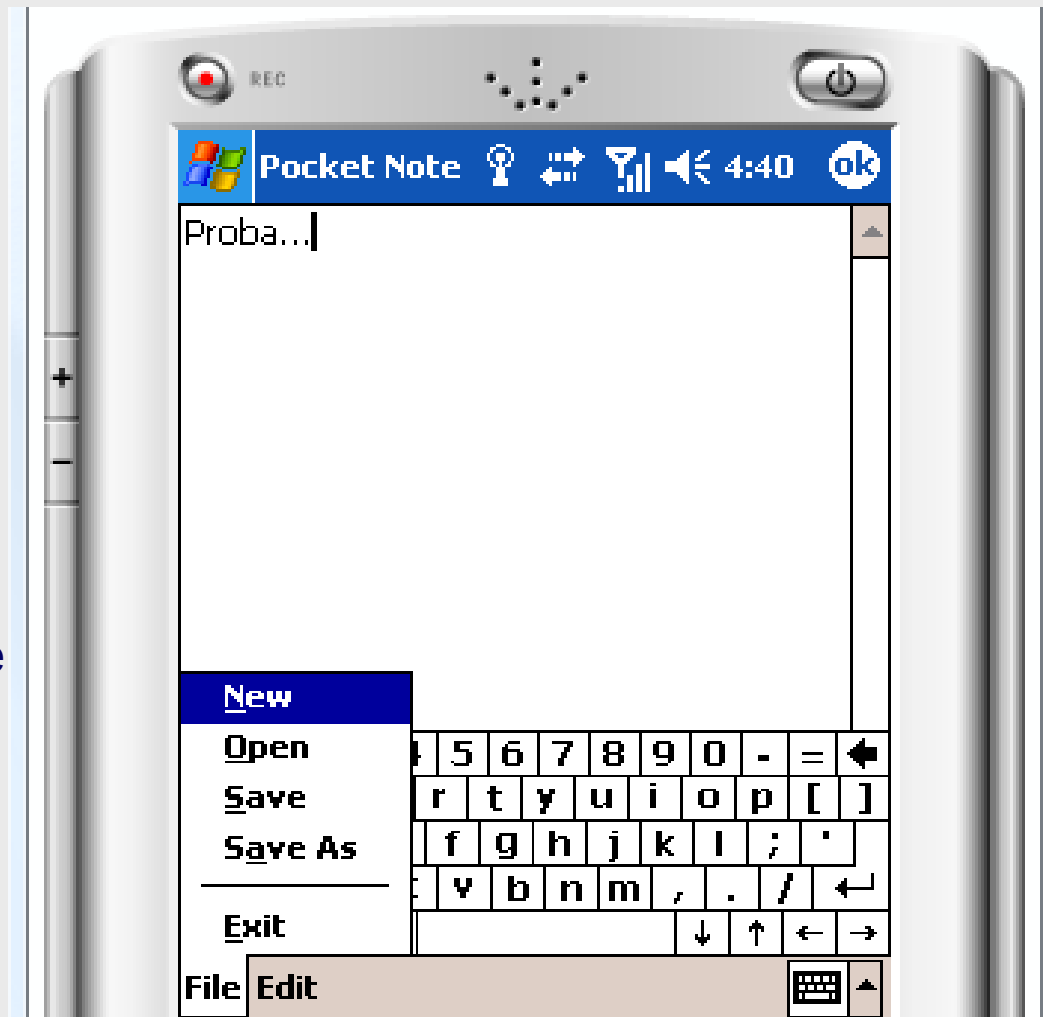
📁 Mobilne/PocketNotepad

## □ Funkcije uobičajenog programa

- Pregled, uređivanje i spremanje tekstovnog dokumenta

## □ Testiranje aplikacije

- radi bržeg izvođenja, testiranje stabilne aplikacije u emulatoru bolje je pokretati sa *Start without debugging*
- u slučaju više projekata potvrditi ugradnju (Deploy) samo željenog programa



# Prostor imena `Microsoft.WindowsCE.Forms`

## ❑ Prostor imena `Microsoft.WindowsCE.Forms`

- Sadrži razrede za programiranje aplikacija na mobilnim uređajima (koristeći .NET Compact Framework)

## ❑ Kontrola `InputPanel`

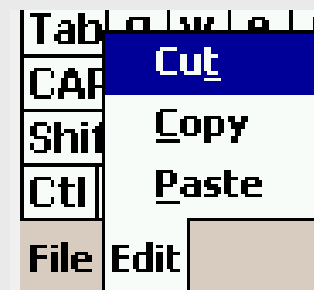
- `Microsoft.WindowsCE.Forms.InputPanel`
- Sučelje između korisnika i SIP na mobilnom uređaju
- SIP (*Soft Input Panel*)
  - služi za jednostavniji unos podataka u mobilni uređaj
  - najčešće preko virtualne tipkovnice
- Neka svojstva i događaji:
  - `Enabled` – određuje vidljivost SIP
  - `CurrentInputMethod` – `Keyboard`, `Letter Recognizer`, `Block Recognizer`
  - Događaj `EnabledChanged` – okida se pri promjeni vidljivosti SIP

# Kontrola InputPanel

## ❑ Prikaz kontrole `inputPanel` ovisno o fokusu

```
this.textBox.GotFocus +=  
    new System.EventHandler(this.textBox_GotFocus);  
this.textBox.LostFocus +=  
    new System.EventHandler(this.textBox_LostFocus);  
  
private void textBox_GotFocus(object sender, System.EventArgs e)  
{  
    inputPanel.Enabled = true;  
}  
private void textBox_LostFocus(object sender, System.EventArgs e)  
{  
    inputPanel.Enabled = false;  
}
```

# Uređivanje teksta



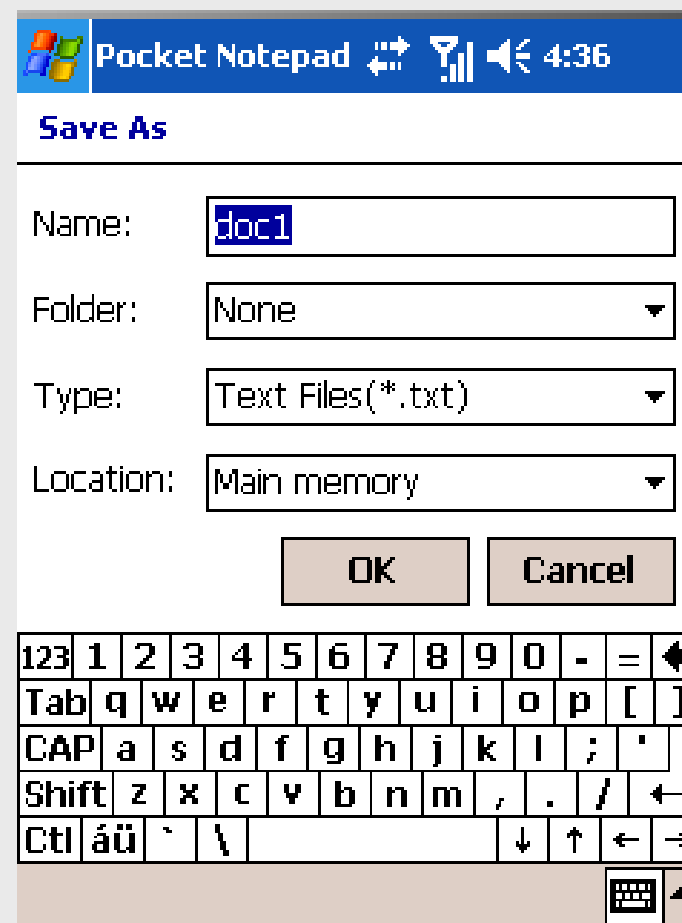
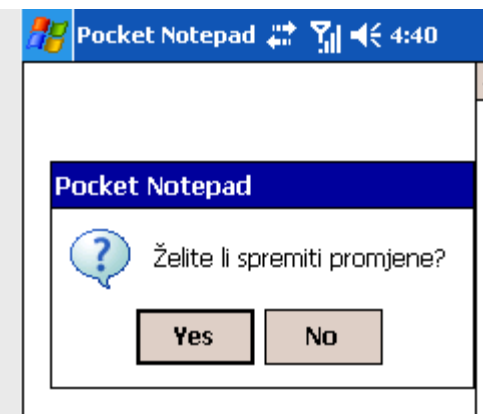
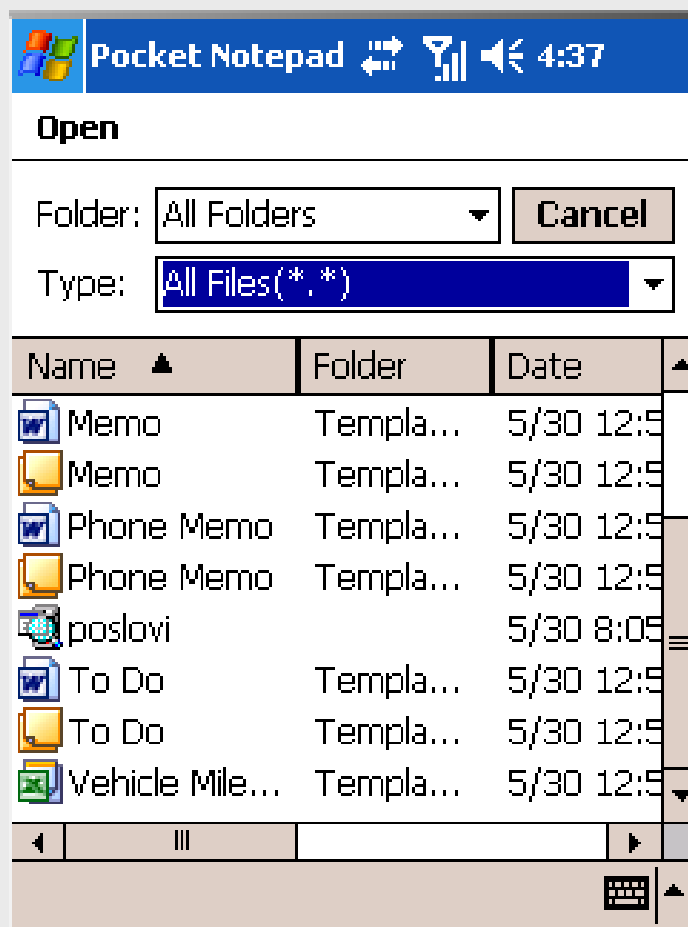
## ❑ Postupci za obradu događaja *cut*, *copy*, *paste*

- Klik na element izbornika (*cut*, *copy*, *paste*)
- Kao privremeni spremnik označenog teksta koristi se član forme
  - *private string clipboard = ""*;

```
private void mnuCut_Click(object sender, System.EventArgs e)
{
    clipboard = textBox.SelectedText;
    textBox.Text = textBox.Text.Replace(textBox.SelectedText, "");
}
private void mnuCopy_Click(object sender, System.EventArgs e)
{
    clipboard = textBox.SelectedText;
}
private void mnuPaste_Click(object sender, System.EventArgs e)
{
    textBox.SelectedText = clipboard;
    textBox.ScrollToCaret();
}
```



# Dijalozi na mobilnom uređaju

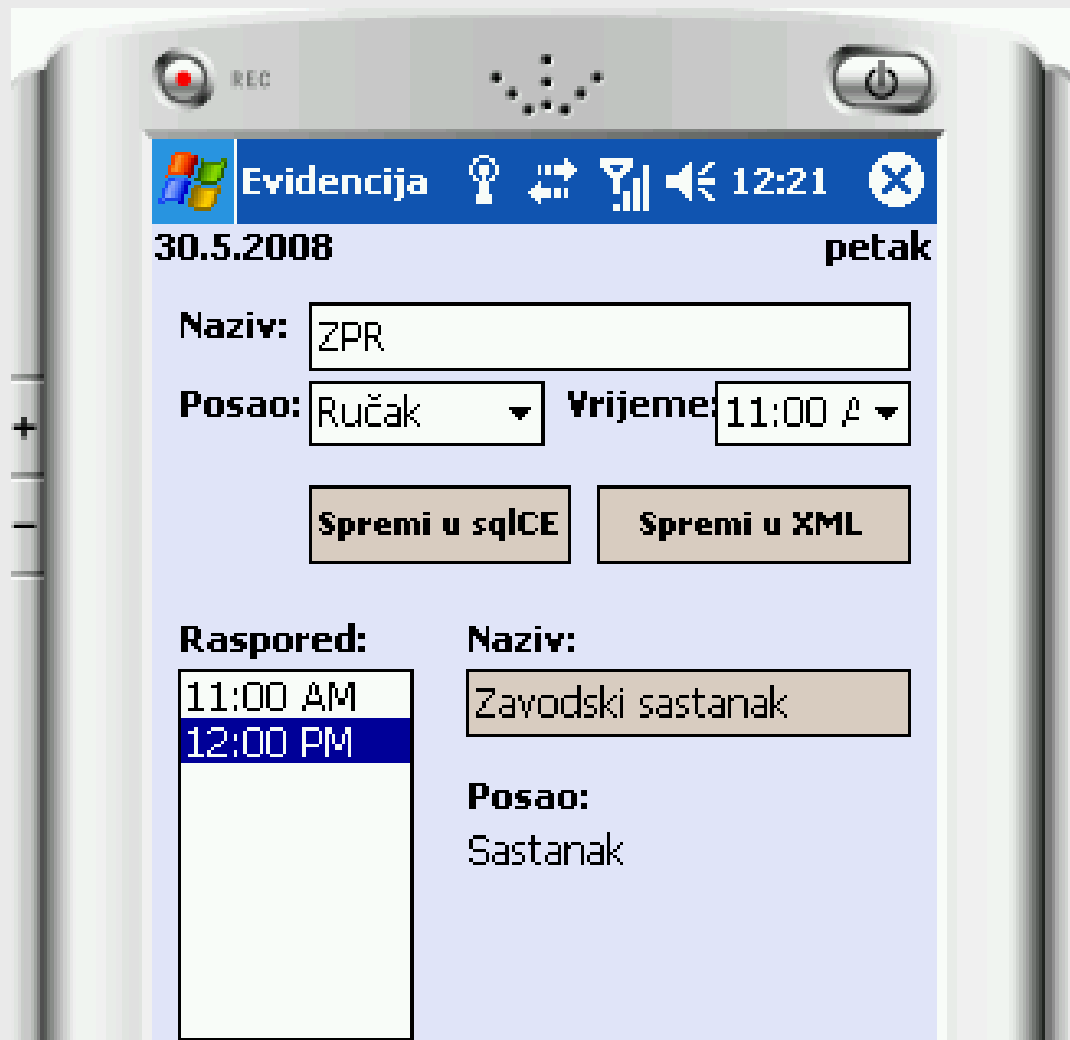


# Lokalna evidencija na mobilnom uređaju

❑ Primjer  Mobilne \ EvidencijaPoslova

❑ Aplikacija za vođenje poslova

- Mogućnost dodavanja i pregleda poslova za današnji dan
- Podaci se spremaju u XML datoteku ili u bazu podataka na mobilnom uređaju



REC

Evidencija 12:21

30.5.2008 petak

Naziv: ZPR

Posao: Ručak Vrijeme: 11:00 A

Spremi u sqlCE Spremi u XML

Raspored:

11:00 AM  
12:00 PM

Naziv: Zavodski sastanak

Posao: Sastanak

# Inicijalizacija zaslonske maske i povezivanje

## ❑ Inicijalizacija zaslonske maske (AppointmentList\_Load)

```
//Hrvatska verzija dana u tjednu
System.Globalization.CultureInfo hrvatski =
    System.Globalization.CultureInfo.GetCultureInfo("hr-HR");

int day = (int)DateTime.Now.DayOfWeek;
DanLabel.Text = hrvatski.DateTimeFormat.DayNames[day];
DatumLabel.Text = DateTime.Now.ToString("dd.MM.yyyy");
```

## ❑ Inicijalno čitanje podataka (AppointmentList\_Load)

## ❑ Postupak SetControls

### ■ Povezivanje kontrola s podacima

```
vrijemeComboBox.DataSource = vrijemeStringArray;
rasporedListBox.DataSource = zadatakDataTable;
rasporedListBox.DisplayMember = "vrijeme";
nazivRasporedTextBox.DataBindings.Add("Text", zadatakDataTable,
    "naziv");
...
```

# Lokalna baza podataka

## ❑ Prostor imena `System.Data.SqlServerCe`

- Razredi za rad s mobilnom bazom podataka
  - `SqlCeConnection`, `SqlCeCommand`, `SqlCeDataReader`, `SqlCeDataAdapter`, ...
- `SqlServerEngine` – razred za rad s objektima sustava za upravljanje mobilnom bazom podataka (kreiranje baze podataka)

## ❑ Baza podataka na mobilnom uređaju je datoteka s ekstenzijom `.sdf`

## ❑ Primjer Mobilne/EvidencijaPoslova

- Stvaranje baze podataka i tablice

```
private void CreateDatabase() {  
    SqlCeEngine newSqlDB = new SqlCeEngine(sqlCEConnectionString);  
    newSqlDB.CreateDatabase();  
    SqlCeCommand makeTableCommand = new SqlCeCommand("CREATE TABLE...",  
        new SqlCeConnection(sqlCEConnectionString)  
    makeTableCommand.Connection.Open();  
    makeTableCommand.ExecuteNonQuery();  
}
```

# Čitanje podataka iz baze podataka

## ❑ Čitanje zapisa iz mobilne baze podataka

```
private void GetDataFromSLQCE()  
{  
    if (zadatakDataTable == null)  
    {  
        zadatakDataTable = new DataTable();  
    }  
    SqlCeDataAdapter zadatakDataAdapter = new SqlCeDataAdapter  
        ("Select vrijeme, naziv, posao from raspored order by vrijeme",  
        new SqlCeConnection(sqlCEConnectionString));  
    zadatakDataTable.Clear();  
    zadatakDataAdapter.Fill(zadatakDataTable);  
}
```

# Spremanje podataka u bazu podataka

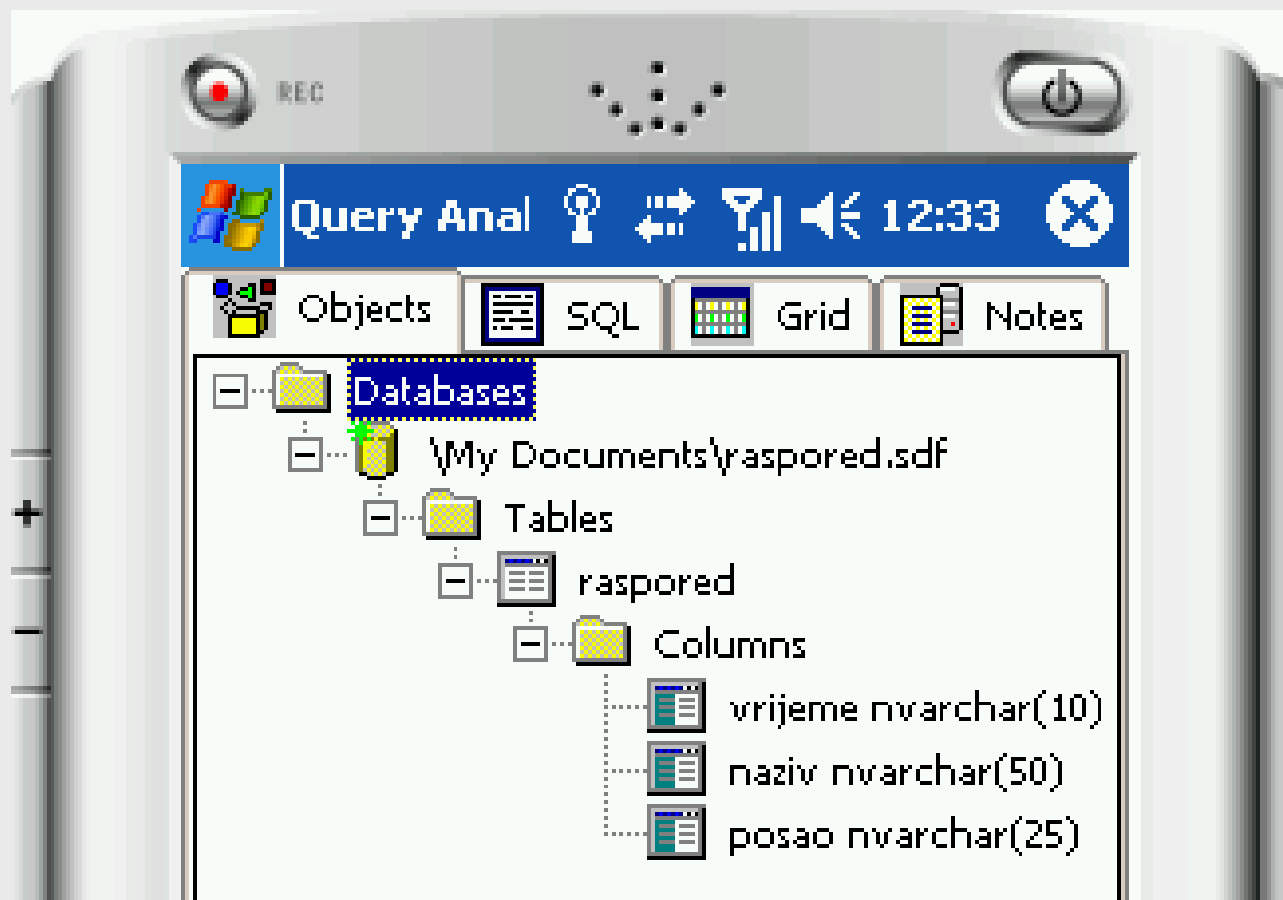
## ❑ SaveToSqlCEButton\_Click

```
DataRow newRow = zadatakDataTable.NewRow();
newRow["vrijeme"] = vrijemeComboBox.Text;
...
zadatakDataTable.Rows.Add(newRow);
SqlCeCommand newDataInsertCommand = new SqlCeCommand
    ("INSERT INTO raspored (vrijeme, naziv, posao) VALUES (?, ?, ?)",
    new SqlCeConnection(sqlCEConnectionString));
newDataInsertCommand.Connection.Open();
newDataInsertCommand.Parameters.Add("vrijeme", SqlDbType.NVarChar, 10);
...
foreach (DataRow dr in zadatakDataTable.Rows) {
    if (dr.RowState == DataRowState.Added) {
        newDataInsertCommand.Parameters["vrijeme"].Value = dr["vrijeme"];
        ...
        newDataInsertCommand.ExecuteNonQuery(); }
}
newDataInsertCommand.Connection.Close(); }
```

# Upravljanje mobilnom bazom podataka

## ❑ Pregled baze podataka na mobilnom uređaju

- *Programs->Query Analyzer*
- Datoteka @*"My Documents\raspored.sdf"*



# Stvaranje i čitanje XML datoteke

- ❑ **Primjer**  Mobilne \ EvidencijaPoslova - raspored.xml
- ❑ **BuildXMLDoc** - kreiranje datoteke (ako ne postoji na uređaju)

```
string xmlDataFile = @"Program Files\EvidencijaPoslova\raspored.xml";
XmlDocument xmlDoc = new XmlDocument();

...
if (!File.Exists(xmlDataFile)){
    XmlDeclaration dec =
        xmlDoc.CreateXmlDeclaration("1.0", "utf-8", null);
    xmlDoc.AppendChild(dec);
    XmlElement root = xmlDoc.CreateElement("raspored");
    xmlDoc.AppendChild(root);}
```

- ❑ **Postupak GetDataFromXML** - učitavanje iz datoteke u tablicu

```
DataSet ds = new DataSet();
ds.ReadXml(xmlDataFile);
DataTable tempDataTable = ds.Tables[0];
zadatakDataTable.DefaultView.Sort = "vrijeme";
```



# Zapisivanje u XML datoteku

## ❑ Vlastiti postupak BuildXMLDoc


- Dodavanje elementa “zadatak” s vrijednostima s korisničkog sučelja

```
XmlNode zadatakNode, vrijemeNode, ...;  
xmlDoc.Load(xmlDataFile);  
  
zadatakNode = xmlDoc.CreateElement("zadatak");  
xmlDoc.DocumentElement.AppendChild(zadatakNode);  
  
vrijemeNode = xmlDoc.CreateElement("vrijeme");  
vrijemeNode.InnerText = vrijemeComboBox.SelectedValue.ToString();  
zadatakNode.AppendChild(vrijemeNode);  
  
...  
FileStream strmSaveToFile = new FileStream(xmlDataFile,  
    FileMode.OpenOrCreate, FileAccess.Write);  
StreamWriter wrDataWriter = new StreamWriter(strmSaveToFile);  
wrDataWriter.Write(xmlDoc.InnerXml);  
wrDataWriter.Close();
```

# Primjer spajanja mobilnog uređaja na web servis

## ❑ Primjer Mobilne/Skladiste

- Skladištar unosi datume, te dohvaća popis artikala i prodane količine za odabrano razdoblje iz središnje baze podataka
- Skladištar pregledava popis te evidentira koliko pojedinih artikala treba naručiti
- Narudžba se zapisuje u središnju bazu podataka




The screenshot shows a handheld device screen with a blue header bar containing the Windows logo, the title "Skladište", and various status icons (signal, battery, time 4:32). Below the header, there are date selection fields: "Od: 1 / 1 / 07" and "Do: 2 / 1 / 07". A table with 5 columns is displayed: "SifArtikl", "NazArtikl", "KolArtikl", "Kolicina", and a small triangle icon. The table contains 8 rows of data. The row with "SifArtikl" 24857 and "NazArtikl" "Knjiga" is highlighted in blue. Below the table, the text "Knjiga 'CCNA' + CD (Knjiga 'CCNA' + CD)" is displayed. A label "Količina za naručiti:" is followed by a text input field containing the number "10". At the bottom, there is a button labeled "Prikaži Naruči" with a keyboard icon and a small triangle icon.

SifArtikl	NazArtikl	KolArtikl	Kolicina	
26035	DVD/DivX,	169.0000	0	
21824	FireWire	143.0000	0	
25018	DVD/DivX,	142.0000	0	
24857	Knjiga	125.0000	10	
30647	Notebook	124.0000	20	
24830	Knjiga	122.0000	0	
9399	GPS	121.0000	0	
5382	Knjiga	120.0000	0	

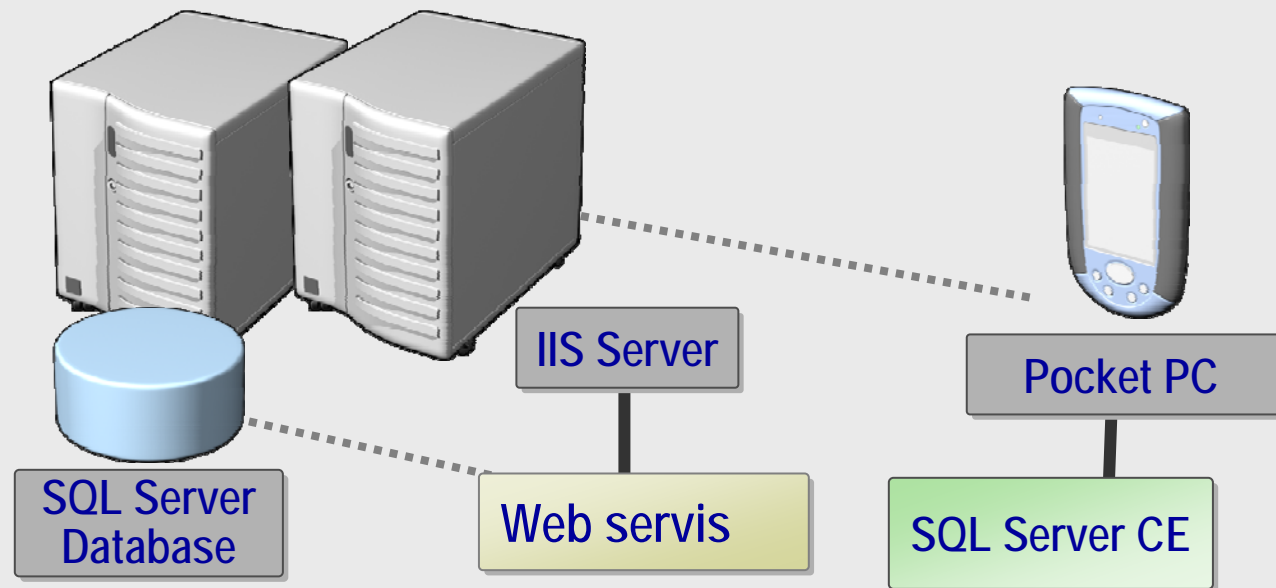
Knjiga "CCNA" + CD (Knjiga "CCNA" + CD)

Količina za naručiti:

Prikaži Naruči 

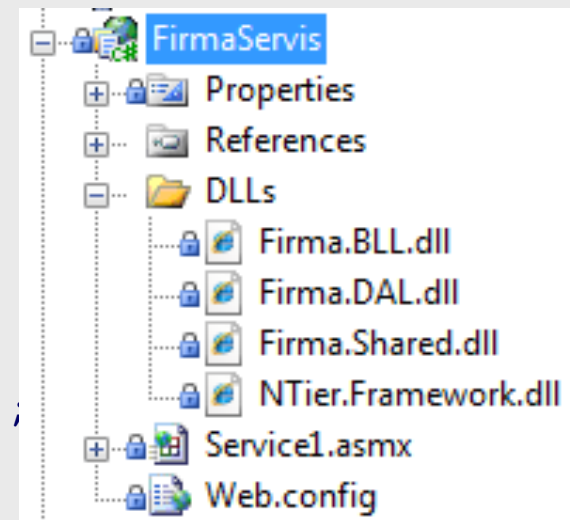
# Udaljena baza podataka

## ❑ Pristup podacima putem Web servisa



## ❑ Primjer Mobilne \ FirmaServis

- Dodavanje .dll datoteka
  - *Add->Existing Item* – odabrati datoteke
- Korištenje razreda definiranih u .dll datotekama
  - `using Firma;`
  - `StavkaList lista = new StavkaList();`



# Dohvat podataka putem web servisa

## ❑ Primjer Mobilne \ FirmaServis - DohvatiStavkeUkupno

- Dohvat liste ukupnog broja prodanih artikala u zadanom razdoblju
- Direktan upit na bazu, spremanje u `StavkaList` iz *Firma.Shared.dll*

```
[WebMethod]
public StavkaList DohvatiStavkeUkupno(
    DateTime DatumOd, DateTime DatumDo){
    upit = "...";
    SqlConnection conn = new
        SqlConnection(ConfigurationManager.ConnectionStrings
            ["ConnectionString"].ToString());
    ...
    StavkaList lista = new StavkaList();
    while (reader.Read()){
        Stavka s = new Stavka();
        ...
        lista.Add(s);
    }
    return lista;}
```

# Pohrana podataka putem web servisa

## ❑ Stvaranje narudžbe od dobivenih stavki – NapraviNarudzbu

- Stvaranje novog dokumenta tipa 'N' (Naružba)
- Dodavanje stavki predanih kao parametar postupka
- Poziv `SaveChanges` na BLL sloju

**[WebMethod]**

```
public void NapraviNarudzbu(Stavka[] listaStavki){
    DokumentList dokumenti = new DokumentList();
    Dokument dok = dokumenti.AddNew();
    dok.VrDokumenta = "N";
    dok.BrDokumenta = NaredniBrDokumenta();
    dok.IdPartnera = 0;
    ...
    foreach (Stavka s in listaStavki){
        Stavka nova = dok.Stavke.AddNew();
        nova.SifArtikla = s.SifArtikla;
        ...}
    DokumentBllProvider dokumentBll = new DokumentBllProvider();
    dokumentBll.SaveChanges(dokumenti.GetChanges());}
```

# Pohrana podataka putem web servisa

- ❑ **Pri stvaranju novog dokumenta ne znamo od koga će se nabaviti pa uvodimo "nepoznatog partnera"**
  - U tablicu Partner dodajemo (jednom) partnera s IdPartnera=0
  - IdPartnera je Identity! – SET IDENTITY\_INSERT Partner ON
- ❑ **BrDokumenta mora biti jedinstven**
  - Dohvaćamo (u web servisu) naredni broj dokumenta kao najveći broj postojećih dokumenata u tablici *Dokument* uvećan za 1

```
SELECT MAX (BrDokumenta)+1 from Dokument
```

- ❑ **Upid nad bazom podataka**

```
SET IDENTITY_INSERT Partner ON
INSERT INTO Partner
        (IdPartnera
        ,TipPartnera
        ,IdMjestaPartnera
        ,AdrPartnera
        ,IdMjestaIsporuke
        ,AdrIsporuke)
VALUES
        (0
        , 'T'
        , null
        , null
        , null
        , null)
SET IDENTITY_INSERT Partner OFF
```

# Spajanje na web servis

## ☐ Dodavanje reference na web servis

- Web servis treba biti objavljen (*publish*) na nekom poslužitelju, ne možemo ga pokrenuti iz *solutiona*, jer je mobilni uređaj drugo (udaljeno) računalo
- Potrebno je dodati referencu na web servis
- Mobilni uređaj mora biti spojen na Internet da bi mogao pokrenuti servis

## ☐ Kreiranje servisa - kopiranjem na poslužitelj

- kopiranje *.asmx* datoteke i *bin* foldera u *c:\inetpub\wwwroot\FirmaServis*
- stvaranje virtualnog kazala FirmaServis nad fizičkim folderom
- test - <http://localhost/firmaservis/skladiste.asmx>

## ☐ Na SQL Server potrebno je dodati korisnike baze podataka

- NT AUTHORITY\NETWORK SERVICE
- <POSLUZITELJ>ASPNET
- SQL Server Manager → Security → Logins
- SQL Server Manager → Databases → Firma → Security → Users

# Dohvat podataka na mobilni uređaj

## ❑ Prikaz dohvaćenih podataka u *DataGrid* kontroli

- Kontrola slična *DataGridView* kontroli windows aplikacija
- Referenca na `System.Windows.Forms.DataGrid`
- Elemente *DataGrid* za mobilni uređaj nije moguće uređivati!
- Nije moguće dodavati stupce koji nisu povezani na neki izvor podataka

## ❑ Uređivanje – mogući načini

- Korištenje nove forme za uređivanje podataka (npr. klik na grid -> dohvat elementa -> prikaz detalja elementa u kontrolama koje se mogu uređivati)
- Nevidljivi *TextBox* koji se prilikom promjene aktivne ćelije postavlja na njeno mjesto i omogućava unos – simulacija mreže koja se može uređivati

	SifArtikl.	NazArtik	KolArtikl	Kolicina	▲
	26035	DVD/DivX,	169.0000	0	≡
	21824	FireWire	143.0000	0	
	25018	DVD/DivX,	142.0000	0	
▶	24857	Knjiga	125.0000	10	
	30647	Notebook	124.0000	20	



# Dohvat podataka na mobilni uređaj

## ❑ Primjer Mobilne \ Skladiste – Form1 \ NapuniTablicu

- Spremanje podataka dobivenih od servisa u lokalnu tablicu
- proširenje tablice stupcem količine za narudžbu

```
private DataTable NapuniTablicu() {  
    ...  
    DataTable tablica = new DataTable();  
    tablica.Columns.Add("SifArtikla", typeof(int));  
    ...  
    tablica.Columns.Add("Kolicina", typeof(decimal));  
  
    Servis.Stavka[] lista = servis.DohvatiStavkeUkupno(datumOd,datumDo);  
    foreach (Servis.Stavka stavka in lista){  
        tablica.Rows.Add(stavka.SifArtikla, stavka.NazArtikla,  
            stavka.KolArtikla, 0);  
    }  
    return tablica;  
}
```

# Uređivanje podataka

## ❑ Prikaz detalja za uređivanje

- Klikom mišem na redak mreže, prikazuju se detalji odabranog elementa i *TextBox* kontrola s mogućnošću unosa nove vrijednosti

25018	DVD/DivX,	142.0000	0
24857	Knjiga	125.0000	10
30647	Notebook	124.0000	20
24830	Knjiga	122.0000	0
9399	GPS	121.0000	0
5382	Knjiga	120.0000	0

Knjiga "CCNA" + CD (Knjiga "CCNA" + CD)

Količina za naručiti:

```
private void dataGrid1_CurrentCellChanged(object sender, EventArgs e)
{
    indeksTrenutnogRetka = dataGrid1.CurrentRowIndex;

    labelNazivArtikla.Text = tablica.Rows[indeksTrenutnogRetka]
        [(int)Redak.NazArtikla].ToString()
        + " (" + tablica.Rows[indeksTrenutnogRetka]
        [(int)Redak.SifArtikla].ToString() + ")";
    textBoxKolicina.Text = tablica.Rows[indeksTrenutnogRetka]
        [(int)Redak.Kolicina].ToString();
}
```

# Slanje podataka na web servis

- ❑ **Slanje stavki (artikala i količine) servisu koji će napraviti narudžbu**
  - Slanje samo onih stavki čija je količina >0

```
private void menuItemNaruci_Click(object sender, EventArgs e){
    List<Servis.Stavka> lista = new List<Servis.Stavka>();
    for (int i=0; i < tablica.Rows.Count; i++){
        DataRow redak = tablica.Rows[i];
        if ((decimal) redak[(int)Redak.Kolicina] > 0){
            Servis.Stavka stavka = new Servis.Stavka();
            stavka.SifArtikla = (int)redak[(int)Redak.SifArtikla];
            ...
            lista.Add(stavka);
        }
    }
    Servis.Stavka[] listaStavki = new Servis.Stavka[lista.Count];
    ...
    servis.NapraviNarudzbu(listaStavki);
}
```

# Zadaci za vježbu

- ☐ Aplikaciju *EvidencijaPoslova* nadopuniti funkcionalnošću brisanja posla.
- ☐ Napisati aplikaciju za pregled, dodavanje i uređivanje podataka o poslovnim partnerima.
- ☐ Web servisu iz primjera dodati postupke za dohvat i spremanje podataka o artiklima (koristeći .dll datoteke iz WinFirma). Napraviti aplikaciju za pregled, izmjenu i pohranu promjena o artiklima.



## Domaća zadaća

- ☐ **Svaki član treba izraditi samostalnu aplikaciju za mobilni uređaj, koja podržava sljedeće funkcije**
  - Stvaranje lokalne baze podataka
  - Prikaz i uređivanje podataka iz baze podataka
  - Spremanje podataka u bazu podataka
- ☐ **Rješenje cijele grupe mora biti sadržano unutar jednog *Solutiona***
- ☐ **Nije potrebno koristiti web servis!**

# Reference

## ☐ Resursi

- Mobilne-Upute.PDF

## ☐ URL

- <http://msdn2.microsoft.com/en-us/windowsmobile/default.aspx>
- <http://www.microsoft.com/sql/editions/compact/default.mspx>
- <http://www.microsoft.com/windowsmobile/activesync/>
- <http://blogs.msdn.com/akhune/archive/2005/11/16/493329.aspx>
- <http://blogs.msdn.com/vsdteam/archive/2005/10/07/478295.aspx>
- <http://support.microsoft.com/kb/839013>
- <http://samples.gotdotnet.com/quickstart/CompactFramework/doc/sqlcemergereplication.aspx>
- [http://msdn.microsoft.com/en-us/library/ms229684\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms229684(VS.80).aspx)