

Pogled (View) MVC aplikacije:

a)	Sve od navedenog
b)	definira objekte modela
c)	upravlja podacima iz modela
d)	definira poslovna praavila
<input checked="" type="checkbox"/> e)	definira izgled korisničkog sučelja

Svaka serverska kontrola unutar web-aplikacije mora imati atribute

☐ a) id i runat="server"

☐ b) id i name

☐ c) id i runat="client"

☒ d) name i runat="server"

☐ e) text i runat="server"

„Centralizirana ili distribuirana obrada i pohrana? Kako? Tehnologije?“ Ova pitanja se postavljaju kod:

- | | |
|-------------------------------------|---|
| <input checked="" type="checkbox"/> | a) dizajna sučelja |
| <input type="checkbox"/> | b) analize i distribucije procesa |
| <input type="checkbox"/> | c) analize i distribucije podataka |
| <input type="checkbox"/> | d) odabira tehničke arhitekture sustava |
| <input type="checkbox"/> | e) izrade fizičkog modela podataka |

Svojstvo da promjena označenog elementa (primjerice na nekoj listi) uzrokuje trenutni postback zove se

<input checked="" type="radio"/>	DoPostBack
<input type="radio"/>	RunPostBack
<input type="radio"/>	NoPostBack
<input type="radio"/>	ChangePostBack
<input type="radio"/>	AutoPostBack

U kojoj je arhitekturi osnovni problem to što svi podaci sa poslužitelja moraju doći do klijenta za obradu?

a) Klijent-poslužitelj (client-server based)

b) Klijent-mrežni poslužitelj (client-network based)

c) Mrežni poslužitelj (network-server based)

☐ d) Klijentskoj (client-based)

☒ e) Poslužiteljskoj (server based)

S obzirom na opt-out i opt-in principe serijalizacije koji od navedenih odgovora je ekvivalentan sljedećem razredu

```
[DataContract]
public class Mjesto {
    [DataMember] public int Pbr { get; set; }
    public string NazMjesta { get; set; }
    [DataMember] public string NazDrzave { get; set; }
    [DataMember] public string PostNazMjesta { get; set; }
}
```

a) ☐

```
public class Mjesto {
    [DataMember] public int Pbr { get; set; }
    public string NazMjesta { get; set; }
    [DataMember] public string NazDrzave { get; set; }
    [DataMember] public string PostNazMjesta { get; set; }
}
```

b) ☐

```
[DataContract]
public class Mjesto {
    public int Pbr { get; set; }
    [XmlIgnore] public string NazMjesta { get; set; }
    public string NazDrzave { get; set; }
    public string PostNazMjesta { get; set; }
}
```

☒

```
[DataContract]
public class Mjesto {
    public int Pbr { get; set; }
    public string NazMjesta { get; set; }
    public string NazDrzave { get; set; }
    public string PostNazMjesta { get; set; }
}
```

d) ☐

```
public class Mjesto{
    public int Pbr { get; set; }
    [IgnoreDataMember] public string NazMjesta { get; set; }
    public string NazDrzave { get; set; }
    public string PostNazMjesta { get; set; }
}
```

☐

```
public class Mjesto{
    public int Pbr { get; set; }
    public string NazMjesta { get; set; }
}
```

Kod višeslojne aplikacije, interakcija prema dolje (Top-down interaction) odnosi se na:

☒ a) komunikaciju viših i nižih slojeva preko dijeljenog (shared) projekta

☐ b) komunikaciju nižih slojeva s višim slojevima, ali ne i obrnuto

☐ c) komunikaciju među slojevima općenito

☐ d) komunikaciju višeslojne aplikacije sa spremištem podataka

☐ e) komunikaciju viših slojeva s nižim slojevima, ali ne i obrnuto

Koje aplikacijske funkcije obavlja poslužitelj u dvoslojnoj arhitekturi klijent-poslužitelj?

- a) ☐ Pristup podacima (data access logic) i sučelje (presentation logic)
- ☒ b) Elementi obrade (application logic) i pohrana podataka (data storage)
- c) ☐ Elementi obrade (application logic) i sučelje (presentation logic)
- d) ☐ Pohrana podataka (data storage) i pristup podacima (data access logic)
- e) ☐ Elementi obrade (application logic) i pristup podacima (data access logic)

Čemu služe atributi KnownType i ServiceKnownType kod WCF servisa?

- ☒ a) Za najavu svih poznatih izvedenih razreda u postupku koji vraća općenitu vrijednost (sučelje, bazni ili apstraktni razred, tip Object)
- b) Za označavanje koja svojstva razreda će biti serijalizirana, a koja ne
- c) Za označavanje koji postupci su postupci WCF servisa
- d) Za definiranje načina povezivanja WCF servisa
- e) Za definiranje pristupne točke WCF servisa

Koja od navedenih tvrdnji vrijedi za izvješća:

- a) Izvješće je preslika windows formi u format prikladan za printanje.
- ☒ b) Ista struktura izvješća dinamički se može puniti različitim podacima
- c) Datoteka u kojoj je definirana struktura izvješća ima ekstenziju .xml
- d) Svako polje izvješća pridruženo je poljima iz baze podataka.
- e) Izvješće je interaktivna komponenta windows formi.

Ako je unutar web servisa definiran sljedeći razred

```
[XmlType("D")]  
public class Drzava{  
    [XmlIgnore]  
    public string Sifra { get; set; }  
    [XmlElement("Ozn")]  
    public string Oznaka { get; set; }  
    public string Naziv { get; set; }  
    [XmlAttribute("ISO3")]  
    public string Iso3Oznaka { get; set; }  
}
```

koji od odgovora predstavlja isječak xml-a koji se vraća pozivatelju servisa

a)

```
<Drzava ISO3="HRV">  
    <Oznaka>HR</Ozn>  
    <Naziv>Croatia</Naziv>  
</Drzava>
```

b)

```
<D ISO3="HRV">  
    <Ozn>HR</Ozn>  
    <Naziv>Croatia</Naziv>  
</D>
```

c)

```
<Drzava>  
    <Oznaka>HR</Oznaka>  
    <Naziv>Croatia</Naziv>  
    <ISO3>HRV</ISO3>  
</Drzava>
```

d)

```
<D ISO3="HRV" Ozn="HR" Naziv="Hrvatska" />
```

e)

```
<Drzava>  
    <Sifra>13</Sifra>  
    <Oznaka>HR</Oznaka>  
    <Naziv>Croatia</Naziv>  
    <ISO3>HRV</ISO3>  
</Drzava>
```

Koji razred je osnova za refleksiju, za pristup metapodacima?

a) System.Attribute

☒ b) System.Type

c) System.Typical

☐ d) System.Data

e) System.Windows.Metadata

Gdje se, u višeslojnoj aplikaciji, savjetuje stvarati poslovni objekt?

- ☒ a) u poslovnom sloju
- ☐ b) u podatkovnom sloju
- ☐ c) u prezentacijskom sloju
- ☐ d) ne treba stvarati poslovne objekte uopće
- ☐ e) u bazi podataka

U kojoj je arhitekturi osnovni problem to što poslužitelj mora obraditi sve poruke?

- a) Mrežni poslužitelj (network-server based)
- ☒ b) Poslužiteljskoj (server based)
- c) Klijent-mrežni poslužitelj (client-network based)
- d) Klijent-poslužitelj (client-server based)
- e) Klijentskoj (client-based)