

Zajednički sustav tipova podataka .NET Frameworka naziva se:

a

Common Language Runtime

b

Intermediate Language

c

Assembly

d

Base Class Library

e

Common Type System

Zahtjevi koji opisuju zadatke koje korisnik mora moći obaviti služeći se aplikacijama su:

a nefunkcionalni

b poslovni

c funkcionalni

d obvezni

e korisnički

Koja od navedenih tvrdnji **NIJE** istinita?

a

Isti prostor imena može se protezati u više datoteka.

b

Prostor imena sadrži definicije razreda, struktura, sučelja, pobrojanih tipova, delegata i deklaraciju drugih prostora imena.

c

Svaka datoteka sadrži jedan ili više prostora imena.

d

Isti naziv tipa **ne smije se** upotrijebiti u različitim prostorima imena.

e

C# program se može sastojati od više datoteka.

Izbacite uljeza

a

debeli klijent

b

tanki klijent

c

hibridna aplikacija

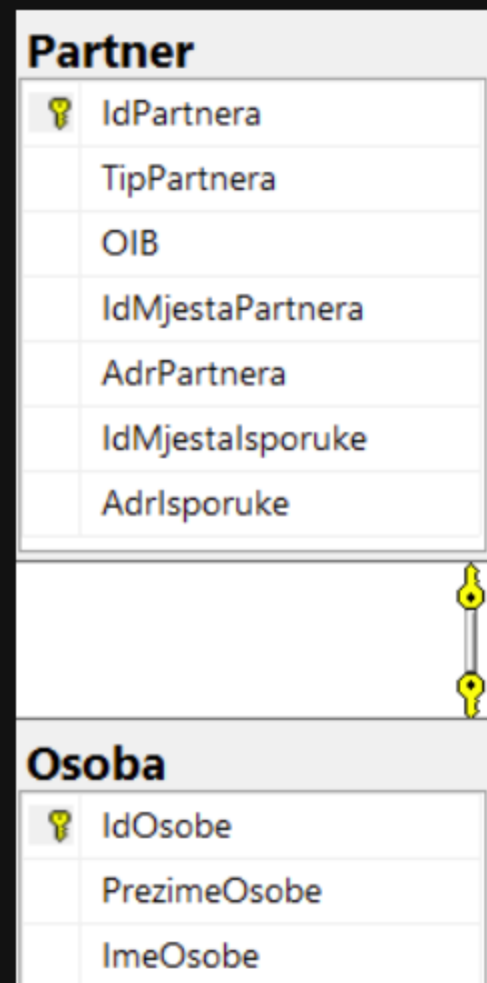
d

višeslojna aplikacija

e

mobilna aplikacija

Ako je u bazi podataka odnos među tablicama *Partner* i *Osoba* kao na sljedećoj slici tada razred *Osoba* u Entity Framework modelu generiranom prema navednoj BP izgleda nalik



a

```
public partial class Osoba {  
    ...  
    public ICollection<Partner> Partner { get; set; }  
}
```

b

```
public partial class Osoba {  
    ...  
    public int IdOsobe {get; set; }  
    public int IdPartnera { get; set; }  
    ...  
    public Partner IdOsobeNavigation { get; set; }  
}
```

c

```
public partial class Osoba {  
    ...  
    public int IdOsobe { get; set; }  
    ...  
    public Osoba IdOsobe { get; set; }  
}
```

d

```
public partial class Osoba {  
    ...  
    public Partner IdOsobeNavigation { get; set; }  
}
```

e

```
public partial class Osoba {  
    ...  
    public int IdPartnera { get; set; }  
    ...  
    public ICollection<Partner> IdOsobeNavigation { get; set; }  
}
```

U kojoj fazi RUP-a se prikupljaju zahtjevi

a

u fazi počinjanja i elaboracije

b

u fazi prijelaza

c

u fazi počinjanja

d

u svim osim u fazu prijelaza

e

u fazi konstrukcije

Hijerarhijska lista faza, zadataka i prekretnica se skraćenom naziva

a

Top-down

b

PTM

c

WBS

d

PhaTaMi

e

ABS


```
SELECT OznDrzave, NazDrzave FROM Drzava;
```

i odgovarajuće instanciranje objekata tipa DbCommand i IDataReader za gore navedenu naredbu, što od navedenog **NIJE** ispravan način dohvata vrijednosti nekog retka u tablici Drzava?

a

```
string oznaka = reader[0].ToString();  
string naziv = reader[1].ToString();
```

b

```
string oznaka = reader.GetString(0);  
string naziv = reader.GetString(1);
```

c

```
string oznaka = (string) reader.GetName("OznDrzave");  
string naziv = (string) reader.GetName("NazDrzave");
```

d

```
string oznaka = reader["OznDrzave"].ToString();  
string naziv = reader["NazDrzave"].ToString();
```

e

```
string oznaka = (string) reader["OznDrzave"];  
string naziv = (string) reader["NazDrzave"];
```

Što od navedenog nije vrijednosni tip

a

int?

b

Nullable

c

string

d

System.Int32

e

float

```
Razred r1 = new Razred("A1");  
using (Razred r2 = new Razred("B2")) {  
    Razred r3 = new Razred("C3");  
    r3 = null;  
    throw new Exception("Poruka");  
}  
r1.Dispose();
```

Ako je razred *Razred* implementiran tako da u postupku *Dispose* ispisuje parametar primljen u konstruktoru, gornji programski odsječak će ispisati

a

C3

b

B2
C3

c

B2

d

B2
C3
A1

e

B2
A1

Parcijalni razred je razred

a

čija definicija može biti napisana u više datoteka.

b

koji nema implementirane sve postupke.

c

koji je napisan u nekoliko prostora imena

d

koji se ne može instancirati

e

koji sadrži samo popis postupaka.

Ako *Automobil* i *Motocikl* nasljeđuju *MotornoVozilo* onda je u postupak *IspisiVozilo* moguće poslati i `List< Automobil>`, jer je sučelje `IEnumerable< T>`

```
void IspisiVozila(IEnumerable<MotornoVozilo> vozila) {  
    for(var vozilo in vozila)  
        Console.WriteLine(vozilo.Model);  
}
```

a

kontravarijantno

b

irelevantno

c

kovarijantno

d

varijantno

e

invarijantno

"Parser će brzo generirati izvješće o pogreškama HTML oznaka, koje omogućava brzi ispravak pogrešaka kada program koriste početnici u HTML-u." Ovo je primjer:

a zahtjeva krajnjih korisnika

b nepotpunog zahtjeva

c neodređenog zahtjeva

d neostvarivog zahtjeva

e zahtjeva vlasnika sustava

```
try {  
    ...  
    throw new NullReferenceException();  
}  
catch (Exception exc){  
    Console.WriteLine("EXC");  
}  
catch (NullReferenceException nre) {  
    Console.WriteLine("NRE");  
}  
finally {  
    Console.WriteLine("F");  
}
```

Izvršavanjem gornjeg odsječka na ekranu se ispisuje

a

EXC

F

b

EXC

NRE

F

c

NRE

F

d

NRE

e

F

Koja tvrdnja ne vrijedi za klasični vodopadni model?

- a** Iterativan postupak razvoja
- b** Nisu dozvoljene naknadne promjene rezultata prethodnih faza
- c** Razvoj započinje tek kad su svi zahtjevi dobro dokumentirani
- d** Slijedno napredovanje iz faze u fazu
- e** Problemi u slučaju pogrešaka ili novih/promijenjenih zahtjeva


```
class Primjer
{
    private int index;

    public int Index
    {
        get { return index; }
        set {
            if (value <= 0) { this.index = 0; }
            else if (value > 100) { this.index = 100; }
            else { this.index = value; }
        }
    }
}
```

Programski odsječak je primjer

a

iznimke

b

ofenzivnog programiranja

c

tvrdnje

d

indeksera

e

barikade