Id: 5367

Koja tvrdnja ne vrijedi za klasični vodopadni model?

**a**

Iterativan postupak razvoja

**b**

Razvoj započinje tek kad su svi zahtjevi dobro dokumentirani

**c**

Problemi u slučaju pogrešaka ili novih/promijenjenih zahtjeva

**d**

Slijedno napredovanje iz faze u fazu

**e**

Nisu dozvoljene naknadne promjene rezultata prethodnih faza

Id: 5371

Koja od navedenih tvrdnji **NIJE** istinita?

**a**

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

**b**

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

**c**

C# program se može sastojati od više datoteka.

**d**

Svaka datoteka sadrži jedan ili više prostora imena.

**e**

Isti prostor imena može se protezati u više datoteka.

Id: 5373

Zajednički sustav tipova podataka .NET Frameworka naziva se:

**a**

Intermediate Language

**b**

Common Language Runtime

**c**

Common Type System

**d**

Base Class Library

**e**

Assembly

Id: 5385

"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**

nepotpunog zahtjeva

**b**

zahtjeva vlasnika sustava

**c**

neodređenog zahtjeva

**d**

zahtjeva krajnjih korisnika

**e**

neostvarivog zahtjeva

Id: 5386

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

**a**

obvezni

**b**

poslovni

**c**

funkcionalni

**d**

korisnički

**e**

nefunkcionalni

Id: 20106

Parcijalni razred je razred

**a**

koji sadrži samo popis postupaka.

**b**

koji je napisan u nekoliko prostora imena

**c**

koji nema implementirane sve postupke.

**d**

koji se ne može instancirati

**e**

čija definicija može biti napisana u više datoteka.

Id: 20107

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**

barikade

**b**

ofenzivnog programiranja

**c**

tvrdnje

**d**

iznimke

**e**

indeksera

Id: 20108

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**

F

**d**

NRE

F

**e**

NRE

Id: 20109

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**

invarijantno

**d**

kovarijantno

**e**

varijantno

Id: 20110

Izbacite uljeza

**a**

tanki klijent

**b**

hibridna aplikacija

**c**

debeli klijent

**d**

višeslojna aplikacija

**e**

mobilna aplikacija

Id: 20111

U kojoj fazi RUP-a se prikupljaju zahtjevi

**a**

u fazi počinjanja

**b**

u fazi počinjanja i elaboracije

**c**

u fazi konstrukcije

**d**

u fazi prijelaza

**e**

u svim osim u fazu prijelaza

Id: 20112

Hijerarhijska lista faza, zadataka i prekretnica se skraćenom naziva

**a**

ABS

**b**

WBS

**c**

Top-down

**d**

PhaTaMi

**e**

PTM

Id: 20113

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  
A1

**d**

B2  
C3  
A1

**e**

B2

Id: 20114

Što od navedenog nije vrijednosni tip

**a**

System.Int32

**b**

int?

**c**

float

**d**

string

**e**

Nullable

Id: 38263

Za upit definaran s

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 = reader["OznDrzave"].ToString();

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

**d**

string oznaka = (string) reader["OznDrzave"];

string naziv = (string) reader["NazDrzave"];

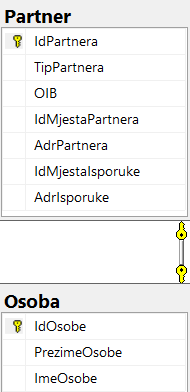
**e**

string oznaka = (string) reader.GetName("OznDrzave");

string naziv = (string) reader.GetName("NazDrzave");

Id: 38265

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 int IdPartnera { get; set; }

...

public ICollection<Partner> IdOsobeNavigation { get; set; }

}

**b**

public partial class Osoba {

...

public int IdOsobe { get; set; }

...

public Osoba IdOsobe { get; set; }

}

**c**

public partial class Osoba {

...

public Partner IdOsobeNavigation { get; set; }

}

**d**

public partial class Osoba {

...

public ICollection<Partner> Partner { get; set; }

}

**e**

public partial class Osoba {

...

public int IdOsobe {get; set; }

public int IdPartnera { get; set; }

...

public Partner IdOsobeNavigation { get; set; }

}