

Id: 5579

Nadjačanom članu `Postupak()` razreda `roditelja` pristupa se naredbom:

**a** `base.Postupak();`

**b** `new.Postupak();`

**c** Nadjačanom članu ne može se pristupiti.

**d** `Postupak();`

**e** `this.Postupak();`

Id: 5410

U kojoj fazi životnog ciklusa programske potpore se razvija specifikacija zahtjeva?

**a** oblikovanje

**b** primjena

**c** planiranje

**d** implementacija

**e** analiza

Id: 43058

Označite kriterije kvalitete zahtjeva.

**Napomena: Pitanje ima više točnih odgovora**

**a** singularnost

**b** modularnost

**c** izvedivost

**d** sljedivost

**e** konzistentnost

Id: 5604

Tehnika programiranja čija je ideja da će potprogram s neispravnim podacima "opstati" i onda kada su pogreškom pozivajuće procedure predani neispravni argumenti, naziva se:

**a** Ofenzivno programiranje

**b** Ekstremno programiranje

**c** Defenzivno programiranje

**d** Korektivno programiranje

**e** Generičko programiranje

Id: 6185

Što od navedenog **NIJE** istina za razrede izvedene iz sučelja `IDataReader`

- a** Moguće je vratiti se na prethodni redak
- b** Podatku u pojedinom retku može se pristupiti korištenjem naziva stupca
- c** Rezultat se može sastojati od više skupova rezultata
- d** Podatku u pojedinom retku može se pristupiti korištenjem rednog broja stupca
- e** Podaci dohvaćeni kroz ovo sučelje su samo za čitanje

Funkcionalni zahtjevi odgovaraju na pitanje:

**a**

kako ili kako dobro

**b**

što

**c**

zašto

**d**

kada

**e**

kojim redom

Koje su karakteristične aktivnosti za fazu (životnog ciklusa) analize?

- a** Prikupljanje informacija, modeliranje procesa i podataka
- b** Dizajn arhitekture, sučelja i programa
- c** Konstrukcija, testiranje i implementacija
- d** Dokumentiranje, održavanje
- e** Utvrđivanje ciljeva, izrada plana projekta, analiza izvedivosti

Id: 9759

Ako se unutar pogleda u MVC aplikaciji kao model koristi razred Mjesto koji ima svojstvo Naziv čija se vrijednost ispisuje unutar pogleda, kako izgleda odsječak takvog pogleda?

**a** @Model.Mjesto ...  
@Naziv

**b** @model.Mjesto ...  
@Model.Naziv

**c** @model.Mjesto ...  
@Naziv

**d** @model.Mjesto ...  
@Mjesto.Naziv

**e** @Model.Mjesto ...  
@model.Naziv



U planu projekta vrijedi sljedeća formula

**a**  $\text{Jedinice} = \text{Posao} * \text{Trajanje}$  ( $\text{Units} = \text{Work} * \text{Duration}$ )

**b**  $\text{Jedinice} = \text{Trajanje} + \text{Posao}$  ( $\text{Units} = \text{Duration} + \text{Work}$ )

**c**  $\text{Trajanje} = \text{Posao} * \text{Jedinice}$  ( $\text{Duration} = \text{Work} * \text{Units}$ )

**d**  $\text{Posao} = \text{Trajanje} * \text{Jedinice}$  ( $\text{Work} = \text{Duration} * \text{Units}$ )

**e**  $\text{Posao} = \text{Trajanje}$  ( $\text{Work} = \text{Duration}$ )

Koji od navedenih pojmova **nije** neka vrsta aplikacije?

**a**

umetnuta aplikacija

**b**

poslužiteljska aplikacija

**c**

mobilna aplikacija

**d**

samostalna aplikacija

**e**

višeslojna aplikacija

Id: 43076

Ako je u datoteci Startup.cs ASP.NET Core aplikacije u postupku Configure definirana sljedeća ruta

```
app.UseEndpoints(endpoints => {
    endpoints.MapControllerRoute("Mjesta i artikli",
        "{action}/{controller:regex:^(Mjesto|Artikl)$}"/Page{page}/Sort{sort:int}/ASC-{ascending:bool}
        new { action = "Index" }
    );
    endpoints.MapDefaultControllerRoute();
});
```

kako izgleda zadnji odsječak url adrese koja bi odgovarala sljedećem postupku iz datoteke MjestoContro

```
public IActionResult Edit(int id, int page, int sort, bool ascending){...}
```

- a** .../Edit/Mjesto/Page2/Sort3/ASC-True/1234
- b** .../Edit/Mjesto/Page2/Sort3/ASC-True?id=1234
- c** .../Edit/Mjesto/1234?page=2&sort=3&asc=True
- d** .../Mjesto/Edit/Page2/Sort3/ASC-True/1234
- e** .../Index/Mjesto|Artikl/Page2/Sort3/ASC-True?id=1234

Id: 20095

Planiranje (u životnom ciklusu programske potpore) daje odgovor na pitanje:

- a** Za koji operacijski sustav će se sustav implementirati?
- b** Što sustav mora raditi?
- c** Kako projektirati sustav?
- d** Zašto graditi sustav?
- e** Tko će koristiti sustav?

11100200

Skup poslanih promjena s jedinstvenim identifikatorom na centraliziranim sustavima za upravljanje izvornim kodom naziva se

**a** check out

**b** main trunk

**c** check in

**d** changeset

**e** workspace

Što od navedenog **nije ispravan** pristup za izradu dobrog programa

**a** smeće unutra, ništa van

**b** smeću zabranjen ulaz

**c** smeće unutra, poruka o pogrešci van

**d** smeće unutra, smeće van

**e** počisti smeće

Id: 38272

Neka su definirane sljedeće hijerarhije:

- Vozilo <- Automobil <- ElektricniAutomobil
- Vozilo <- Motocikl
- Osoba <- Djelatnik <- Direktor

i postupak `Djelatnik ZaduzenZa(Automobil a)`

Što je od navedenog ispravno

**a**

```
Func<ElektricniAutomobil, Osoba> func = ZaduzenZa;
```

**b**

```
Func<Vozilo, Direktor> func = ZaduzenZa;
```

**c**

```
Func<Vozilo, Osoba> func = ZaduzenZa;
```

**d**

```
Func<ElektricniAutomobil, Direktor> func = ZaduzenZa;
```

**e**

```
Func<Osoba, ElektricniAutomobil> func = ZaduzenZa;
```

Score: 14.500 (=72.5%)

27 😊

28 😐

29 😊

30 😊

31 😊

32 😊

33 😊

34 😊

40 😞

Id: 5419

Kod iterativnog postupka razvoja svaka se iteracija obavlja:

**a**

kroz faze ekstremnog programiranja

**b**

standardnim životnim ciklusom koji uključuje analizu, oblikovanje, ugradnju i provjeru

**c**

posebnim slučajem životnog ciklusa koji uključuje analizu i provjeru

**d**

kroz faze karakteristične za pojedinu iteraciju

**e**

kroz faze koje se definiraju ovisno o primjeni



Id: 20160

Način kreiranja Entity Framework modela u kojem baza podataka već postoji i model nastaje reverznim inženjerstvom BP naziva s

**a** Model First

**b** Reverse Model

**c** Database First

**d** Forward Model

**e** Migration Model

Što će se ispisati izvođenjem sljedećeg programskog odsječka

```
int a = 0, b = 0, c;  
try {  
    try {  
        Console.Write("T1 ");  
        c = b / a;  
    }  
    catch (Exception exc) {  
        Console.Write("E1 ");  
        c = a / b;  
    }  
    finally {  
        Console.Write("F1 ");  
    }  
}  
catch (Exception exc2) {  
    Console.Write("E2 ");  
}  
finally {  
    Console.Write("F2 ");  
}
```

**a**

T1 E2 F2

**b**

T1 E1 E2

**c**

T1 E1 F1 E2 F2

Id: 43065

Proces u kojem nakon svake promjene u repozitoriju dolazi do izgradnje programa i pokretanja testova, a zatim i do distribucije programa korisnicima (npr. u smislu objave nove verzije web-aplikacije ili dostupnosti ažuriranja) naziva se

**a** Push to publish (PTP)

**b** Continuous integration

**c** Commit to publish (CTP)

**d** Continuous publishing

**e** Continuous delivery

Id: 43078

`return RedirectToAction(...` unutar upravljača klijentu šalje statusnu poruku broj

**a** 500

**b** 200

**c** 302

**d** 404

**e** 204

Id: 5469

Što je kritični put?

- a** Skup zadataka s najduljim trajanjem
- b** Niz primitivnih zadataka koji se ne mogu podijeliti na jednostavnije zadatke
- c** Put koji treba zaobići
- d** Niz zadataka koji moraju završiti na vrijeme da bi projekt završio na vrijeme
- e** Osnova po kojoj se resursu dodjeljuje obavljanje posla

Id: 9587

Što će se ispisati izvođenjem sljedećeg programskog odsječka

```
class B {  
    public virtual void Test() {  
        Console.WriteLine("Pozdrav iz B");  
    }  
}  
  
class I : B {  
    public override void Test() {  
        Console.WriteLine("Pozdrav iz I");  
    }  
}  
  
class Program {  
    static void Main(string[] args)  
    {  
        I i = new I();  
        i.Test();  
        ((B)i).Test();  
        B b = i;  
        b.Test();  
    }  
}
```

**a**

Pozdrav iz B  
Pozdrav iz B  
Pozdrav iz B

**a**

Pozdrav iz B  
Pozdrav iz B  
Pozdrav iz B

**b**

Pozdrav iz B  
Pozdrav iz I  
Pozdrav iz B  
Pozdrav iz I  
Pozdrav iz B

**c**

Pozdrav iz I  
Pozdrav iz B  
Pozdrav iz B

**d**

Pozdrav iz I  
Pozdrav iz I  
Pozdrav iz B

**e**

Pozdrav iz I  
Pozdrav iz I  
Pozdrav iz I

Id: 43087

Što se sve može definirati i pretvoriti u Intermediate Language (razredi, strukture, sučelja, delegati...) definirano je s

**a** Common Language Runtime

**b** NuGetom

**c** Common Type System

**d** .NET Standardom

**e** Base Class Library



Id: 20173

Parovi oblika naziv=vrijednost navedeni iza upitnika u adresi neke stranice

- npr. **p=1&q=7&user=student** u adresi <http://apps.zpr.fer.hr/myapp/run?p=1&q=7&user=student>

tvore pojam koji se naziva

**a**

POST request

**b**

GET request

**c**

Query string

**d**

CGI parameter

**e**

URI string

Id: 5612

Što od navedenog nije preporuka za korištenje tvrdnji?

- a** Koristiti tvrdnje tamo gdje očekujemo pogreške
- b** Izbjegavati poziv metoda u izrazima tvrdnji
- c** Koristiti tvrdnje za dokumentiranje i verificiranje uvjeta koji moraju vrijediti poslije djelovanja metode ili rada s razredom ("pc
- d** Koristiti tvrdnje za dokumentiranje i verificiranje uvjeta koji moraju vrijediti prije pozivanja metode ili instanciranja razreda ("
- e** Za jako robustan programski kôd pored tvrdnji koristiti i kod za obradu pogreške

Id: 9851

Informacijski sustav je

- a** aplikacija koja pruža neku vrstu informacije
- b** skup elemenata ili objekata u jedinstvenoj konfiguraciji koju čine računalni programi, podaci i dokumentacija
- c** sustav aplikacija za upravljanje ljudskim aktivnostima
- d** računalom podržanom rješenje jednog ili više poslovnih problema ili potreba
- e** računalni program + baza podataka

Id: 5421

Za koji je razvoj karakteristično da se softver razvija i objavljuje po dijelovima, a glavne faze se obavljaju kroz niz iteracija?

**a** Spori i sistematični

**b** Razvoj po vodopadnom modelu

**c** Iterativni i sistematični

**d** Brzi i inkrementalni

**e** Iterativni i inkrementalni

Id: 20147

Što od navedenog **nije** moguća međuzavisnost zadatka

- a** količina posla prethodnika je jednaka količini posla sljedbenika
- b** završni datum prethodnika utvrđuje početni datum sljedbenika
- c** početni datum prethodnika utvrđuje početni datum sljedbenika
- d** završni datum prethodnika jest početni datum sljedbenika
- e** početni datum prethodnika utvrđuje završni datum sljedbenika

Id: 38276

Što ispisuje sljedeći programski odsječak?

```
static void Main(string[] args)
{
    Action<Func<int, int, int>> action = (f1) => Console.WriteLine(f1(5, 20));
    action += (f2) => Console.WriteLine(f2(3, 6));
    action((f1, f2) => f1 + f2);
}
```

**a**

5, 20

**b**

25  
9

**c**

9

**d**

5, 20  
3, 6

**e**

8  
26

Id: 20174

`asp-action` u sljedećem odsječku

```
<a asp-action="Create">Unos nove države</a>
```

je primjer korištenja

- a** util-atributa
- b** css stila
- c** vlastite komponente
- d** tag-helpera
- e** parcijalnog pogleda

```
}
```

**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 IdPartnera { get; set; }  
    ...  
    public ICollection<Partner> IdOsobeNavigation { get; set; }  
}
```

**d**

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

**e**

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



Id: 9752

Što od navedenog nije mogući rezultat neke akcije u nekom upravljaču (MVC)

**a** PartialViewResult

**b** ViewResult

**c** ControllerResult

**d** RedirectToRouteResult

**e** FileResult

Id: 5465

Planiranje s vrha prema dolje (Top-down) je jedan od pristupa razvoju zadataka i faza u kojem:

- a** je osnova pregledni raspored prekretnica
- b** su svi zadaci međusobno ovisni
- c** pristupa od specifičnom prema općem
- d** se identificiraju glavne faze i rezultati projekta prije dodavanja zadataka potrebnih za završetak tih faza
- e** se identificiraju zadaci najnižeg sloja prije grupiranja u faze

Id: 20118

Generičko sučelje definirano modifikatorom *in*, npr.

```
public interface IComparer<in T>
```

je

- a** invarijantno
- b** varijantno
- c** kontravarijantno
- d** kovarijantno
- e** izlazno-varijantno

Id: 20156

Koja razina važnosti zapisa se koristi za evidentiranje pogrešaka koje ne utječu na daljnji rad aplikacije, ali predstavljaju potencijalno opasne situacije te zahtijevaju naknadnu pažnju (npr. konfiguracijska datoteka sadrži traženu vrijednost, pa se koristi predodređena)?

**a** Trace

**b** Information

**c** Debug

**d** Warning

**e** Error

Id: 38260

Želimo li u nekom upitu na bazu podataka korištenjem ADO.NET-a izvršiti UPDATE, INSERT ili DELETE naredbu na nekom objektu tipa DbCommand pozvat ćemo postupak

**a** ExecuteReader

**b** ExecuteNonQuery

**c** ExecuteScalar

**d** SetData

**e** PostQuery

Id: 38266

*Semantic versioning* kojeg, između ostalih koristi i .NET Core za označavanje verzije objektna datoteke ima oblik

**a**

major.minor.patch-suffix

**b**

major.varijant.suffix

**c**

version.revision.variant

**d**

major.minor.baseline-suffix

**e**

baseline.release.suffix

Što ispisuje sljedeći programski odsječak?

```
static void Main(string[] args)
{
    Action<int, string, Func<char, string, int>> action = (i, s, f) => Console.WriteLine(f(s[i], s));
    action(6, "ispit iz RPPP-a", (x, s) =>
    {
        int sum = 0;
        foreach (var c in s)
        {
            if (x == c)
                ++sum;
        }
        return sum;
    });
}
```

**a** ništa

**b** 6

**c** 3

**d** f(s[6], "ispit iz RPPP-a")

**e** 0

```
using (var context = new FirmaContext()){  
    for(int i=1; i<=10 ; i++)  
    {  
        Artikl artikl = new Artikl  
        {  
            SifArtikla = i/2, CijArtikla = 10m,  
            JedMjere = "kom", NazArtikla = i.ToString()  
        }  
        context.Artikl.Add(artikl);  
    };  
    context.SaveChanges();  
}
```

broj zapisa u tablici Artikl povećat će se za

**a** 5

**b** 2

**c** 1

**d** 6

**e** 0