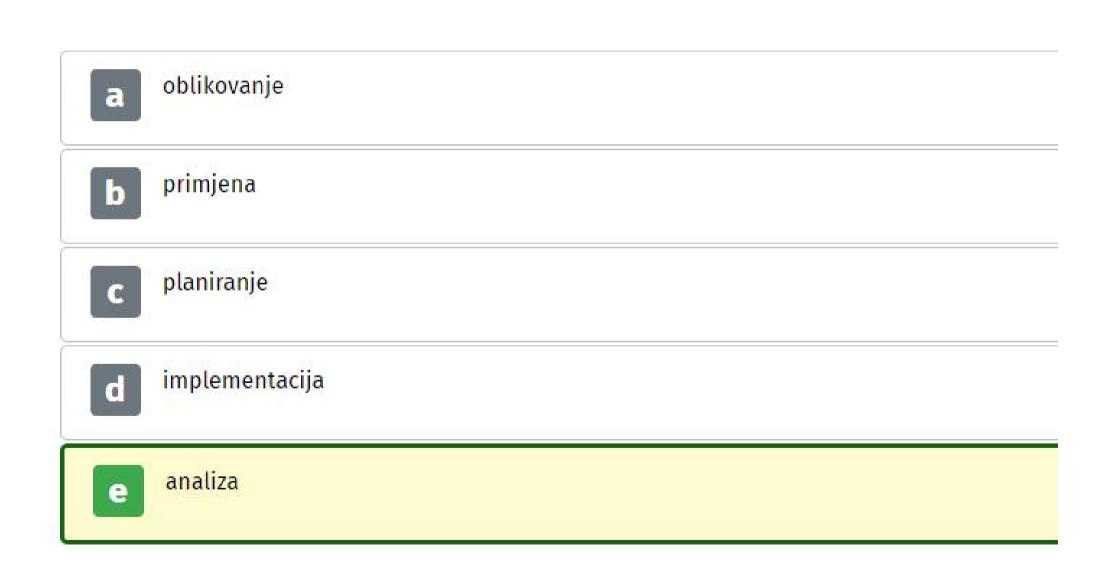


U kojoj fazi životnog ciklusa programske potpore se razvija specifikacija zahtjeva?



ld: 43058

Označite kriterije kvalitete zahtjeva.

### Napomena: Pitanje ima više točnih odgovora

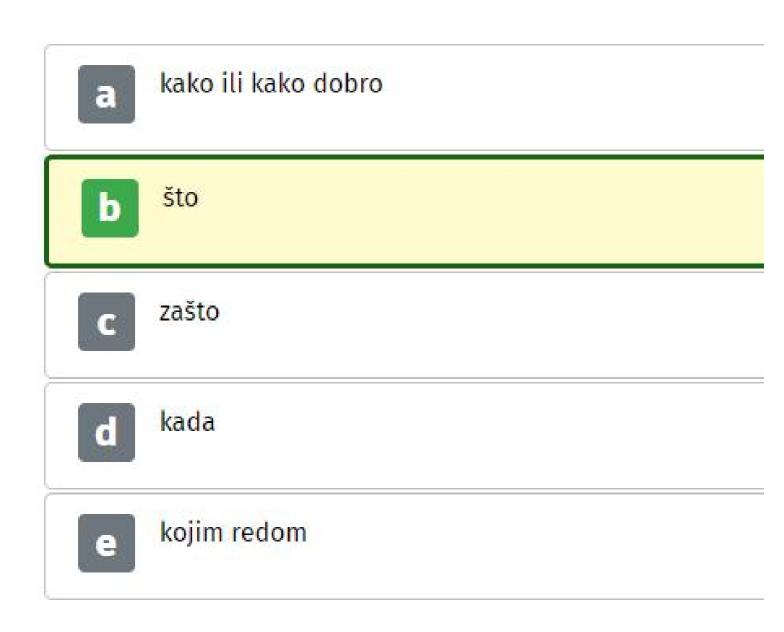


aziva se:

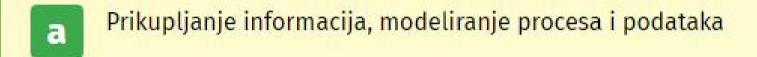
Što od navedenog NIJE istina za razrede izvedene iz sučelja IDataReader

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

### Funkcionalni zahtjevi odgovaraju na pitanje:



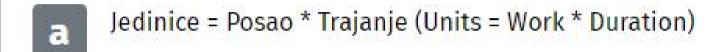
Koje su karatkteristične aktivnosti za fazu (životnog ciklusa) analize?



- Dizajn arhitekture, sučelja i programa
- Konstrukcija, testiranje i implementacija
- Dokumentiranje, održavanje
- Utvrđivanje ciljeva, izrada plana projekta, analiza izvedivosti

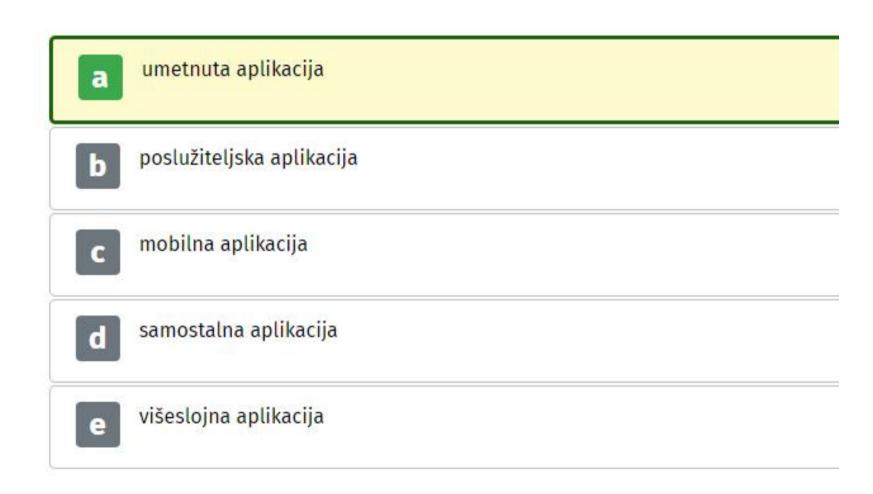
## ld: 9759 Ako se unutar pogleda u MVC aplikaciji kao model koristi razred Mjesto koji ima svojstvo Naziv čija se vrijednost ispisuje unutar pogleda, kako izgleda odsječak takvog pogleda? @Model Mjesto ... @Naziv @model Mjesto ... @Model.Naziv @model Mjesto ... @Naziv @model Mjesto ... @Mjesto.Naziv @Model Mjesto ... @model.Naziv

U planu projekta vrijedi sljedeća formula



- Jedinice = Trajanje + Posao (Units = Duration + Work)
- Trajanje = Posao \* Jedinice (Duration = Work \* Units)
- Posao = Trajanje \* Jedinice (Work = Duration \* Units)
- Posao = Trajanje (Work = Duration)

Koji od navedenih pojmova nije neka vrsta aplikacije?



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

Id: 20095

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

Za koji operacijski sustav će se sustav implementirati?

Što sustav mora raditi?

Kako projektirati sustav?

Tko će koristiti sustav?

Zašto graditi sustav?

.a. 5525 Skup poslanih promjena s jedinstvenim identifikatorom na centraliziranim sustavima za upravljanje izvornim kodom naziva se		
a	check out	
Ь	main trunk	
C	check in	
d	changeset	
e	workspace	

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



#### ld: 38272

Neka su definirane sljedeće hijerarhije:

- Vozilo <- Automobil <- ElektricniAutomobil</li>
- Vozilo <- Motocikl</li>
- Osoba <- Djelatnik <- Direktor

i postupak Djelatnik ZaduzenZa(Automobil a)

Što je od navedenog ispravno

- Func<ElektricniAutomobil, Osoba> func = ZaduzenZa;
- b Func<Vozilo, Direktor> func = ZaduzenZa;
- Func<Vozilo, Osoba> func = ZaduzenZa;
- Func<ElektricniAutomobil, Direktor> func = ZaduzenZa;
- Func<Osoba, ElekricniAutomobil> func = ZaduzenZa;

#### Id: 5419

Kod iterativnog postupka razvoja svaka se iteracija obavlja:

- a kroz faze ekstremnog programiranja
- standardnim životnim ciklusom koji uključuje analizu, oblikovanje, ugradnju i provjeru
- posebnim slučajem životnog ciklusa koji uključuje analizu i provjeru
- kroz faze karakteristične za pojedinu iteraciju
- kroz faze koje se definiraju ovisno o primjeni

#### ld: 20160

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



Što će se ispisati izvođenjem sljedećeg programskog odsječka

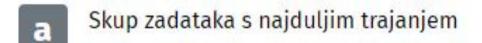
```
int a = 0, b = 0, c;
try {
   try {
       Console.Write("T1 ");
       c = b / a;
   catch (Exception exc) {
       Console.Write("E1 ");
       c = a / b;
   finally {
       Console.Write("F1 ");
catch (Exception exc2) {
   Console.Write("E2 ");
}
finally {
   Console.Write("F2 ");
}
```

- a T1 E2 F2
- b T1 E1 E2
- T1 E1 F1 E2 F2

# ld: 43065 Proces u kojem nakon svake promjene u repozitoriju dolazi do izgradnje programa i pokretanja testova, a zatim i do distrubije programa korisnicima (npr. u smislu objave nove verzije web-aplikacije ili dostur ažuriranja) naziva se Push to publish (PTP) Continuos integration Commit to publish (CTP) Continuos publishing Continuos delivery

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





- Niz primitivnih zadataka koji se ne mogu podijeliti na jednostavnije zadatke
- Put koji treba zaobići
- Niz zadataka koji moraju završiti na vrijeme da bi projekt završio na vrijeme
- Osnova po kojoj se resursu dodjeljuje obavljanje posla

ld: 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();
        Bb=i;
       b.Test();
}
```

a Pozdrav iz B

Pozdrav iz B

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

ld: 43087		
Što se sve može definirati i pretvoriti u Intermediate Language (razredi, strukture, sučelja, delegati) defnirano je s		
Common Language Runtime		
a Common Language Runtime		
<b>b</b> NuGetom		
C Common Type System		
NET Co		
d .NET Standardom		
Base Class Library		
e Base Class Library		

#### ld: 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
URI string

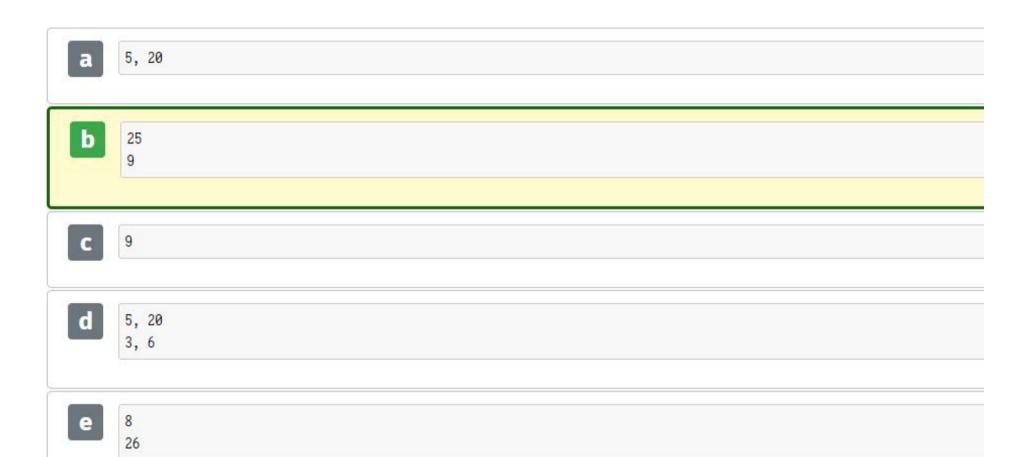
- aplikacija koja pruža neku vrstu informacije
- b skup elemenata ili objekata u jedinstvenoj konfiguraciji koju čine računalni programi, podaci i dokumentacija
- sustav aplikacija za upravljanje ljudskim aktivnostima
- računalom podržanom rješenje jednog ili više poslovnih problema ili potreba
- e računalni 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?		
Spori i sistematični		
Brzi i inkermentalni		
Iterativni i inkrementalni		
b Razvoj po vodopadnom modelu c Iterativni i sistematični d Brzi i inkermentalni		

Što od navedenog nije moguća međuzavisnost zadatka

- količina posla prethodnika je jednaka količini posla sljedbenika
- b završni datum prethodnika utvrđuje početni datum sljedbenika
- početni datum prethodnika utvrđuje početni datum sljedbenika
- završni datum prethodnika jest početni datum sljedbenika
- početni datum prethodnika utvrđuje završni datum sljedbenika

```
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);
}
```



ld: 20174 asp-action u sljedećem odsječku <a asp-action="Create">Unos nove države</a> je primjer korištenja util-atributa css stila b vlastite komponente tag-helpera parcijalnog pogleda

}

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

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

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



## ld: 5465 Planiranje s vrha prema dolje (Top-down) je jedan od pristupa razvoju zadataka i faza u kojem: je osnova pregledni raspored prekretnica a su svi zadaci međusobno ovisni pristupa od specifičnom prema općem se identificiraju glavne faze i rezultati projekta prije dodavanja zadataka potrebnih za završetak tih faza d se identificiraju zadaci najnižeg sloja prije grupiranja u faze

#### Id: 20118

Generičko sučelje definirano modifikatorom in, npr.

public interface IComparer<in T>

je

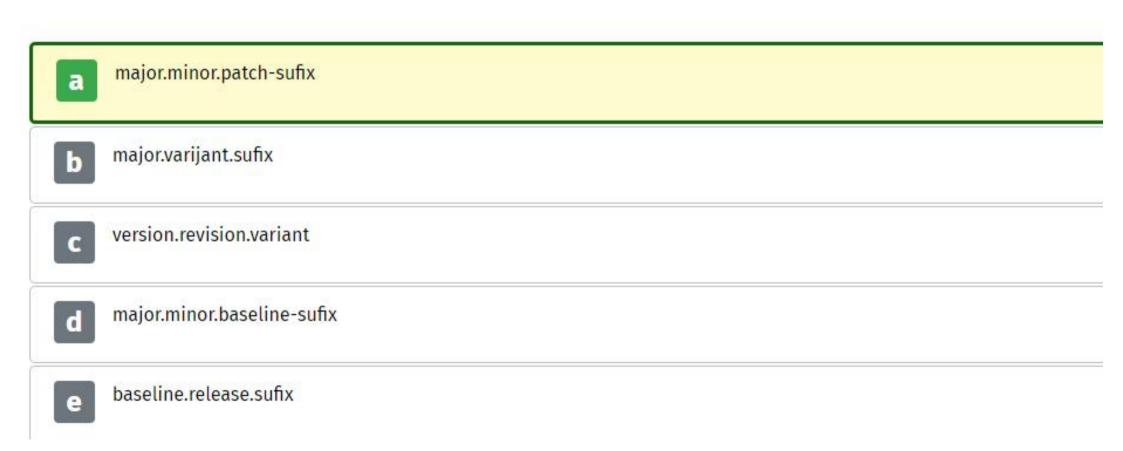
- a invarijantno
- **b** varijantno
- kontravarijantno
- d kovarijantno
- e izlazno-varijantno

ld: 20156 Koja razi sadrži tra	i ina 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 datotel aženu vrijednost, pa se koristi predodređena)?
a	Trace
b	Information
c	Debug
d	Warning
е	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
Ь	ExecuteNonQuery
[c]	ExecuteScalar
d	SetData
е	PostQuery

ld: 38266

Semantic versioning kojeg, između ostalih koristi i .NET Core za označavenje verzije objektne datoteke ima oblik



Što ispisuje sljedeći programski odsječak?

- a ništa
- **b**
- **C** 3
- d f(s[6], "ispit iz RPPP-a")
- е

broj zapisa u tablici Artikl povećat će se za

