

Programsko inženjerstvo

Primjer međuispita 2

				UKUPNO
	I3	I5	I8	
	10	12	16	38
Vrijeme	20min	24min	32min	76 min

UPUTE:

- Ispit se piše na papiru.
- Potrebno je napisati redni broj stranice na svaki od korištenih papira.
- Obvezno napisati redni broj ishoda učenja i zadatka (Prvi dio ili projektni zadatak).

Na ispitu nije dozvoljeno korištenje bilo kakve opreme, usluge ili proizvoda koji nisu izrijekom navedeni kao dozvoljeni u uputama.

Zadaci

Ishod učenja 5 – (5 bodova – minimalni ishod učenja)

Ako je zadan dizajn informacijskog sustava koji omogućava organiziranje volontera i služba za spašavanje i pomoć unesrećenim osoba u području pogođenom razornim potresom koji je potrebno hitno implementirati i isporučiti, objasniti na koji način bi se obavilo procjenjivanje trajanja zadataka u vremenu (satima), definirati zadatke koje je potrebno implementirati, označiti one koje su međusobno ovisni te naznačiti koje od njih je potrebno ranije riješiti kao preduvjete za druge zadatke, te ih procijeniti u vremenu.

Ishod učenja 8 – (16 bodova)

1. (6 bodova – 4 minimalni ishod učenja, 2 željeni ishod učenja)

- a.** Na zadanom programskom isječku prepoznati o kojem od SOLID principa se radi, uočiti pogrešku i predložiti na koji način bi se ispravila. **(2 boda – željeni ishod učenja)**

```
public class Automobil {
    private Proizvodjac proizvodjac;
    private String tip;
    private VrstaMotora vrstaMotora;
    private BigDecimal cijena;

    public Automobil(final Proizvodjac proizvodjac,
                     final String tip,
                     final VrstaMotora vrstaMotora,
                     final BigDecimal cijena)
    {
        this.proizvodjac = proizvodjac;
        this.tip = tip;
        this.vrstaMotora = vrstaMotora;
        this.cijena = cijena;
    }

    public BigDecimal getCijena() {
        return cijena;
    }
}
```

```

public void setCijena(BigDecimal cijena) {
    this.cijena = cijena;
}

public Proizvodjac getProizvodjac() {
    return proizvodjac;
}

public void setProizvodjac(Proizvodjac proizvodjac) {
    this.proizvodjac = proizvodjac;
}

public String getTip() {
    return tip;
}

public void setTip(String tip) {
    this.tip = tip;
}

public VrstaMotora getVrstaMotora() {
    return vrstaMotora;
}

public void setVrstaMotora(VrstaMotora vrstaMotora) {
    this.vrstaMotora = vrstaMotora;
}

public void promijeniVrstuMotora(final VrstaMotora novaVrstaMotora) {
    this.vrstaMotora = novaVrstaMotora;
}

public BigDecimal izracunajFinalnuCijenu() {
    BigDecimal porez = new BigDecimal(25);
    return porez.multiply(porez);
}
}

using System;

public class Automobil
{
    public Proizvodjac Proizvodjac { get; set; }
    public string Tip { get; set; }
    public VrstaMotora VrstaMotora { get; set; }
    public decimal Cijena { get; set; }

    public Automobil(Proizvodjac proizvodjac, string tip,
                    VrstaMotora vrstaMotora, decimal cijena)
    {
        Proizvodjac = proizvodjac;
        Tip = tip;
        VrstaMotora = vrstaMotora;
        Cijena = cijena;
    }

    public void PromijeniVrstuMotora(VrstaMotora novaVrstaMotora)
    {
        VrstaMotora = novaVrstaMotora;
    }
}

```

```
public decimal IzracunajFinalnuCijenu()
{
    decimal porez = 25m;
    return porez * porez;
}
}
```

- b. Neka je potrebno razviti vlastiti programski okvir za razvoj web aplikacija koji će programerima omogućavati implementaciju sigurnosnih aspekata (autentifikaciju i autorizaciju). Programski okvir je potrebno razviti na način da ga je moguće koristiti na najjednostavniji mogući način, uz najmanju moguću količinu utrošenog koda i sa što manje „boilerplate“ koda. Predložiti SOLID principe koje bi bilo potrebno koristiti u ovom slučaju te obrazložiti svoj odabir. Sama definicija SOLID principa bez navođenja detalja gdje bi se oni koristili u sklopu opisanog informacijskog sustava neće se bodovati. **(4 boda – minimalni ishod učenja)**

2. (6 bodova – 4 minimalni ishod učenja, 2 željeni ishod učenja)

- a. Na zadanom programskom isječku prepoznati o kojem od oblikovnih obrazaca (*Design patterns*) principa se radi, uočiti pogrešku i predložiti na koji način bi se ispravila. **(2 boda – željeni ishod učenja)**

```
public final class LogManager {
    private static LogManager instance;

    public LogManager() {
    }

    public static LogManager getInstance() {
        if (instance == null) {
            instance = new LogManager();
        }
        return instance;
    }

    public void log(final String message) {
        System.out.println(message);
    }
}

using System;

public sealed class LogManager
{
    private static LogManager _instance;

    public LogManager() { }

    public static LogManager Instance
    {
        get
        {
            if (_instance == null)
            {
                _instance = new LogManager();
            }
            return _instance;
        }
    }

    public void Log(string message)
```

```
{
    Console.WriteLine(message);
}
}
```

- b. Neka je potrebno razviti sustav notifikacija koji podržava sljedeće tipove notifikacija: e-mail, Skype, Teams, Slack, WhatsApp, Viber, Telegram i Signal. Potrebno je i da sustav podržava dinamičko postavljanje bilo koje od kombinacija tih tipova. (npr. e-mail + Teams, e-mail + WhatsApp + Viber itd.). Predložite oblikovne obrasce (*Design pattern*) koje bi bilo potrebno koristiti da se omogući takvo dinamičko kombiniranje notifikacija uz najmanju količinu utrošenog koda te obrazložite svoj odabir. Sama definicija oblikovnih obrazaca bez navođenja detalja gdje bi se oni koristili u sklopu opisanog informacijskog sustava neće se bodovati. **(4 boda – minimalni ishod učenja)**

3. (4 boda – 2 minimalni ishod učenja, 2 željeni ishod učenja)

- a. Za navedenu metodu „translation“ prikazanu u nastavku napisati *unit* testove koji će provjeravati radi li prevođenje riječi „Algebra“ s hrvatskog na engleski te provjeriti što će se dogoditi u slučaju kad se unese jezik koji je nepodržan, npr. „kineski“. **(3 boda – 1 bod minimalni ishod učenja + 2 željeni ishod učenja)**

```
public class Test {

    public String translation(String word, Language from, Language to) {
        if(Database.isLanguageSupported(from) &&
            Database.isLanguageSupported(to)) {
            return Database.translate(word, from, to);
        }
        else {
            throw new LanguageNotSupportedException("Provided language " +
                from + " or " + to + " not supported!");
        }
    }
}

using System;

public class Test
{
    public string Translation(string word, Language from, Language to)
    {
        if (Database.IsLanguageSupported(from) && Database.IsLanguageSupported(to))
        {
            return Database.Translate(word, from, to);
        }
        else
        {
            throw new LanguageNotSupportedException(
                $"Provided language {from} or {to} not supported!");
        }
    }
}
```

- b. Predložiti *unit* testove koje je moguće napisati za navedenu metodu, osim dva iz prvog dijela zadatka (nije potrebno napisati programski kod testova, već ih samo opisati što bi točno testirali). **(1 bod – minimalni ishod učenja)**

Projektni zadatak

UPUTE:

Projektni zadatak naveden u nastavku se referencira u nekoliko zadataka. Pročitajte projektni zadatak te zatim nastavite na zadatke. Sve odgovore pišite u predviđeni prostor ispod pitanja ili u predloške priložene na sljedećim stranicama.

Projektni zadatak:

Potrebno je implementirati sustav za upravljanje zaposlenicima kroz koji će administrator tvrtke moći unositi i upravljati svim podacima o zaposlenicima, njihovim ugovorima, plaći, beneficijama, te godišnjim odmorima i službenim putovanjima itd.

Vi ste vlasnik produkta (*Product Owner*) a na raspolaganju vam je razvojni tim od 5 ljudi koji uključuje tri programera, jednog testera i jednog devops-a.

Dogovoreno je sa naručiteljem da ćete isporučivati sustav u iteracijama po 3 tjedna.

Vaš tim treba osigurati i isporučiti kompletan sustav što uključuje podizanje infrastrukture, sve softverske komponente, te popratnu dokumentaciju.

Ishod učenja 3 – (10 bodova)

1. **(5 boda – minimalni ishod učenja)** Za zadani projektni zadatak predložite strukturu tima koji će po agilnoj metodologiji organizirati isporuke navedenog projekta, za svaku ulogu navesti karakteristike i aktivnosti koja bi ona obavljala na projektu te procijeniti koliki dio radnog vremena (0-100%) bi ta uloga bila uključena u aktivnosti na projektu.
2. **(2 boda – minimalni ishod učenja)** Predložite agilne ceremonije koje bi se koristile za vrijeme trajanja projekta, opišite karakteristike svake od njih te procijenjeno trajanje.
3. **(3 bod – željeni ishod učenja)** Predložiti artefakte koji bi se isporučivali u prva tri sprinta projekta te opisati što bi sve sadržavali i na koji način bi se prezentirali naručiteljima.

Ishod učenja 5 – (7 bodova)

1. **(3 boda - minimalni ishod učenja)** Za zadani projektni zadatak predložite definirati zadatke koje je potrebno implementirati, rangirati ih po prioritetima te navesti koje sve uloge bi sudjelovale na izvršavanju zadataka i procijeniti ih u vremenu (u satima).
2. **(4 boda – željeni ishod učenja)** Zadatke koje ste procijenili na više od jednog dana potrebno je podijeliti na manje zadatke, opisati ih te ih procijeniti u vremenu (u satima).