



RFP470 Program prekvalifikacija u IT sektoru
JAVA Obuka

OSNOVE JAVA PROGRAMIRANJA

-VEŽBE-

Predavači:

Anićić dr Nenad, redovni profesor, FON

Bošković Zoran, softver inženjer, BREZA

Sadržaj

Sadržaj.....	2
Vežbe.....	3
2.1 Uvod u okruženje	3
Zadatak 1	3
Zadatak 2	3
2.2 Uvod u Javu I	4
Zadatak 1 - klasa, instance varijable, reference varijable, main metoda	4
Zadatak 2 - konstruktori, this, metode, final	4
Zadatak 3 - enkapsulacija.....	5
Rešenja.....	7
2.1 Uvod u okruženje	7
Zadatak 1.....	7
Zadatak 2.....	9
2.2 Uvod u Javu I	9
Zadatak 1.....	9
Zadatak 2.....	10
Zadatak 3.....	12

Vežbe

2.1 Uvod u okruženje

Zadatak 1

Instalirati Javu. Postaviti vrednost za PATH varijablu.

Zadatak 2

Instalirati Netbeans.

2.2 Uvod u Javu I

Zadatak 1 - klasa, instance varijable, reference varijable, main metoda

1. Kreirati novu Java aplikaciju.
2. Kreirati klasu *Racun*, u default paketu.
3. Kreirati instance varijablu *stanje* (tipa double, public vidljivost).
4. Kreirati instance varijablu *brRacuna* (tipa String, public vidljivost).
5. Kreirati klasu *RacunTest*
6. Kreirati main metodu
7. U main metodi kreirati novi objekat *Racun* (pozivom default konstruktora) i dodeliti ga reference varijabli *r*.
8. Postaviti vrednosti za broj računa (po default-u, stanje je inicijalizovano na 0.0).
9. Povećati stanje računa referenciranog varijablom *r* za 500. Odštampati na konzoli trenutno stanje računa.
10. Umanjiti stanje računa referenciranog varijablom *r* za 250. Odštampati na konzoli trenutno stanje računa.
11. Pokrenuti klasu *RacunTest*.

Zadatak 2 - konstruktori, this, metode, final

1. Kreirati klasu *Klijent*.
2. Klijent ima instance varijablu:
 - *ime*, tipa String, public vidljivost
3. Kreirati konstruktor koji prima parametar tipa String, sa nazivom *ime*. Vrednost parametra se dodeljuje instance varijabli *ime*.
4. U klasu *Racun* dodati reference varijablu *vlasnik*, tipa *Klijent*, public vidljivosti.
5. Kreirati dve public metode (nijedna ne vraća vrednost):
 - *uplati(double iznos)*, kojom se stanje povećava za prosleđeni iznos.
 - *isplati(double iznos)*, kojom se stanje umanjuje za prosleđeni iznos.
6. Kreirati tri konstruktora:
 - Konstruktor bez parametara
 - inicijalizuje instance varijablu *stanje* na 300
 - Konstruktor sa parametrom *brRacuna*, tipa String
 - poziva prethodni konstruktor
 - inicijalizuje instance varijablu *brRacuna* na prosleđenu vrednost
 - Konstruktor sa parametrima: *brRacuna* tipa String, *stanje* tipa double, *vlasnik* tipa *Klijent*
 - poziva prethodni konstruktor
 - inicijalizuje ostale instance varijable
7. Dodati final varijablu *dozvoljeniMinus* tipa double.
 - ne inicijalizovati je pri deklaraciji, već u bezparametarskom konstruktoru, na 0.0. Obzirom da svi ostali konstruktori pozivaju ovaj neparametarski inicijalizacija samo u njemu je dovoljna, za sada.

8. Izmeniti main metodu *RacunTest* klase tako da se računu setuje i klijent (kreirati i njega) i da se koristi neki od parametarskih konstruktora.

Zadatak 3 - enkapsulacija

1. Enkapsulirati klasu *Klijent*.
 - promeniti modifikator pristupa za *ime* na *private*.
 - kreirati getter i setter metode za varijablu *ime*.
2. Enkapsulirati klasu *Racun*.
 - promeniti modifikator pristupa instance varijabli na *private*.
 - kreirati getter i setter metode za varijable *brRacuna* i *vlasnik*.
 - za varijablu *stanje* kreirati samo getter metodu, obzirom da za ažuriranje vrednosti već postoje dve metode.
 - za final varijablu *dozvoljeniMinus* kreirati samo getter metodu, a izmeniti bezparametarski konstruktor da prima parametar za inicijalizaciju final varijable. Izmeniti i ostale konstruktore u skladu sa promenom bezparametarskog konstruktora.
3. Implementirati *isplati* metodu da vrši proveru da li je prekoračen dozvoljeni minus i, u zavisnosti od toga, vrši ažuriranje stanja.
4. Izmeniti implementaciju main metode *RacunTest* klase, kako bi odgovarala izmenama kreiranim u prethodnim koracima.

Rešenja

2.1 Uvod u okruženje

Zadatak 1

Download-ovati javu sa <http://www.java.com/en/download/manual.jsp>
Instalirati prateći korake wizard-a.

Postavljanje PATH varijable (Windows):

Windows XP

1. Izabrati Start, izabrati Control Panel. Dupli klik na System, selektovati Advanced tab.
2. Kliknuti na Environment Variables. U sekciji System Variables, naći PATH environment varijablu i izabrati je. Kliknuti na Edit. Ukoliko PATH environment varijabla ne postoji, kliknuti New.
3. U Edit System Variable (ili New System Variable) prozoru, specificirati vrednost PATH environment varijable (putanja do bin podfoldera instalacionog foldera Java-e). Kliknuti OK.

Windows Vista

1. Na desktopu, desni klik na My Computer ikonu.
2. Izabrati Properties iz kontekstualnog menija.
3. Kliknuti na Advanced tab (Advanced system settings link u Visti).
4. Kliknuti na Environment Variables. U sekciji System Variables, naći PATH environment varijablu i izabrati je. Kliknuti na Edit. Ukoliko PATH environment varijabla ne postoji, kliknuti New.
5. U Edit System Variable (ili New System Variable) prozoru, specificirati vrednost PATH environment varijable (putanja do bin podfoldera instalacionog foldera Java-e). Kliknuti OK.

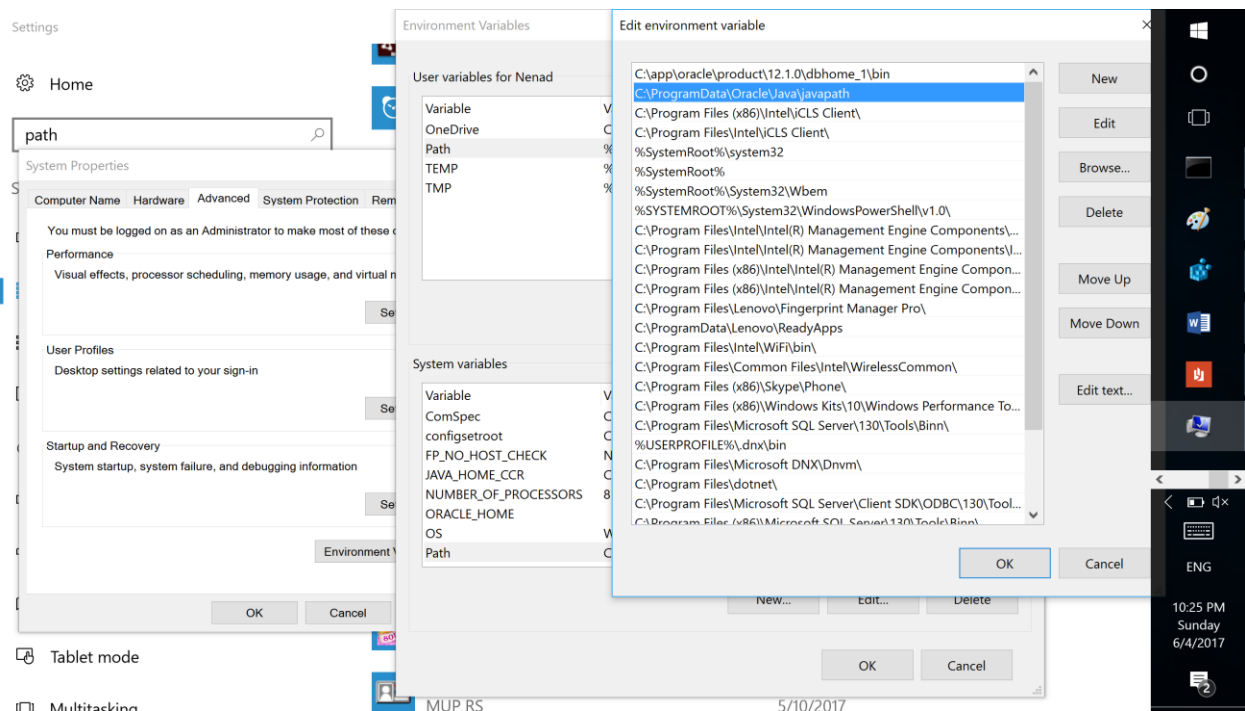
Windows 7

1. Na desktopu, desni klik na My Computer ikonu.

Osnove JAVA programiranja - Vežbe

2. Izabrati Properties iz kontekstualnog menija.
3. Kliknuti na Advanced system settings link.
4. Kliknuti na Environment Variables. U sekciji System Variables, naći PATH environment varijablu i izabrati je. Kliknuti na Edit. Ukoliko PATH environment varijabla ne postoji, kliknuti New.
5. U Edit System Variable (ili New System Variable) prozoru, specificirati vrednost PATH environment varijable (putanja do bin podfoldera instalacionog foldera Java-e). Kliknuti OK.

Windows 10 System Settings



Postavljanje PATH varijable (Solaris i Unix):

Za C shell (csh), editovati startup fajl (~/.cshrc):

```
set path=(/usr/local/jdk1.7.0/bin )
```

Za bash, editovati startup fajl (~/.bashrc):

```
PATH=/usr/local/jdk1.7.0/bin:  
export PATH
```


Osnove JAVA programiranja - Vežbe

Za ksh, startup fajl se naziva po environment varijabli, ENV. PATH se postavlja:

```
PATH=/usr/local/jdk1.7.0/bin:  
export PATH
```

Za sh, editovati profile fajl (~/.profile):

```
PATH=/usr/local/jdk1.7.0/bin:  
export PATH
```

Load-ovati startup fajl and i verifikovati da je putanja podešena java komandom:

Za C shell (csh):

```
% source ~/.cshrc  
% java -version
```

Za ksh, bash, or sh:

```
% . /.profile  
% java -version
```

Zadatak 2

Download-ovati NetBeans sa <https://netbeans.org/downloads/> .
Instalirati prateći korake wizarada.

“JDK 8 and later versions are required for installing and running the Java SE, Java EE and All NetBeans Bundles. You can download [standalone JDK](#) or download the latest [JDK with NetBeans IDE Java SE bundle](#).”

2.2 Uvod u Javu I

Zadatak 1

Osnove JAVA programiranja - Vežbe

```
public class Racun {

    public double stanje;
    public String brRacuna;

}

public class RacunTest {

    public static void main(String[] args) {
        Racun r = new Racun();
        r.brRacuna = "123-456789000123-00";

        r.stanje = r.stanje + 500;
        System.out.println(r.brRacuna + " : " + r.stanje);

        r.stanje = r.stanje - 250;
        System.out.println(r.brRacuna + " : " + r.stanje);
    }
}
```

Zadatak 2

```
public class Klient {

    public String ime;

    public Klient(String ime){
        this.ime = ime;
    }

}

public class Racun {

    public double stanje;
    public String brRacuna;
    public Klient vlasnik;
    public final double dozvoljeniMinus;
```

```
public Racun() {
    this.stanje = 300;
    dozvoljeniMinus = 0.0;
}

public Racun(String brRacuna) {
    this();
    this.brRacuna = brRacuna;
}

public Racun(String brRacuna, double stanje, Klient vlasnik) {
    this(brRacuna);
    this.stanje = stanje;
    this.vlasnik = vlasnik;
}

public void uplati(double iznos){
    stanje += iznos;
}

public void isplati(double iznos){
    stanje -= iznos;
}

public class RacunTest {
    public static void main(String[] args) {

        Klient k = new Klient("Pera");
        Racun r = new Racun("123-456789000123-00", 5000.00, k);

        r.stanje = r.stanje + 500;
        System.out.println(r.brRacuna + " : " + r.stanje);

        r.stanje = r.stanje - 250;
        System.out.println(r.brRacuna + " : " + r.stanje);
    }
}
```

Zadatak 3

```
public class Klijent {

    private String ime;

    public Klijent(String ime){
        this.ime = ime;
    }

    public String getIme() {
        return ime;
    }

    public void setIme(String ime) {
        this.ime = ime;
    }

}

public class Racun {

    private double stanje;
    private String brRacuna;
    private Klijent vlasnik;
    private final double dozvoljeniMinus;

    public Racun(double minus) {
        this.stanje = 300;
        dozvoljeniMinus = minus;
    }

    public Racun(String brRacuna, double minus) {
        this(minus);
        this.brRacuna = brRacuna;
    }

    public Racun(String brRacuna, double stanje, Klijent vlasnik, double minus){
        this(brRacuna, minus);
        this.stanje = stanje;
        this.vlasnik = vlasnik;
    }

}
```

```
public void uplati(double iznos){
    stanje += iznos;
}

public void isplati(double iznos){
    if(stanje - iznos < dozvoljeniMinus){
        System.out.println("Prekoracen dozvoljeni minus!!!");
    }else{
        stanje -= iznos;
        System.out.println("Novo stanje: " + stanje);
    }
}

public String getBrRacuna() {
    return brRacuna;
}

public void setBrRacuna(String brRacuna) {
    this.brRacuna = brRacuna;
}

public Klient getVlasnik() {
    return vlasnik;
}

public void setVlasnik(Klijent vlasnik) {
    this.vlasnik = vlasnik;
}

public double getStanje() {
    return stanje;
}

public double getDozvoljeniMinus() {
    return dozvoljeniMinus;
}
}
```

