

```

/*
Definisati klasu Tacka opisanu sa:
dve realne promenljive (koordinate tačke u ravni),
metodom konstruktorom koji inicijalizuje obe koordinate tačke,
metodom za translaciju tačke u ravni za zadate vrednosti
metodom za prikaz pozicije konkretne tačke u ravni.
*/
class Tacka{
    float x,y;
    Tacka (float a, float b) {           // metod konstruktor
        x = a;
        y = b;
    }
    void transliraj (float promenaX, float promenaY) {
        x = x + promenaX;
        y = y + promenaY;
    }
    String prikaz() {
        return "Tacka sa koordinatama (" + x + "," + y + ")";
    }
    public static void main (String[] args ) {
        Tacka a = new Tacka(1,3);       // kreiranje objekta (instance) a
        System.out.println(a.prikaz( ));

        // (1.0, 3.0)
        a.transliraj(1,2);              // poziv metoda
        System.out.println(a.prikaz( ));

        // (2.0, 5.0)
        a.x = 0;                        // obraćanje promenljivima
        a.y = 0;                         // x i y objekta a
        System.out.println(a.prikaz( ));

        // (0.0, 0.0)
    }
}

```

```

/*
Definisati klasu Voce koja je opisana sa atributima:
ime - tipa String i dva celobrojna atributa grami(za tezinu voca) i kalorijePoGramu,
kao i tri metoda:
dva metoda konstruktora – implicitnim, kao i konstruktorom koji inicijalizuje sve aribute na zadate
vrednosti,
i metodom ukupnoKalorija( ) - koji izračunava ukupne kalorije kao proizvod težine (u gramima) i
kalorija po gramu.
U aplikaciji treba kreirati dva objekta, prikazati njihove atribute i kupnu kalorijsku vrednost.
Na kraju uporediti njihove jedinicne kalorijske vrednosti.
*/

```

```

class Voce {
    String ime;
    int grami;                // težina u gramima
    int kalorijePoGramu;

    Voce ( ) {}                // default konstruktor
    Voce (String i, int a, int b){ // metod konstruktor
        ime=i;
        grami = a;
        kalorijePoGramu = b;
    }
    int ukupnoKalorija() {
        return (grami* kalorijePoGramu);
    }
    void prikaz () {
        System.out.println(ime + ", tezina " + grami + ", kalorije po gramu " + kalorijePoGramu);
        System.out.println("Ukupno kalorija: " + ukupnoKalorija( ));
    }
}

public static void main (String[ ] args) {
    Voce sljiva = new Voce( );
    sljiva.prikaz( );
    sljiva.ime ="sljiva";
    sljiva.grami = 7;
    sljiva.kalorijePoGramu = 80;
    sljiva.prikaz( );
    Voce kruska = new Voce("kruska",50,10);
    kruska.prikaz( );

    // int ukupno = sljiva.grami + kruska.grami;
    // Primer obracanja promenljivim instance i njihovog ucesca u izrazima

    if (sljiva.kalorijePoGramu > kruska.kalorijePoGramu)
        System.out.print("Sljiva je kaloricnije voce");
    else {
        if (sljiva.kalorijePoGramu < kruska.kalorijePoGramu)
            System.out.print("Kruska je kaloricnije voce");
        else
            System.out.print("Kruska i sljiva su jednako kaloricne");
    }
}
}
}

```

/\* Primer definisanja klase za opis cudovista razlicitih boja, polova i sitosti.

Ovo je primer kreiranja (instanciranja) objekta, poziva metoda instance i obracanja promenljivima instance \*/

```
class Cudo {
    String boja;
    boolean sitost;
    char pol;
    Cudo() {} // default konstruktor
    Cudo(String b, boolean s, char p) { // overload konstruktor
        boja = b;
        sitost = s;
        pol = p;
    }
    void hrani() {
        if (sitost)
            System.out.println("Hvala, jelo sam!");
        else {
            System.out.println("Njam, njam...");
            sitost=true;
        }
    }
    void prikaz() {
        System.out.print("Ja sam " + boja + " boje, " + pol + " pola, ");
        if (sitost)
            System.out.println(" sito sam");
        else
            System.out.println(" gladno sam");
    }
    public static void main( String [ ] args) {
        Cudo j = new Cudo( );
        Cudo k = new Cudo("plave", true, 'M');
        j.prikaz( );
        k.prikaz( );
        j.boja="crvene";
        j.sitost=false;
        j.pol='Z';
        j.prikaz( );
        j.hrani( );
        j.prikaz( );
        j.hrani( );
    }
}
```

```
/* Primer definisanja klase Avion, ciji su objekti opisani identifikatorom tipa aviona i sa tri
   koordinate koje odredjuju poziciju aviona u prostoru */
```

```
class Avion {
// promenljive koje opisuju stanje objekata klase Avion - promenljive instance
```

```
String tip;
int x;
int y;
int z;
```

```
// metode koje opisuju ponasanje objekata klase Avion
```

```
Avion pomeri (int a, int b, int c) {
    x=a;
    y=b;
    z=c;
    return this;
}
```

```
// isto na drugi nacin
```

```
void pomeriAvion (int x, int y, int z) {
    this.x=x;
    this.y=y;
    this.z=z;
}
```

```
void prizemlji( ) {
    x=0;
    y=0;
    z=0;
}
```

```
void vratiPos( ) {
    System.out.println("Avion "+tip+" je na poziciji "+ x+ " "+y+" "+z);
}
```

```
boolean proveripos (int a, int b, int c) {
    boolean pom;
    pom=(x==a) && (y==b) && (z==c);
    return pom;
}
```

```
public static void main (String [ ]args) {
```

```
    Avion o1=new Avion( );
    Avion o2=new Avion( );
```

```

    o1.vratiPos( );
    o1.tip="boing727 JAT1";
    o2.tip="AIRBUS France33";
    o1.pomeri(2,4,5);
    o1.vratiPos( );
    o2.pomeriAvion(10,20,25);
    o2.vratiPos( );
    o1.prizemlji( );
    o1.vratiPos( );
    if (o2.proveriPos(10,20,30))
        System.out.println ("Avion "+o2.tip+" se nalazi na datoj poziciji");
    else
        System.out.println ("Avion "+o2.tip+" se ne nalazi na datoj poziciji");
    }
}

```

/\* Jednostavni primer definicije klase Student sa promenljivima:

broj indeksa, ime, smer, godina studija i metodama za:

promenu smeru,

prelazak na sledecu godinu studija,

metodom za upis studenta (konstruktorom) i

metodama za predstavljanje objekata klase Student.

U aplikaciji treba da se:

kreiraju dva objekta klase Student,

prvom studentu se menja smer,

drugi student se upisuje u narednu godinu studija,

na kraju se lista promenjeni spisak studenata.

\*/

```

import java.util.*;
public class Student1{
    private String brInd;
    private String ime;
    private String smer;
    private int godStudija;    // 1, 2, 3
    Student1(String brInd,String ime, String smer) {
        this.brInd=brInd;
        this.ime=ime;
        this.smer=smer;
        this.godStudija=1;
    }
    public void promeniSmer(String noviSmer) {
        smer=noviSmer;
    }
    public void povecajGodStudija( ) {
        godStudija++;
    }
}

```

```

String dajIme( ) {
    return ime;
}
String dajBrInd( ) {
    return brInd;
}
String dajSmer( ) {
    return smer;
}
int dajGodSt( ) {
    return godStudija;
}
void prikaziSt( ) {
    System.out.println("Broj indeksa: " + brInd);
    System.out.println("Ime i prezime " + ime);
    System.out.println("Smer: " + smer);
    System.out.println("Godina studija: " + godStudija);
}

public static void main(String [ ] args) {
    Student1 a;
    a=new Student1("123/06", "Pera Petrovic", "informatika");
    Student1 b=new Student1("124/06", "Ana Stanic", "turizam");    // 2. nacin
    System.out.println("Spisak studenata");
    a.prikaziSt( );
    b.prikaziSt( );
    System.out.println("Unesi ime novog smera za prvog studenta: ");
    Scanner ulaz=new Scanner(System.in);
    String ns=ulaz.nextLine( );
    a.promeniSmer(ns);
    System.out.println("Upisi drugog studenta u sledecu godinu");
    b.povecajGodStudija( );
    System.out.println("Azurirani spisak studenata");
    a.prikaziSt( );
    b.prikaziSt( );
} // main
} // class

```

```

public class Krug1 {
    int x;
    int y;
    int r;

    Krug( ){ }
    Krug(int x, int y, int r) {
        this.x = x;
        this.y = y;
    }
}

```

```

    this.r = r;
}

/* prvi nacin za konstruktor
   Krug(int x, int y) {
       this.x = x;
       this.y = y;
       r = 1;
   }
*/

// drugi nacin za isti konstruktor

    Krug(int x, int y){
        this(x,y,1);      // poziv konstruktora tekuce klase
    }
double obim( ) {
    return 2*r*Math.PI;
}
double povrsina( ) {
    return r*r*Math.PI;
}
void prikazK( ) {
    System.out.println("Krug u poloazaju (" + x + "," + y);
    System.out.println("poluprecnik = " + r);
    System.out.println("Obim = " + obim( ));
    System.out.println("Povrsina = " + povrsina( ));
}
void pomeranje(int a, int b) {
    x = a;
    y = b;
}

public static void main(String [ ] args) {
    Krug k1 = new Krug( );
    Krug k2 = new Krug(2, 2, 4);
    Krug k3 = new Krug(3, 3);
    k1.prikazK( );
    k2.prikazK( );
    k3.prikazK( );
    k2.pomeranje(0, 0);
    k3.pomeranje(-1, 1);
    k1.r = 10;
    System.out.println("Stanje nakon promena");
    k1.prikazK( );
    k2.prikazK( );
    k3.prikazK( );
}
}

```

```
/* Napisati program za obradu rezultata prijemnog ispita na VPS.  
Ulazni podaci koji se dobijaju sa tastature su:  
ime i prezime, broj donetih bodova, broj osvojenih bodova za prvi predmet,  
broj osvojenih bodova za drugi predmet.  
Prikazati rang listu kandidata koja sadrzi:  
    redni broj kandidata, ime i prezime, ukupan broj bodova  
    kandidata, status kandidata (budzet, samofinansiranje ili nije primljen).  
Studenti sa preko 100 bodova su budzetski, sa preko 50 bodova su  
samofinansirajuci, a ostali nisu primljeni.  
*/
```

```
import java.util.*;
```

```
public class Vps {  
    String prIme;  
    int dbod;  
    int obod1;  
    int obod2;  
    Vps( ){}  
    Vps (String prIme,int dbod,int obod1,int obod2) {  
        this.prIme=prIme;  
        this.dbod=dbod;  
        this.obod1=obod1;  
        this.obod2=obod2;  
    }  
    void prikaz1( ) {  
        System.out.println("Ime i prezime: "+ prIme);  
        System.out.println("Ukupan broj bodova: "+ brBod( ));  
        System.out.println(status1(brBod( )));  
    }  
    int brBod( ) {  
        return dbod+obod1+obod2;  
    }  
  
    String status1(int br) {  
        if (br >= 100)  
            return ("Status: Budzet");  
        else {  
            if (br > 50)  
                return ("Status: samofinansiranje");  
            else  
                return ("Nije primljen");  
        }  
    }  
}
```



```

public static void main (String [] args) {
    Scanner ulaz = new Scanner(System.in);
    System.out.println("Unesi broj kandidata");
    int n = ulaz.nextInt( );
    Vps kandidati[ ] = new Vps[n];      // keiranje niza kandidati za sme[taj objekata kase Vps
    System.out.println ("Unesite podatke prijavljenih kandidata: ");
    for (int i=0; i<n; i++) {
    System.out.println( );
    ulaz.nextLine( );
    System.out.println ("Unesite prezime i ime "+ (i+1)+" . kandidata: ");
    String pim=ulaz.nextLine( );
    System.out.println ("Unesite broj donetih bodova "+ (i+1)+ " . kandidata");
    int br0 = ulaz.nextInt( );
    System.out.println ("Unesite broj osvojenih bodova za 1. predmet"+ (i+1)+" . kandidata");
    int br1 = ulaz.nextInt( );
    System.out.println ("Unesite broj osvojenih bodova za 2. predmet"+ (i+1) + " . kandidata");
    int br2 = ulaz.nextInt( );
    kandidati[i] = new Vps(pim, br0, br1, br2); // kreiranje objekata klase Vps
    }

    // sortiranje po ukupnom broju bodova

    Vps obj=new Vps( );    // pomocni objekat
    for (int i=0; i<(n-1); i++)
        for(int j=i+1; j<n; j++)
            if ((kandidati[i].brBod( )) < (kandidati[j].brBod( )))
                {
                    obj=kandidati[i];
                    kandidati[i]=kandidati[j];
                    kandidati[j]=obj;
                }

    System.out.println("\nRang lista : ");
    for (int i=0; i<n; i++) {
        System.out.println( (i+1) + " . kandidat je: ");

        kandidati[i].prikaz1( );

    } // for
} // main
} // class

```

```
/* Vozila mogu biti putnicka ili teretna.  
Sva vozila imaju id i sopstvenu tezinu,  
max broj putnika i prosečnu tezinu putnika (za putnicka)  
odnosno tezinu tereta (za teretna).  
Izlistati sva vozila, zatim samo ona vozila koja mogu preći  
preko mosta zadate nosivosti.  
*/
```

```
import java.util.*;  
class Vozila {  
    String id;  
    int tezinu;    // sopstvena tezinu vozila  
    Vozila (){}    // default konstruktor  

```

```
class PVozilo extends Vozila{  
    int brojPut;  
    int prosTez;  
    PVozilo(){}  
  
    PVozilo (String s,int t,int brP, int pT){  
        super(s, t);  
        brojPut=brP;  
        prosTez=pT;  
    }  
    int ukTez(){  
        return super.ukTez() + brojPut*prosTez;  
    }  
}
```

```
/* ili ovako:
```

```
    int pom;  
    pom = super.ukTez() + brojPut*prosTez;  
    return pom;  
*/  
}  
void prikaz(){  
    super.prikaz();  
}
```

```

System.out.println("Broj putnika " + brojPut);
System.out.println("Prosečna težina putnika " + prosTez);
System.out.println("Ukupna težina " + ukTez());
}
} // class

```

```

class TVozilo extends Vozila {
    int terTez;

    TVozilo(){}
    TVozilo (String s, int t, int tT){
        super(s,t);
        terTez=tT;
    }

    int ukTez(){
        return super.ukTez() + terTez;
    }
}

```

```

void prikaz(){
    super.prikaz();
    System.out.println("Težina tereta " + terTez);
    System.out.println("Ukupna težina " + ukTez());
}
} //class

```

```

class Most1{
    public static void main (String [ ]args) {
        Scanner ulaz=new Scanner(System.in);
        System.out.print("Unesite broj vozila: ");
        int broj= ulaz.nextInt( );
    }
}

```

```

Vozila nizV[ ] = new Vozila[broj]; //kreiranje niza objekata

```

```

System.out.println("Unesite podatke o vozilima: ");
ulaz.nextLine( );
for(int i=0;i<broj;i++) {
    System.out.println("Registarski broj vozila: " );
    String reg = ulaz.nextLine( );
    ulaz.nextLine( );
    System.out.println("Težina vozila: " );
    int tez= ulaz.nextInt( );
    ulaz.nextLine( );
    System.out.println("Da li je putnicko(P) ili teretno(T)? " );
    String odg =ulaz.nextLine( );
    if (odg.charAt(0)=='P') {
        System.out.println("Prosečna težina putnika: " );
        int ptez= ulaz.nextInt();
    }
}
}

```

```

        System.out.println("Broj putnika: " );
        int brp= ulaz.nextInt();
        nizV[i] = new PVozilo(reg,tez,brp,ptez);
    }
    else {
        System.out.println("Tezina tereta : " );
        int ttez = ulaz.nextInt( );

        nizV[j]=new TVozilo(reg,tez,ttez);
    } //else
} // for

System.out.println("\nSpisak svih vozila");

for (int i=0; i<broj; i++)
    nizV[i].prikaz( );           // POLIMORFIZAM na delu!
System.out.println("\nNosivost mosta: ");
int nosivost= ulaz.nextInt();

System.out.println("Vozila koja mogu preci most nosivosti " + nosivost + " su: ");
for (int i=0; i<broj; i++) {

    if (nizV[i].ukTez( ) < nosivost)           // POLIMORFIZAM na delu!
        nizV[i].prikaz( );

} //for
} //main
} //class

```

```

/* Napisati JAVA aplikaciju za potrebe jedne knjižare. Vode se sledeci podaci o knjigama:
    naziv, glavni autor, izdavac, cena, broj primeraka.
    Na cenu svih knjiga odobrava se promenljivi popust (rabat).
    Ako je knjiga strucna, prati se još i
        tematska oblast kao i poseban procenat popusta.
    Pored unosa i listanja stanja u knjižari, omoguciti:
        listanje svih knjiga iz oblasti RACUNARSTVO ciji je broj primeraka 1, zatim
        listanje strucnih knjiga sa popustom vecim od 10%,
        upit o bilo kojoj knjizi na osnovu autora i naziva - ispisati njenu aktuelnu (prodajnu) cenu i broj
        primeraka u knjižari.    (2.kolokvijum)
*/

```

```

import java.util.*;
class Knjiga {
    String naziv;
    String autor;
    int cena;
    int brp;

```

```

Knjiga(String naziv,String autor,int cena,int brp) {
    this.naziv=naziv;
    this.autor=autor;
    this.cena=cena;
    this.brp=brp;
}
void prikaz( ) {
    System.out.println("\nNaziv knjige: "+naziv);
    System.out.println("Autor knjige: "+autor);
    System.out.println("Cena knjige: "+cena);
    System.out.println("Broj primeraka knjige: "+brp);
}

float konCena(int vr) {
    return (cena - cena*(float)vr/100);           //cena knjige umanjena za rabat vr
}

void rac1( ){ }
void popust10( ){ }
}

class Struck extends Knjiga {
    String oblast;
    int popust;

    Struck(String naziv,String autor,int cena,int brp,String oblast,int popust) {
        super(naziv,autor,cena,brp);
        this.oblast = oblast;
        this.popust = popust;
    }
    void prikaz( ) {
        super.prikaz( );
        System.out.println("Oblast knjige: " + oblast);
        System.out.println("Popust na knjigu: " + popust);
    }
    void rac1( ){
        if((oblast.equalsIgnoreCase("RACUNARSTVO"))&&(brp == 1))
            prikaz( );
    }
    void popust10( ) {
        if(popust > 10)
            prikaz( );
    }
    float konCena(int vr){
        return (super.konCena(vr)- super.konCena(vr)*popust/100);

        /* cena knjige umanjena za rabat i popust od 10%

```

```

    Ili:
    int pomCena = super.konCena(vr);
    pomCena -= pomCena*popust/100;
    return pomCena;
    */
}
}

```

```

class Knjizara {
public static void main (String[ ]args) {
    Scanner ulaz = new Scanner (System.in);
    System.out.print("Unesite vrednost rabata u %: ");
    int vr = ulaz.nextInt( );
    System.out.print("\nUnesite ukupan broj knjiga u knjizari: ");
    int n = ulaz.nextInt( );
    ulaz.nextLine( );
    Knjiga[ ] knjige = new Knjiga[n];

    for(int i=0; i<n; i++) {
        System.out.print("\n\nUnesite naziv knjige: ");
        String naz = ulaz.nextLine( );
        System.out.print("\nUnesite autora knjige: ");
        String aut = ulaz.nextLine( );
        System.out.print("\nUnesite cenu knjige: ");
        int cen = ulaz.nextInt( );
        System.out.print("\nUnesite broj primeraka knjige: ");
        int bp = ulaz.nextInt( );
        System.out.print("\nDa li je knjiga strucna? (DA/NE)");
        String odg = ulaz.nextLine( );
        if(odg.equalsIgnoreCase("DA")) {
            System.out.print("\nUnesite oblast knjige: ");
            String obl = ulaz.nextLine( );
            System.out.print("\nUnesite popust: ");
            int pop = ulaz.nextInt( );
            knjige[i]=new Struck(naz,aut,cen,bp,obl,pop);
        }
        else
            knjige[i] = new Knjiga(naz,aut,cen,bp);
    } // for

    System.out.println("\n\nSPISAK SVIH KNJIGA U KNJIZARI");
    for(int i=0; i<n; i++)
        knjige[i].prikaz( );

    System.out.println("\n\nKNJIGE IZ OBLASTI RACUNARSTVO SA BR. PRIMERAKA 1");
    for(int i=0; i<n; i++)
        if (knjige[i] instanceof Struck)
            knjige[i].rac1( );
}
}

```

```

System.out.println("\n\nKNJIGE SA POPUSTOM VECIM OD 10%");
for(int i=0; i<n; i++)
    if (knjige[i] instanceof StrucK)
        knjige[i].popust10( );

while(true) {
    int indikator=0;
    System.out.print("\n\nUnesite naziv trazene knjige: ");
    String nk = ulaz.nextLine();
    System.out.print("\n\nUnesite autora trazene knjige: ");
    String ak = ulaz.nextLine( );

    for(int i=0; i<n; i++) {
        if (((knjige[i].naziv.equalsIgnoreCase(nk))&& ((knjige[i].autor).equalsIgnoreCase(ak))) {
            indikator = 1;
            System.out.println ("\n\nCena knjige "+nk+" je: " + knjige[i].konCena(vr));
            System.out.println ("Broj primeraka knjige "+nk+" je: " + knjige[i].brp);
        }
    } // for

    if (indikator = 0)
        System.out.println ("Ove knjige nema u knjižari!");

    System.out.println ("Ako ste završili ukucajte 'KRAJ', ako niste 'ENTER'");
    String kraj=ulaz.nextLine( );
    If (kraj.equalsIgnoreCase("KRAJ"))
        break;
} // while
}
}

```

/\* ISPITNI ZADATAK (2.kolokvijum)

Napraviti Java aplikaciju za vodjenje biblioteke.

Sve publikacije u biblioteci se dele na knjige i casopise.

Prate se sledeci podaci o publikacijama:

bibliotecki broj, naziv publikacije i broj slobodnih primeraka.

Kod knjiga se vodi jos naziv glavnog autora.

Kod casopisa se vodi jos redni broj primerka i godina izdanja.

Obezbediti:

listanje biblioteckog fonda,

listanje svih casopisa iz 2018. Godine,

izdavanje primerka publikacije ako je slobodna.

\*/

```
import java.util.*;
```

```
class Publikacija {
```

```
    int bibBr;
```

```

String nazivP;
int brSl;

Publikacija(int bibBr, String nazivP, int br){
    this.bibBr = bibBr;
    this.nazivP = nazivP;
    brSl = br;
}
void prikaz(){
    System.out.println("\nBibliotecki broj: " + bibBr);
    System.out.println("Naziv publikacije: " + nazivP);
    System.out.println("Broj slobodnih primeraka: " + brSl);
}
void izdavanje(){
    if(brSl>0){
        System.out.println("\nIZDATA JE SLEDECA PUBLIKACIJA:");
        System.out.println("Bibliotecki broj: " + bibBr);
        System.out.println("Naziv publikacije: " + nazivP);
        brSl--;}
    else
        System.out.println("\n Nema sobodnog primerka trazene publikacije!");
}
}
class Knjiga1 extends Publikacija {
    String nazivA;

    Knjiga1 (int bibBr, String nazivP, int br, String autor){
        super(bibBr, nazivP, br);
        nazivA=autor;
    }
    void prikaz( ){
        super.prikaz( );
        System.out.println("Naziv autora: " + nazivA);
    }
}
class Casopis extends Publikacija {
    int rbrC;
    int godl;

    Casopis (int bibBr, String nazivP, int br, int rbrC, int godl){
        super(bibBr, nazivP, br);
        this.rbrC=rbrC;
        this.godl=godl;
    }
    void prikaz( ){
        super.prikaz( );
        System.out.println("Redni broj casopis: " + rbrC);
    }
}

```



```

    System.out.println("Godina izdavanja: " + godl);
}

boolean proveraGod (int gg ){
    if (godl == gg)
        return true;
    else
        return false;
}
}
class Biblioteka {
    public static void main (String[ ]args){

        Scanner ulaz = new Scanner(System.in);
        System.out.print("Unesite broj razlicitih publikacija: ");
        int n= ulaz.nextInt( );
        Publikacija [ ] nizPub = new Publikacija[n];
        for (int i=0; i<n; i++){
            System.out.print
            ("\nUnesite bibliotecki broj publikacije: ");
            int bb= ulaz.nextInt( );
            System.out.print("Unesite naziv publikacije: ");
            String naz =ulaz.nextLine( );
            System.out.print
            ("Unesite broj slobodnih primeraka publikacije: ");
            int sp = ulaz.nextInt( );
            System.out.print
            ("Da li je publikacija knjiga ili casopis? (K/C)");
            String odg = ulaz.nextLine( );
            if (odg.equalsIgnoreCase("K")) {
                System.out.print("Unesite naziv autora knjige: ");
                String na = ulaz.nextLine( );
                nizPub[i] = new Knjiga1(bb,naz,sp,na);
            }
            else{
                System.out.print("Unesite redni broj casopisa: ");
                int rbc = ulaz.nextInt( );
                System.out.print("Unesite godinu izdavanja casopisa: ");
                int gi = ulaz.nextInt( );
                nizPub[i] = new Casopis(bb,naz,sp,rbc,gi);
            }
        }
        // for - kraj unosa biblioteckog fonda

        // listanje publikacija
        System.out.println
        ("\nU BIBLIOTECKOM FONDU SE NALAZE SLEDECE PUBLIKACIJE:");
        for (int i=0; i<n; i++)
            nizPub[i].prikaz( );
    }
}

```

```

// listanje casopisa iz zadate godine

System.out.println
    ("\n Spisak casopisa iz 2018. godine:");
Casopis s;
for (int i=0; i<n; i++){
    if (nizPub[i] instanceof Casopis){
        s=(Casopis)nizPub[i];

        if (s.proveraGod(2018))
            s.prikaz( );
    }
}

// izdavanje publikacija

while (true){ // beskonacna petlja za visestruki unos brojeva publikacija, sve dok se ne unese NE
    System.out.println
        ("\n\nDA LI ZELITE IZDAVANJE PUBLIKACIJE? (DA/NE)");
        String odg = ulaz.nextLine( );
    if (odg.equalsIgnoreCase("DA")) {
        System.out.print("\nUNESITE BROJ PUBLIKACIJE KOJU ZELITE: ");
        int brp = ulaz.nextInt( );
        int indTrazenja = 0;
        for (int i=0; i<n; i++) {
            if (brp == nizPub[i].bibBr) {
                nizPub[i].izdavanje( );
                indTrazenja = 1;
                break;
            }
        } // for
        if (ind == 0)
            System.out.println("\n Nije nadjena publikacija!");
        } // if
    else {
        if (odg.equalsIgnoreCase("NE"))
            break; // kraj programa
        } // else
    } // while
} // main
} //class

```

/\* Kreirati JAVA aplikaciju za potrebe zoološkog vrta.

Prate se sledeci podaci o životinjama:

šifra vrste, naziv vrste, redni broj primerka, starost primerka, zemlja porekla.

Omogućiti unos testnih podataka o svim životinjama u jednom zoo vrtu sa tastature.

Izlistati podatke o primercima vrste 'zebra' (redni broj, starost, zemlja porekla) i dati podatak koliko ih je ukupno.

Na kraju izlistati sve životinje poreklom iz 'Egipta' starije od 10 godina.

**Test aktivnosti 5.6.2019.**

\*/

```
import java.util.*;
class Zivotinje1{
    int sifv;
    String nazivv;
    int rbp;
    int starostp;
    String zemljap;
    Zivotinje(int sifv,String nazivv,int rbp,int starostp,String zemljap){
        this.sifv=sifv;
        this.nazivv=nazivv;
        this.rbp=rbp;
        this.starostp=starostp;
        this.zemljap=zemljap;
    }
    boolean zebra( ) {
        if (nazivv.equalsIgnoreCase("zebra"))
            return true;
        else
            return false;
    }
    void zebraP( ) {
        System.out.println("\nRedni br.: "+rbp);
        System.out.println("Starost: "+starostp);
        System.out.println("Zemlja porekla: "+zemljap);
    }
    boolean egipat10( ) {
        if ((zemljap.equalsIgnoreCase("Egipat")) && (starostp>10))
            return true;
        else
            return false;
    }
    void prikaz( ) {
        System.out.println("\nSifra vrste: "+sifv);
        System.out.println("Naziv vrste: "+nazivv);
        System.out.println("Redni broj: "+rbp);
        System.out.println("Starost: "+starostp);
        System.out.println("Zemlja porekla: "+zemljap);
    }
}
```

```

public static void main (String [ ] args) {
    Scanner ulaz = new Scanner(System.in);
    System.out.print ("Unesite koliko ima zivotinja u ZOO vrtu: ");
    int n = ulaz.nextInt( );
    Zivotinje[ ] primerci = new Zivotinje[n]; // niz objekata primerci
    for (int i=0; i<n;i++) {
        System.out.print("Sifra vrste: ");
        int sz = ulaz.nextInt( );
        System.out.print("Naziv vrste: ");
        String nz = ulaz.nextLine( );
        System.out.print("Redni broj: ");
        int rb = ulaz.nextInt( );
        System.out.print("Starost: ");
        int str = ulaz.nextInt();
        System.out.print("Zemlja porekla: ");
        String zp=ulaz.nextLine( );
        primerci[i]=new Zivotinje(sz,nz,rb,str,zp);
    }
    System.out.println("\n\nPODACI O ZEBRAMA:");
    int brojac=0;
    for (int i=0; i<n;i++){
        if (primerci[i].zebra( )) {
            primerci[i].zebraP( );
            brojac++;
        }
    }
    System.out.println("\nUkupno zebri ima: " + brojac);

    System.out.println("\n\nPODACI O ZIVOTINJAMA IZ EGIPTA STARIJIM OD 10 GOD.:");
    for (int i=0; i<n;i++) {
        if (primerci[i].egipat10( ))
            primerci[i].prikaz());
    }
}
}

```