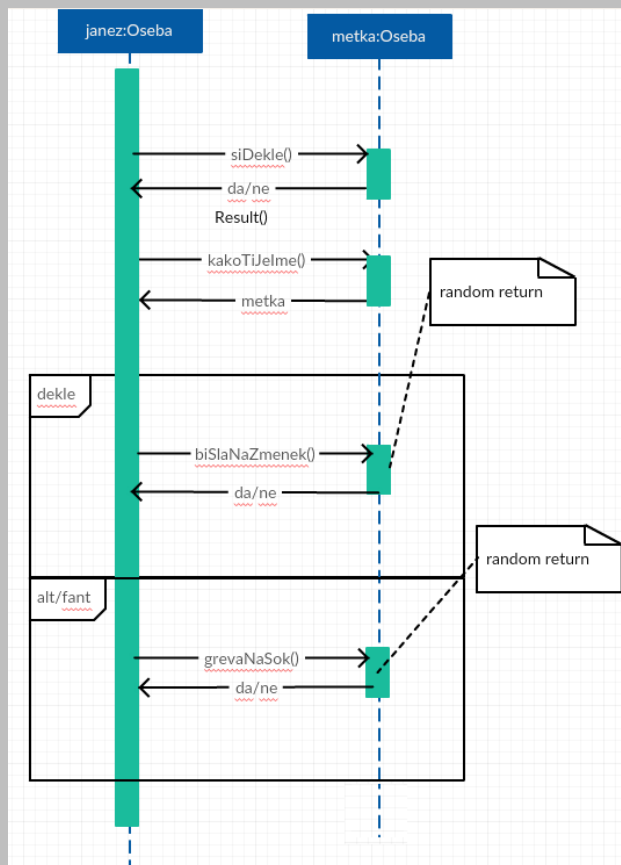


Teme:

- **OOP: konstrukcija razreda, konstrukcija objekta, raba objekta**
 - **Razred, objekt**
 - **Lastnosti**
 - **Metode**
 - **Geterji, seterji(vračalci, nastavljalci)**
 - **Konstruktorji**
 - **Uporabniško definirane metode**
 - **Metoda toString**
 - **Diagrami: razredni diagram, objektni diagram, sekvenčni diagram**

Želeli bi spisati program, ki bi bil sposoben med dvema objektoma izvesti konverzacijo, kot je podana z naslednjim sekvenčnim diagramom (diagramom zaporedja, SD- sequence diagram):



<http://creately.com>, marec 2017)

- črtkana črta predstavlja časovno os (čas teče navzdol)
- sporočila med objektoma so označena s puščicami (realizacija:metode)
- zeleno je označena aktivnost (dogajanje)
- uokvirjeni bloki označujejo pogojno izvajanje medsebojne komunikacije (drugače glede na to, ali je cilj komunikacije dekle ali kaj drugega).

Janez in Metka sta primerka tipa Oseba (objekta). Vsaka oseba mora poleg zahtev v SD diagramu znati odgovarjati na atomarna vprašanja o imenu, priimku in spolu (metode vrste get). Če je vprašana, zna oseba povedati tudi kombinacijo svojega priimka in imena.

Poleg v besedilu omenjenih lastnosti vsaki osebi pripišemo še oddelek/razred in enolično vpisno številko.

Omejitve pri določanju podatkov za osebo so: ime, priimek : najmanj 3 znaki, največ 15 znakov, razred je nekaj izmed {g1a,g1b,g2a,g2b,g3a,g3b,g4a,g4b}, vpisna številka je cela 5 mestna številka iz intervala od 10000 do 99999; pri tem dve osebi ne moreta imeti iste vpisne številke. Spol je element množice {fant,dekle}

(slika je na hitro narejena online na

Naloga 1

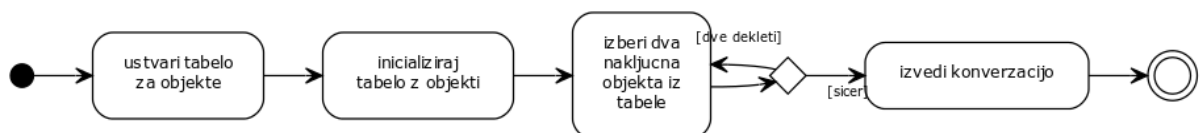
- Kreirajte razred in opredelite lastnosti (opredeliti v kodi pomeni: definirati oz. spisati. Vse lastnosti naj bodo striktno privatne.
- Za vsako atomarno lastnost definirajte nastavljalce in vračalce; pri nastavljalcih upoštevajte zahtevane omejitve; v primeru, da je dano ime, priimek prekratko, ga podaljšajte na minimalno zahtevano dolžino z dodajanjem naključnih numeričnih znakov; če sta predolga ju skrajšajte na maksimalno zahtevano dolžino.
- Dodajte konstruktor; za nastavljanje vrednosti podatkov naj uporabi že definirane nastavljalce; konstruktorju podate ime, priimek, spol in razred.
- Dodajte konstruktor kot nadomestilo privzetemu konstruktorju; ime, priimek, spol, razred naj se generirajo naključno ali prevzamejo iz nekega prej definiranega slovarja.
- Dodajte preostale metode, ki so zahtevane z besedilom in z diagramom z opisom konverzacije.

Naloga 2

- Izrišite razredni diagram razreda Oseba.
- Izrišite objektna diagrama (slika: objekta Janez, Metka; enak diagram kot razredni, samo da odraža trenutno stanje – atributi oz. lastnosti imajo vrednosti. Vzemite trenutek po inicializaciji obeh objektov, pa preden se prične konverzacija med njima (sporočila med objektoma zanemarimo).

Naloga 3

Spišite javanski program, ki bo ustvaril kolekcijo (tabelo) 20 oseb, nato bo iz tabele vzel 2 naključni osebi in izvedel konverzacijo, kot je opisana na diagramu (SD). Konverzacijo razširite z vljudnostjo, pri kateri se fant, ki začenja pogovor najprej predstavi. (op.: tabela oseb naj bo razredna lastnost). Celoto aktivnosti programa ponazarja spodnji diagram (diagram aktivnosti, activity diagram), ki v primeru izbire dveh deklet ponovi izbiro oseb:



(slika je na hitro narejena online na: <https://yuml.me/diagram/scruffy/activity/draw>, marec 2017)

Naloga 4

- a) Spišite javanski program, ki bo ustvaril dva objekta vrste Oseba; prvi naj bo mirko(ime:Mirko), drugi sabina(ime:Sabina). Nato izpišite oba objekta kot:
- ```
System.out.println(sabina); in System.out.println(mirko);
```
- Ali se objekta izpišeta ali program pri izpisu javi napako?
  - Kako se oba objekta izpišeta ?
  - Ali se pri ponovnem zagonu izpišejo za ista objekta iste vrednosti? Kako si to razlagate?
- b) V definicijo razreda Oseba dodajte metodo

```
public String toString(){
 return "oseba:< "+ime+", "+priimek+", "+spol+">";
}
```

Pri predpostavki, da so v Oseba definirane objektne lastnosti poimenovane ime, priimek, spol.

Ponovno poskusite z izpisom iz dela a) te naloge in podajte odgovore na vsa tri vsebovana podvprašanja.

## Naloga 5

Spišite javanski program Naloga5, ki bo implementiral tabelo 20-tih oseb (kot v naloga 3). Programu dodajte metodo izpisiVseOsebe. Ta bo vse osebe tabele 20-tih oseb (podane kot argument metodi ?) izpisala na zaslon s pomočjo metode toString/0, kot je definirana v 4.b) .

## Naloga 6

Javanski program Naloga6 implementira tabelo kot objektno lastnost. Glavna metoda danega programa v komentarjih opredeli delovanje programa. Pri tem strukture razreda Naloga6 ne spreminjamo, temveč uporabimo glavno metodo zgolj kot ovojnico za kreiranje objekta s katerim oz. nad katerim bomo izvedli s predhodnim besedilom zahtevano. (zasledujemo filozofijo, da v OO svetu obstajajo zgolj objekti in sporočila, prvega je pač nekako potrebno narediti; in to naredimo z zagonskim razredom oz. njegovo zagonsko metodo):

```
class Naloga6{

 private Oseba[] tabOseb = new Oseba[20];

 public static void main(String[] args){
 /* vse dovoljene spremembe so znotraj tega bloka kode */
 // naredi 20 oseb in jih vpiši v this.tabOseba
 // izpiši vseh 20 oseb iz tabele
 }
}
```