

Intervju pitanja iz oblasti HTML5,CSS3,JS ES6&5

JQuery i OOP u Javi

Skripta sa pitanjima i odgovorima koju sam napravio za podsecanje.

Q – Question, A- Answer

HTML5 & CSS3 pitanja

Q: Sta je HTML5?

A: HTML ili Hyper Text Markup Language je markup jezik za prikaz web stranice. Verzija HTML 5.2 je trenutno najnovija verzija izasla 2017.

Q: Razlika izmedju HTML 5 I HTML 4?

A: HTML 5 update je doneo razne novine I neke promene u pisanju elemenata. Primer novina su audio I video tagovi koji su zamenili flash player reprodukciju video I audio sadrzaja, drag & drop, HTML 5 koristi SQL za skladište podataka lokalno dok je HTML 4 koristio kolacice, HTML 5 dozvoljava da js radi u pozadini dok HTML 4 ovo nije dozvoljavao, Geolokacija je novina u HTML 5 koriscenjem Geo api u js-u...

Q: Liste u HTML-u?

A: U HTML-u postoje ordered lists `` I unordered lists ``. Svaki child liste je `` - stavka liste.

Q: Sta znaci `<!DOCTYPE html>` ?

A: `<!DOCTYPE html>` se nalazi na vrhu html fajla I služi da web stranica prepozna HTML dokument.

Q: Sta su tagovi u HTML-u?

A: HTML tagovi su definisani pocetnim tagom, nekim sadrzajem, završim tagom:
`<tag> test </tag>`

Q: Nested elementi?

A: Pitanje je glupo I da ne dodje do zabune to je samo parent I child element (tag u tagu).

Q: Atributi u HTML-u?

A :Samo daju dodatne informacije o elementu primeri atributa su type,href,src,style...

Q: Headings/poglavlja?

A: Heading su naslovi u HTML I po velicini su od najveceg <h1> pa do najmanjeg <h6>

Q & A :Paragraf? <p></p> ne moram ni da pisem o njemu

Q: Formatiranje u HTML-u?

A:

- - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Q: Slike tag?

A: mora da ima dva atributa src i alt.

Q: Kreiranje tabela?

A: Uz pomoc tagova <table> <tr> - table row, <td> - table data

←! Ja ne znam sta ce pitaju za ovaj HTML ->

Q: Forme u HTML-u?

A: <input> sa obaveznim tipom kao na primer checkbox itd.

Q: HTML <iframe>

A: <iframe> omogućava da se jos jedan document prikazuje na HTML stranici.

Q: Blok I inline elemnti u HTML-u?

A: <div>

Q: Sta je CSS3?

A: CSS je stylesheet jezik za uređivanje izgleda HTML stranice. CSS je skracenica za Cascade Style Sheet. CSS3 je poslednja verzija(u trenutku pisanja 2021. godine). Razlika izmedju CSS2 i CSS3 je u uglavnom ta sto je CSS3 podeljen u dva modula.

Q: Razlika izmedju ID i CLASS u HTML-u i CSS-u?

A: ID je unique(jedinstven) atribut dok class to nije, ovo znaci da ukoliko elementu stavimo ID atribut ne mozemo drugom elementu dodeliti isti ID dok kod CLASS ovo nije slucaj. ID selektor '#' class selector '.'.

Q: Pozicioniranje elementa u CSS3?

A: Pozicioniranje elementa u CSS3 izvodi se uz position property-a. Position property moze da ima sledece vrednosti:

Position: static , staticka pozicija stranice po default-u, normalna pozicija kao pri normalnom toku dokumenata(normal flow), ne moze se uticati na promenu pozicije koriscenjem left, right, bottom i top.

Position: relative, relativna pozicija od normalne pozicije, promena pozicije se vrši uz pomoc left,right,top,bottom.

Position: fixed, iskljucena iz normalnog toka(removed from normal flow), moze da prelazi preko druge elemente stranice(overlap), mozemo koristiti i z-index koji moze imati negative value da bi odlucili koji element ce da overlap.

Position: absolute: Apsolutna pozicija, iskljucena iz normalnog toka, moze da prelazi preko drugih elemenata.

Position: sticky: na pocetku kao static reaguje na scroll i u slucaju scroll-a postaje fixed.

Q: CSS liste

A: CSS list property služi za menjanje markera liste, u HTML-u postoje uredjenje liste i liste koje nisu uredjene. List-style-type: <value> menja tip liste za uredjene liste default je numericko uredjenje(1,2,3,4...n) dok za neuredjene je popunjen krug. Takođe uz pomoć CSS-a možemo numerisati listu uz pomoć slika.

Q: CSS margine

A: CSS margine služe za određivanje praznog prostora HTML elementa. Možemo koristiti:

Margin(određivanje celokupnog praznog prostora oko HTML elementa), margin-top, margin-right, margin-bottom, margin-left.

Q: CSS float

A: CSS float property služi za pozicioniranje elemenata u odnosu na druge HTML elemente. Moguće vrednosti su: left, right, none(default) i inherit(nasledjuje parent podesavanje).

Da bi izbegli interakciju float-a sa nekim drugim elementom stavljamo drugom elementu property clear sa vrednostima: left, right, none, both, inherit.

Q: Pseudo klasa u CSS-u

A: Pseudo klase služe za definisanje specijalnog stanja elementa, sintaksa je primera :hover. Stanje može da bude active, hover, visited, first-child..

Q: Pseudo atribut u CSS-u

A:

Javascript

Q: Koraci izvršavanja JS koda?

A: Interpretor ide od linije do linije koda i alokira memoriju u memory heap-u. Pre nego se interpretor pokrene desi se hoisting i JIT(Just in Time) kompajliranje da bi

se kod preveo u masinskom kodu. Kada interpretor naide na metodu on je stavlja u Call Stack-u gde se metode zovu po principu FILO stack-a (First In Last Out).

Q: Sta je hoisting?

A: Hoisting je automatsko pomeranje deklaracija varijabli na pocetku samog koda.

Q: Kako deklarismo globalnu promenljivu, kako deklarismo promenljivu za global scope i kako deklarismo promenljivu koja je "ziva" samo u trenutnom bloku?

A:

Globalnu promenljivu deklarismo bez ikakvih parametra, npr: promenljiva

Promenljivu za global scope deklarismo sa "var", npr: var promenljiva

Promenljivu u trenutnom bloku deklarismo sa "let", npr: let promenljiva

Q: Razlika izmedju ES5 i ES6?

A:

1. Support for constants or immutable variables
2. Block-scope support for variables, constants and functions
3. Arrow functions
4. Default parameters
5. Rest and Spread Parameters
6. Template Literals
7. Multi-line Strings
8. Destructuring Assignment
9. Enhanced Object Literals
10. Promises
11. Classes
12. Module

Q: Kako kreirati objekat u JS-u? ES5? ES6?

A:

Vise nacina, jedan od nacina za ES5:

```
Let object = {};
```

Nacin za ES6:

```
Class test(){}  
  
Test object = new test();
```

Q: Keyword this u JS-u?

A: Vraca referencu objekta, ako koristimo lambda izrate ne mozemo koristiti this.

Kako da zaustavimo osvezavanje stranice putem js-a?

A: Void(0)

Q: Push metoda u JS-u?

A: Dodaje element na kraju array-a.

Q: Spice i Array Splice razlika?

A: Spice uzima elemente iz array-a i smesta ih u novi array dokk Splice brise elemente iz array-a i prikazuje nam koji su elementi izbrisani/izbaceni iz array-a.

Q: Promise?

A: Promise je proxy za vrednost ne moramo da znamo na pocetku o kojoj se vrednosti radi.

Promise ima 3 statusa: pending, fulfilled, rejected.

Pending je da je u toku izvršavanje, fulfilled je da je uspesno izvršen i da se pozove callback za uspesno izvršenje dok rejected predstavlja odbijen zahtev i tada se poziva callback za rejected.

Q: Closure?

A: Kada jedna funkcija sadrži drugu funkciju i sadržana funkcija može da uzme property funkcije u kojoj je ugnježena.

Q: Razlika izmedju == i === operatora?

A: == ispituje samo za isti tip podataka dok === ispituje da li je isti tip i da li je ista vrednost.

Q: Sta je IIFE?

A: IIFE je funkcija koja se pokrece odmah kada je definisemo, primer:

```
(function () { // logic here } ) ();
```

Q: Sta je callback funkcija?

A: callback funkcija je funkcija u drugoj funkciji koja se poziva da obavi odradi odredjeni task.

Q: Sta je callback hell?

A: callback hell su nested callback funkcije cija je posledica tesko citanje koda i tesko debug-ovanje. Primer callback hell-a:

```
async1(function(){  
  async2(function(){  
    async3(function(){  
      async4(function(){  
        ....  
      });  
    });  
  });  
});
```

Q: Sta je strict mode?

A: Strict mode se deklarise kad zelimo da nam izbacuje sve errore umesto errore samo koji bi mogli da nam break code.

Q: Razlika izmedju Windows i Document?

A: Windows je root element bilo koje stranice dok je document samo deo windows.

Q: Kako pristupamo history u JS-u?

A: windows.history.(operacija)

Q: isNaN() u js-u?

A: ugradjena funkcija koja ispituje da li je dati value broj.

Q: isFinite() u js-u?

A: Ispituje da li je broj konacan ili beskonacan, konacan vraca true dok beskonacan false.

Q: Event Bubbling?

A:

Q: Kako submitujemo formu?

A: document.forms[0].submit();

Q: Kako naci klijentske podatke?

A: Koriscenjem navigator

Q: polyfil?

A: omogucava pokretanje js -a na starim browserima.

Q: freeze?

A: freezuje objekte ili array

Q: <noscript>

A: Onemogucuje js kod u dokumentu.

Q: Primitivni tipovi u JS-u?

A:

1. string

2. number

3. boolean

4. null

5. undefined

6. bigint

7. symbol

Q: concat?

A: Sluzi za kombinovanje 2 array-a.

JQuery pitanja

Q: Sta je JQuery?

A: JQuery je mala javascript biblioteka nastala 2006 sa sloganom "Write less, do more". JQuery kod je baziran na js kodu gde je oznaka \$ oznaka za JQuery. JQuery je lightweight(19 KB), ima Ajax support, Event Handlers i Animaciju.

Q: Hide() funkcija u JQuery-u?

A: Sakriva selektovani element

Q: Sta su selektori i koji tipovi selektora postoje?

A:

- Univerzalni selektor `$(*)` obuhvata ceo DOM
- Selektor po imenu elementa, primer: `$(p)`
- Klasni selektor `$(.imeklase)` obuhvata sve elemente sa atributom klase
- ID selektor `$(#imeid)` obuhvata sve lemente sa atributom id
- Pseudo klasa selektor `$(ime:hover)`

Q: Funkcija za loop u JQuery-u?

A: JQuery.each()

Q: Kako izbeci konflikt simbola \$?

A: U slucaju da u js-u se koristi jos neki library sa simbolom \$ tada iznad JQuery koda pisemo narednu JQuery.noConflict(). Takodje mozemo da zamenimo simbol \$ drugim simbolom na ovaj nacin primer var x = JQuery.noConflict();

Q: Neke funkcije za efekte u JQuery?

A:

- FadeIn()
- Fadeout()
- Toggle() //izmedju hide() and show()
- Animate()
- Show()
- Hide()
- Itd..

Q: Kako da odlozimo izvršenje funkcija u JQuery-u?

A: delay(speed, queueName) odlaze execution u queue

Q: Korist html() metoda?

A: html() metod menja ceo content elementa novim

Q: korist css() metoda?

A: Da se uzme ili izmeni css element.

Q: Razlika izmedju find i child metode?

A: find pretrazuje ceo DOM dok child pretrazuje samo jedan level DOM-a.

//Document Object Model – DOM

Q: Razlika izmedju prom() i attr() ?

A: **attr()**: It gets the value of an attribute for the first element in the set of matched element.

prop(): it gets the value of a property for the first element in the set of matched elements. It is introduced in jQuery 1.6.

Q: Sta korsitimo da bi sacekali da se DOM ucita pa onda tek da implementiramo JQuery?

A: `$(document).ready()`

Q: AJAX funkcije u JQuery?

A:

- `.ajax()` – salje asinhroni zahtev
- `.ajaxSetup()` – postavljamo opcije za ajax pozive
- `.getJSON()` – prihvata url do gde je request poslat

Q: Objasni `.empty()`, `.remove()` i `.detach()` u JQuery?

A:

- `.empty()` brise sve child elemente
- `.remove()` brise sve elemente
- `.detach()` kao remove samo sto ostavlja JQuery podatke

Q: Kako promeniti width preko jquery

Q: `bind()` vs `live()` vs `delegate()`

Q: Kako kreirati cookie sa jquery?

A: `.dough` koriscnjem

Q: `$(window).load`?

A: Kada se sve ucita, ide posle `$(document).ready`

Q: `.val()`?

A: Vraca vrednost prvog elementa u setu elementa koji se match

Q: Koja 4 parametra se koriste za AJAX metode?

- URL – specificiramo URL na kome se salje zahtev(request)

- TYPE – tip http request (POST, GET, HEAD, PATCH, OPTION, DELETE, PUT)
- DATA – koje podatke saljemo ka serveru
- CACHE – da li browser treba da kesira podatke

Q: JQuery chaining?

A: Omogucava da se nekoliko metoda execute u jednom JQuery selektovanju.

HTTP pitanja

Q: Sta je HTTP?

A: HTTP(Hyper Text Transfer Protocol) služi za komunikaciju između klijenta i servera, app layer. Komunikacija između klijenta i servera se odvija tako što se šalje HTTP zahtev(Request) i prima HTTP odgovor(response).

Q: Kako se izvršava HTTP komunikacija?

A:

- Klijent šalje request serveru
- Web server prihvata request
- Web server pusti odgovarajuću aplikaciju da završi request
- Web server šalje http response
- Klijent prihvata http response

Q: HTTP metode?

A:

- GET
- POST
- DELETE
- OPTION
- PUT
- PATCH
- HEAD

Q: GET metoda?

A: GET metoda služi za traženje podataka od servera. GET metoda:

- Nije cached
- Pojavljuje se u URL-u
- Ima restrikciju veličine URL-a
- Nije bezbedna za osetljive podatke
- Služi samo za request

Q: POST metoda?

A: POST metoda služi za kreiranje/update resursa. Podaci se šalju u body HTTP-a.

- POST nikad nije cached
- Nema restrikcije
- Ne može da bude bookmark

Q: PUT metoda?

A: Isto kao i POST metoda služi za kreiranje/update resursa.

Q: PUT vs. POST?

A: POST metodom kad refresh stranicu opet update resurs i može doći do promene koda dok sa PUT metodom ovo nije slučaj.

Q: DELETE metoda?

A: DELETE metoda briše resurs.

Q: OPTIONS metoda?

A: Vraća opcije za komunikaciju između klijenta i servera.

Q: HEAD metoda?

A: HEAD je identičan kao get samo bez response body. Ako na primer request bude tipa lista korisnika, get će da vrati listu dok head ne može da vrati listu.

Q: Error poruke HTTP?

A: Kada klijent posalje request serveru moze doci do greske i one se razlikuju po pocetnom broju:

- 1xx salje informaciju
- 2xx salje potvrdu uspesne komunikacije
- 3xx Preusmeravanje ka drugom serveru
- 4xx Doslo je do greske sa klijentske strane
- 5xx Doslo je do greske sa serverske strane

Osnovna pitanja

Q: Sta su kolacici?

A: Kolacici su lokalni sadrzaj koji cuvaju informacije o korisniku. Kada korisnik sledeci put pristupi sajtu bice upamceni neki podaci(kao npr. username korisnika). Kolacici su vidljivi i korisniku i serveru. Kolacice brisemo tako sto im promenimo Expire Date.

Q: Local storage?

Podaci se cuvaju na racunaru i nakon gasenja web browsera.

Q: Session storage?

Podaci se cujavu samo za vreme aktivnosti web browsera.

Q: Sta je Ajax?

A: Ajax služi za asinhrono slanje zahteva.

Q: Razlika izmedju statickog i dinamickog programiranja?

Q: Sta je JSON?

JSON (JavaScript Object Notation) is a lightweight format that is used for data interchanging. It is based on a subset of JavaScript language in the way objects are built in JavaScript.

Q: JSON sintaksa?

A: Below are the list of syntax rules of JSON

1. The data is in name/value pairs
2. The data is separated by commas
3. Curly braces hold objects
4. Square brackets hold arrays

Q: Sta je enum?

A: Enum je tip restikcije promenljivih na jednu promenljivu. Javascript nema metode enum.

OOP Java

Q: Apstraktna klasa?

A: Apstraktna klasa je klasa koja uz sebe ima keyword abstract. Apstraktna klasa služi kao model kreiranja klasa koje je nasledjuju takodje se naziva i Parent ili Master klasa. Ona ne može instancirati objekat i u slučaju da je bar jedna metoda apstraktna onda i klasa mora biti takodje definisana kao apstraktna. Child klasa je klasa koja nasledjuje atribut i stanje Parent klase koje može preko polimorfizma da override (statički u runtime-u polimorfizam) za nasledjivanje Parent klase koristi se keyword extends u Child klasi. Sto se tiče Jave jedna klasa može naslediti samo jednu apstraktnu klasu zato što Java ne podržava multiple (više) nasledjivanje. Za apstraktnu klasu možemo da kažemo da je nalik blueprint - u.

Note: Da ne bi dozvolili nasledjivanje klase možemo staviti final.

Note: Apstraktna klasa je deo nasledjivanja tj. IS-a relationship.

Note: Svaka klasa po default-u nasledjuju root klasu Object.

Note: Apstraktna klasa ne mora da implementira sve metode interfejsa.

Q: Interfejs?

Interfejs je kolekcija apstraktnih metoda i mehanizam u Javi za multiple nasledjivanje, kako u Javi nije podržano multiple nasledjivanje uvode se interfejsi. Interfejs se sastoji samo od apstraktnih metoda, static final atributa i postoji mogućnost inner klase. Jedna klasa može da implementuje više interfejsa. Kada

klasa implementira interface ona mora da override sve njegove apstraktne metode. Mozemo da kazemo za interfejs da je template po kom se izradjuju ostale klase.

Q: Razlika izmedju apstraktne klase i interfejsa?

A: Kao sto sam pomenuo interfejs je mehanizam da se postigne vise nasledjivanje u Javi. Razlika je u dosta stavki, neke bitne razlike su:

- Klasa moze da implementira vise interfejsa dok moze samo jednu apstraktnu klasu da extend.
- Apstraktna klasa moze imati definisane metode dok interface mora imati samo public abstract metode.
- Apstraktna klasa moze imati svoje attribute dok interface moze imati samo static final attribute.

Q: Zasto Java ne dozvoljava Multiple nasledjivanje?

A: Multiple nasledjivanje predstavlja veliki problem sa kompajler strane, javlja se Diamond problem i zato je uveden mehanizam Interface-a koji na neki nacin dopunjava prazninu Multiple nasledjivanja.

Note: mozda je i zbog hijerarhije stabla, idk.

Q: Enkapsulacija u Javi

A: Enkapsulacija ili Data Hiding je nacin sakrivanja podataka od korisnika da ne bi doslo do zloupotrebe. Enkapsulacija se moze postici na tri nacina:

- private , metodi i atributi su vidljivi samo u unatrosnjosti date klase
- public , metodi i atributi su vidljivi van klase
- protected, metodi i a atributi su vidljivi u unutrasnjosti klase kao i klasa koje nasledjuju datu klasu

Dodatno, ako stavimo neke attribute u private i trebaju nam za manipulaciju mozemo generisati public metode get i set za uzimanje ili modifikovanje datog atributa.

Q: Polimorfizam

A: Polimorfizam je mogucnost da objekat uzme vise formi. Svaki objekat koji je u IS-a relaciji je polimorfik. Postoje dva polimorfizma:

- overload koji se izvrsava u kompajliranju, dozvoljava da imamo dve istoimene metode sa razlicitim parametrima.
- override koji se izvrsava u runtime-u dozvoljava adaptiranje/menjanje metode u child klasi.

Q: Objekat u Javi?

A: Objekat u javi je interpretacija realnog objekta kao sto je i kljuc u OOP-u. Objekat mozemo da poredimo sa objektima u realnom svetu tipa lampa koja ima svoje attribute kao visina,tezina,boja... koje opsiuje njeno stanje i metode ukljuci,pomeri.... Koje predstavljaju njeno ponasanje tako da se objekat sastoji od stanja i ponasanja(state and behavior). Objekat u Javi je instanca klase(ne znaci za sve programske jezike kao na primer js (all my homies hate js)). Objekat alocira memoriju i napravi se keyword-om new.

Note: Default vrednost za objekte je null

Q: Klasa u Javi?

A: Klasa je blueprint za kreiranje objekta i sadrzi stanje i ponasanje objekta. Klasa ne zauzima memoriju i sluzi samo za instanciranje objekta. Metode predstavljaju ponasanje dok atributi stanje objekta.

Q: Konstruktor?

A: Konstruktor je metoda u klasi koja sluzi da dodeli attribute ili ulazne parametre objektu. Ima isti naziv kao i klasa u kojoj se inicijalizuje i nema return tip. Postoje vise tipa konstruktora:

- Default konstrutor koji se automatski instancira ako ga mi ne napisemo
- Copy konstruktor za deep copy objekta
- Paramterizovan konstruktor, zadajemo parametre objekta

Q: This keyword?

A: this keyword je referenca na trenutni objekat. Nothing more.

Q: Exceptioni u Javi?

A: Exceptioni/izuzeci je dogadjaj koji se desi kada se nas program ne moze normalno izvrsiti. Exception je takodje klasa koju nasledjujemo iz java.util paketa. Postoje check i unchecked exceptioni. Unchecked exceptone ne handle-ujemo to su

error i RuntimeException, primer exceptiona koji ne moramo da handle je / by zero (eg. `Int a = 2/0`). Check exceptione moramo da specificiramo na metodi koja baca zadati Exception keywordom throws(mozemo da specificiramo vise Exception-a i odvajamo ih zapetom) uslov na kome se desava Exception definisemo keyword-om throw <Ime Exceptiona> ove exceptione handluje u try/catch/finally bloku.

Q: Try/catch/finally blok u Javi?

A: U Try bloku se izvrsava kod dok ne dodje do nekog izuzetka kada naidje na izuzetak onda ga hvata u catch bloku i handluje ga da ne bi stavljali n catch bloka za n exceptiona mozemo samo napisati catch(Exception e) gde mozemo vratiti poruku ili koristan stacktrace u finally bloku se kod izvrsava nezavisno od izuzetka.

Q: Tredovi/Niti u Javi?

A: Tredovi su deo klase Thread iz java.util. paketa. *(Zvacu ih Thread)*. Threadom postizemo paralilazaciju u Javi tako sto napravimo Workere koji rade na jednom delu koda. Primer koriscenja threada je da niz podelimo na t jednakih delova i nadjemo najmanji element, ovaj proces nam daje na vremenu pa se i vremenska slozenost smanjuje t puta. Broj threadova/workera koje mozemo napraviti nam diktira procesor u racunaru, ako imamo primer 4 jezgra moguće je napraviti max 4 workera, u slucaju da napravimo 5 workera a imamo 4 jezgra u procesoru mi necemo dobiti na brzini zato sto ce na jednom jezgru raditi vise threadova/workera. Thead moze implementirati interfejse callable i runnable, u oba slucaja override-ujemo run funkciju. U slucaju runnable run nema povratne parametre i ne moze bacati exception dok u callable ovo mozemo. Thread se pokrece metodom <ime objekta/threada>.start() , tada se alokira potrebna memorija za thread i on kreće u izvrsavanje.

Q: Deadlock problem?

A: Deadlock problem je problem sa Threadovima. U slucaju da u istom trenutku dva threada uzmu isti kod onda jedan drugog cekaju da se taj kod oslobodi i tu nastaje beskonacno cekanje pa nam se kod blokira. Kako resavamo ovaj problem? Problem moze da se resi na vise nacina primer preko synchronized ili reentrant lock-a. Sustina je da simuliramo neki nacin rampe gde jedan thread ulazi u kod i u

tom trenutku zaključava ulazak drugog threada gde drugi thread konstantno ispituje da li je prvi thread završio svoj posao i ako jeste onda ulazi i izvršava kod a zaključava ulazak narednog threada.

Q: Asocijacija u Javi?

A: Asocijacija je has-a relationship između objekata. Asocijacija se deli na dva tipa:

- Kompozicija
- Agregacija

Q: Agregacija u Javi?

A: Agregacija je tip asocijacije. Objekat sadrži drugi objekat. Ali nisu zavisni. Student ima adresu (Student i adresa su objekti) ako se izbrise adresa student idalje može postojati samo bez adrese.

Q: Kompozicija u Javi?

A: Kompozicija je tip asocijacije gde su objekti međusobno zavisni, jedan objekat sadrži druge objekte u sebi. Ovo je dizajn tehnika. Ako se izbrise kontejner objekat tada se i ostali objekti u njemu brisu. Primer: U biblioteci imamo knjige, ako se uništi biblioteka tada se i knjige uništavaju.

Q: Static keyword?

A: Static-om kažemo da je

Q: Final keyword?

Final, nema promena. Dati atribut/metodu/klasu ne možemo da menjamo. Restrict keyword.

Q: Null?

Je rezervisan za literal values, slično true i false, in a nutshell nema vrednosti

Note: Default value za objekte je null.

Q: Shallow Copy i Deep copy?

A: Shallow copy je plitko kopiranje objekta, obično ako dodeljujemo iste parametre kao vrednosti jednog objekta na drugi ili korišćenjem copy() objekte

funkcije u javi. Deep copy daje identican oblik objekta, postoje vise nacina a najpopularniji je Copy constructor.

Q: Paterni u javi?

A: Paterni u javi su resenja nekih standardnih problema. Paterni se ne moraju koristiti uvek, narocito kada kod mozemo napisati u prostijem obliku. Postoje structural, functional i behavior dizajn paterni.

Q: Singleton?

A: Singleton je dizajn tehnika koju koristimo ako zelimo da se osiguramo da objekat ima samo jednu instancu, stavimo da je privatan konstruktor i obezbedimo metodu getInstance() koja ce da nam vrati instancu objekta.

Q: Lanac odgovornosti?

A: Niz handlera koji su medjusobno povezani. Ako jedan handler ne moze da obavi zadatu instrukciju on prosledjuje drugom koji ispituje da li moze on obaviti pa u slucaju da ne onda prosledjuje trecem itd.

Q: Garbage collector?

A:

Q: Socketi u Javi?

A: Socket je klasa iz java.net paketa. Omogucava two-way konekciju izmedju klijenta i servera. Server ceka da se klijent poveze uz pomoc host name-a i port number-a. Ako je unet ispravan host name i port number server i klijent su povezani uz pomoc socketa.

Q: Iteriranje u Javi?

A: Prolazak kroz listu, mapu, vektor.. vrsimo pomocu Iteratora. Iterator je klasa u javi koja ima ugradjene metode

- hasNext() vraca true ili false ako postoji elementi sledeci u datom nizu
- next() sledeci element u nizu

Neki pomocni sajтови za učenje:

<https://nedschooldevelopers.com/CSS/CSS.php> SRB

<https://www.znanje.org/knjige/computer/java/ib01/300Java/31000110.htm> SRB

//v0.1 Pancic Aleksa