

1. Koja je razlika između overridinga i overloadinga?

Polimorfizam – svojstvo objektno usmjerenog programa da se jedna apstraktna operacija može izvesti na različite načine u različitim razredima. (npr razredi GeometrijskogLika su Krug i Kvadrat a oni imaju različite metode za površine)
-dakle više metoda istog naziva i istog broja parametra npr.

-dakle overloading je tip polimorfizma

Overloading – višestrukost metoda - postojanje više metoda istog naziva, ali različitog broja, tipova i mjesta parametra

Overriding – Nadjačavanje – vezano uz hijerarhiju i nasljeđivanje razreda

-metoda definirana u nadrazredu, nasljeđuje se u podrazred, ali se redefinira zato jer podrazred sadrži inačicu te metode. korištenje: (restrikcija, proširenje (ekstenzija), optimizacija)

2. Kakva nemože biti veza u refleksivnoj relaciji?

Može biti bilo kakva osim nasljeđivanja te realizacije (<<interface>>)

3. Koje dijagrame ste crtali?

Crtali smo dijagrame **obrasca uporabe**, **sekvencijske** dijagrame, dijagram **razreda**, dijagram **objekata**.

4. Koji su statički, koji dinamički?

statički su (strukturni) : dijagram obrasca uporabe (OBOJE!), dijagram razreda, dijagram objekata, dijagram komponenti, dijagram razmještaja, dijagram paketa

dinamički su (ponašajni): dijagram obrasca uporabe (OBOJE!), sekvencijski dijagram, dijagram aktivnosti, dijagram stanja, komunikacijski dijagram.

Dijagrami obrazaca uporabe su statički UML-dijagrami (kao i dijagrami razreda, objekata, paketa i razmještaja), a također pripadaju i skupini ponašajnih dijagrama budući da modeliraju moguće ponašanje korisnika sustava.

5. Sto je dijagram uporabe obrasca?

Dijagram obrasca uporabe je dijagram kojemu je cilj/svrha prikazati ponašanje sustava, dijelova sustava ili konkretnog razreda na način vidljiv korisniku sustava.

6. Sto se sve nalazi u njemu, kakve veze postoje?

U njemu se prikazani **aktori**, **obrasci uporabe te veze i odnosi** među aktorima i obrascima uporabe. *Dodatno se može istaknuti granica sustava koji se ograđuje. Vrste veza: **asocijacija**, **generalizacija**, **uključivanje (<<include>>)**, **proširenje (<<extend>>)**

7. Razlika između include, extend (definicija, razlika u crtanju, kakvog je smjera,...)

Include - Vezom uključivanja se povezuju dva obrasca uporabe na način da jedan obrazac u tijeku svog izvođenja u potpunosti izvede uključeni obrazac uporabe. Smjer: od osnovnog obrasca uporabe prema onom koji se uključuje.

Extend - Vezom proširenja se povezuju dva obrasca uporabe pri čemu jedan proširuje funkcionalnost drugog. Proširenje se ostvaruje ako je zadovoljen određeni uvjet definiran u točki proširenja. Smjer: od onog koji je proširen do onog osnovnog obrasca uporabe.

8. Sto je sekvencijski dijagram?

Sekvencijski dijagrami (engl. sequence diagram) i komunikacijski dijagrami (engl. communication diagram) pripadaju široj skupini UML-dijagrama međudjelovanja

(engl. interaction diagram). Obje vrste dijagrama spadaju u nadskupinu ponašajnih dijagrama.

Sekvencijski i komunikacijski dijagrami daju naglasak na vremenskom redoslijedu kojim se odvija međudjelovanje sudionika u sustavu, tako da ih se svrstava i u dinamičke UML-dijagrame.

Sudionici koji se modeliraju na sekvencijskim i komunikacijskim dijagramima mogu biti aktori (predočavanjem komunikacije aktora s dijagrama obrazaca uporabe) ili objekti, pri čemu se prikazuje komunikacija instanci razreda prikazanih na dijagramu razreda. Dok se prilikom komunikacije aktora govori o porukama, kod objekata to su pozivi postupaka u objektima drugih razreda.

9. Sto se sve nalazi u njemu, kakve poruke postoje?

Nalaze se aktori/objekti, te pripadajuće poruke koje razmjenjuju aktori/objekti.

Postoje sinkrone te asinkrone poruke. Kod sinkronih se čeka odgovor, dok se kod asinkronih nastavlja s radom bez čekanja potvrde/odgovora.

10. Sto je dijagram razreda?

Dijagrami razreda (engl. class diagrams) opisuju razrede i njihove međusobne veze.

Jednako tako, dijagrami razreda opisuju vrste objekata unutar nekog sustava i njihove međusobne statične odnose.

Dva osnovna tipa odnosa između razreda su pridruživanje (veza ili asocijacija) i podtip

11. Sto se sve nalazi u njemu, kakve veze postoje?

Dva osnovna tipa odnosa između razreda su pridruživanje (**veza ili asocijacija**) i podtip (nasljeđivanje).

Veze mogu biti jednostruke, višestruke ili refleksivne. Također mogu biti i **agregacija i kompozicija**. Odnos podtip određen je mehanizmom nasljeđivanja u međusobno suprotnim smjerovima generalizacije i specijalizacije. Osim navedenog, UML-dijagrami razreda mogu prikazati atribut i operacije razreda, njihova svojstva i ograničenja nad njima, višestrukost povezivanja, pakete, ovisnost, tipove podataka, obrojčavanje (enumeraciju), komentare i više drugih strukturnih svojstava.

12. Sto je apstraktna klasa? Navedi primjer.

Apstraktna klasa jest razred koji ima operacija bez implementacije.

13. Sto je sučelje?

Sučelje-interface- je skup operacija koje specificiraju usluge nekog razreda. (npr. start_engine() u razredu auto, kamion, apn..), Slično kao razred samo što nedostaju varijable instanci te implementacija metoda.

Podržava uključiti i koristi koncept. (plug and play).

Realizacija je naziv veze kojim je povezan razred s sučeljem.

14. Sto su funkcionalni zahtjevi?

-zahtjevi koji određuju koje usluge bi sustav morao pružati

-koje je očekivano ponašanje sustava u određenim situacijama

-kakve odgovore daje na poticaje iz okoline u domeni primjene

15. Opiši vas projektni zadatak?

- pdf

16. Koji su dijagrami koje ste napravili statički a koji dinamički ?

Statički: obrazaca uporabe, objekata i razreda

Dinamički: sekvencijski

17. Sta je najvažnije u dijagramu razreda ?

On opisuje objekte unutar nekog sustava i njihove međusobne odnose; daje apstraktan pogled na sustav.

Sastoji se od: razreda, pridruživanja, atributa, operacije i generalizacije.

18. Sto definira razred (u dijagramu razreda) ?

To je osnovni tvorbeni element UML dijagrama razreda, **apstraktna reprezentacija objekta** određenog razreda; predstavljen pravokutnikom s imenom

Graficki simbol može prikazati attribute i operacije, vidljivost i dr.

19. Sto sve definira objekt (u dijagramu objekata) ?

Ime, tip i model/funkcija.

20. Kako se zove vrh strelice u nekoj vezi?

Uloga / rola

21. Kakve sve veze postoje u dijagramu razreda ?

Agregacija, Kompozicija, Asocijacija, Nasljeđivanje

22. Ako imate neki privati atribut, a treba vam, kako dolazimo do njega ?

sa set i get

```
class IWasDesignedPoorly {
    private Hashtable stuffIWant;
}

IWasDesignedPoorly obj = ...;

Field f = obj.getClass().getDeclaredField("stuffIWant");
//NoSuchFieldException
f.setAccessible(true);
Hashtable iWantThis = (Hashtable) f.get(obj); //IllegalAccessException
```

23. Koja je razlika između dijagrama objekata i dijagrama razreda ?

Dijagram objekata prikazuje instance u jednom trenutku izvođenja pa stoga nema brojnosti, dok dijagram razreda daje općenit pogled na odnose između instanci pa ima i brojnosti. Iz jednog dijagrama razreda moguće je napraviti beskonačno dijagrama objekata.

24. Koja je posebnost sekvencijskog dijagrama ?

Služi za prikaz slijeda događaja među objektima pojedinih razreda kao i među aktorima.

Orijentiran je na vremenski prikaz. Ponašanje se modelira dinamičkim interakcijama između objekata pojedinih razreda i između aktora preuzetih iz dijagrama obrazaca uporabe.

25. Kako izgleda sekvencijski dijagram?

Tijek interakcije je od gore lijevo prema dolje desno. U pravokutnicima su aktori. Razmjenjuju se sinkrone/asinkrone poruke koje mogu sadržavati uvjet ili prenositi parametre.

26. Koje sve vrste poruka postoje u sekvencijskom dijagramu ?
Sinkrone i Asinkrone poruke, stvaranje i uništavanje objekata, poruke samom sebi.
Sinkrone - objekt koji ju šalje čeka odgovor i tek kada ga dobije nastavlja sa radom
Asinkrone - nastavlja sa radom i ne čeka odgovor
27. Može li jedan aktor u sekvencijskom dijagramu obustaviti rad (ubiti) drugog ?
Može, npr. otkaz radniku
28. Koje sve veze postoje u dijagramu obrazaca upotrebe ?
Pridruživanje ili asocijacija -----
Generalizacija -----|>
Include ----->
Extend < -----
29. Koja je razlika između include(uključivanja) i extend(proširenja) ?
Include označava da se neka radnja može izvoditi istovremeno s drugom radnjom, dok
extend označava mogućnost ostvarivanja neke radnje dodatno.
30. Koja je razlika između objekata i razreda ?
Objekt - predstavlja entitet iz stvarnog svijeta ili neki koncept; apstrakcija nečeg što ima dobro definirane granice i smisao u sustavu
Razred - opis grupe sa sličnim svojstvima, svaki objekt je pojedinac(instanca) jedne klase !!
31. Sto je apstraktna klasa ?
To je klasa koja ne može imati objekte, iz tog razloga ne može imati implementirane sve metode i treba imati barem jednu apstraktnu metodu (čisto virtualnu) i osnovna svrha joj je da bude "roditelj" drugim, konkretnim klasama.
32. Sto je enumeracija ?
To je oblik tipa podatka koji sadržava uređene parove imenovanih identifikatora i njima pridruženih vrijednosti.
33. Sto je realizacija ?
To je veza UML-a koja označava ostvarenje sučelja.
Veza realizacije je usmjerena od razreda prema sučelju.
34. Ako B ovisi o A, kako bismo to nacrtali u dijagramu razreda ?
Ovisnost pokazuje da jedan razred ili paket dijagrama ovisi o drugome.
Uvijek je jednosmjerna, A se naziva isporučitelj, a B klijent. {oznaka A<-----B } - citamo B ovisi o A
35. Agregacija i kompozicija, sta je to, razlike ?
Agregacija - vrsta pridruživanja koja pokazuje da jedan razred sadrži druge, tj. da je dio drugog razreda { oznaka npr.. 1<>-----0...* } (nezacrnjen dijamant) (dio od)
Kompozicija - vrsta pridruživanja slična agregaciji ali kod uništavanja objekta(tj. pojedinca) uništavaju se i pojedinci razreda koji su dio tog objekta {oznaka npr. 1 <>-----0...* } (zacrnjen dijamant) (dio od)
36. Kako biste programski ostvarili vezu asocijacije ?

```

public class SwipeCard
{
    public void Swipe( Manager obj)
    {
    }
}

public class Manager
{
    public void Logo (SwipeCard obj)
    {
    }
}

```

// ovo ti je agregacija jer ti mozes praviti objekte tipa class posebno da ne ovise o klasi Druga

37. Kako biste programski ostvarili nasljedjivanje ?

```

class A {
public int a
public metoda(){
}
}

class B extends A {
....
// ova klasa ima i int a i "metodu"
metodu moze overrideat da radi nesto drugo
moze dodavati svoje metode i svoje varijable
}

```

38. Sto su nefunkcionalni zahtjevi ?

To su ograničenja u uslugama i funkcijama, kao što su vremenska ograničenja, (ne)usvojeni standardi, zahtjevi kvalitete, platformski zahtjevi, ograničenja u procesu razvoja i oblikovanja itd.

39. Kako se u dijagramu razreda oznacavaju private, protected i public atributi ?

Private: " - "

Protected: " # "

Package: " ~ "

Public: " + "

40. Sto sve odredjuje atribut u razredu ?

Sljedeća svojstva:

- **stupanj vidljivosti**

- **naziv**

- **vrsta ili tip**

- **pocetna vrijednost**

41. Što su to stereotipi?

UML-ov mehanizam proširivosti – stereotip. *Stereotip* proširuje rječnik UML-a, omogućavajući kreiranje novih vrsta građevnih blokova koji su izvedeni iz već postojećih, ali su specifični za neki problem. Npr. ako se programira u jezicima kao što su Java i C++, često postoji potreba za modeliranjem iznimaka (*eng. exceptions*). U tim jezicima, iznimke su, jednostavno, klase, iako se s njima postupa na specijalan način. Tipično, želi se samo mogućnost bacanja i hvatanja iznimaka, ništa više. Stereotipovi ovdje dolaze u pomoć dajući mogućnost pretvaranja iznimke (kao klase) u poseban građevni blok. Klasa koja se želi tako unaprijediti označi se sa određenim stereotipom, kao klasa *Overflow* na Slici 19.

Asistent je rekao da su to stvari kojima bolje semantički opisuješ neku drugu stvar, pišu se u paru "zagrada" << >>.

Tako se na primjer za naznačavanje nekog razreda apstraktnim koristi stereotip <<abstract>>.

42. Koliko apstraktna klasa mora imati apstraktnih metoda da ju zovemo apstraktnu?
bar jednu

43. Razlika između sučelja i apstraktne klase?

Sučelje nema atribute dok apstraktna klasa može imati. Sučelje sadrži samo prototipove metodi dok apstraktna klasa može imati i prototipove (metode bez implementaciji) te metode s implementacijom.

44. Čemu služe paketi? – hijerarhija koda

Razlika između static i frozen? **frozen** (vrijednost atributa ili asocijacije ne smije se promijeniti tijekom života (engl. lifetime) pripadajućeg objekta), **static** (modifikator, vrijednost atributa je konstanta, ne mijenja se i ne ovisi o životu objekta), razlika između frozena i statica je u konstruktoru. Frozen koristi konstruktor dok static ne koristi. Static kao globalna varijabla, ne mora se instancirati da bi ju netko vidio, a frozen valjda ko static ali samo unutar objekta.

45. Što su to refleksive veze i kakve mogu biti?

veza razreda sa samim sobom, pridruživanje, agregacija, kompozicija, jednosmjerna, dvosmjerna, - ne može biti REALIZACIJA niti NASLJEĐIVANJE

Slika dijagrama razreda, sta je ovo na slici ?

Slika dijagrama razreda, zasto je ovaj razred nadklasa ?

Slika dijagrama objekata, sta je ovo na slici ?

Slika sekvencijskog dijagrama, sta je ovo na slici ?

Slika ER modela baze podataka, sta je ovo na slici ?

Slika ER modela baze podataka, kako se zovu brojevi iznad relacija ? (kardinalitet)

Na svojim papirima ima slike:

Graficki prikaz ukljucivanja (include), te pita sta je to.

Na grafickom prikazu ukljucivanja, pita koja je razlika izmedju toga i grafickog prikaza prosirenja (extend).

Graficki prikaz generalizacije, te pita sta je to.

Refleksivna veze na dijagramu razreda, te pita sta je to ?

Refleksivna veze na dijagramu razreda, koja je razlika izmedju refleksivne veze i nasljedjivanja u dijagramu razreda ?

*Ako sam ista zaboravio, ponovite dijagrame obrazaca upotrebe, sekvencijske dijagrame, dijagrame razreda i dijagrame objekata, te veze koje se javljaju izmedju elemenata pojedinih dijagrama.

**Takodjer malo OO paradigme koja se nalazi na slajdovima.

***Proučiti dokumentaciju detaljno