

Grupa2- Zamger

OOAD

SOLID principi

Studenti:

Huso Hamzić

Paša Džumhur

Rijad Handžić

S princip – svaka klasa treba imati samo jednu odgovornost

- Svaka klasa je napravljena samo zbog jednog razloga, kako bi obavljala samo jedan zadatak odnosno služila jednom cilju. Imamo velik broj klasa samo kako bi se ovaj princip zadovoljio. Za sve funkcionalnosti za koje smo smatrali da su nespojive smo pravili posebno klase kako bi sebi olakšali i sve funkcionalnosti pokrili odgovarajućim klasama.

O princip - klasa treba biti otvorena za nadogradnje, ali ne i za modifikacije

- Klase imaju veliki broj metoda te ih se još može dodavati bez da se mijenja sami nukleus klase, odnosno dodavanje novih metoda neće implicirati da se čitava klasa mora mijenjati odnosno modificirati. Također većina klasa sadrži samo kolekcije objekata drugih klasa tako da izmjena u nekoj klasi neće implicirati da se neka druga klasa radi toga mora mijenjati. Napomenimo da je čitava ideja(logika) iza sistema razrađena(dani planiranja) i sve je pravljeno sa jasnom vizijom implementacije svega.

L princip – sva nasljeđivanja moraju biti ispravno implementirana

- Na dijagramu vidimo jednu apstraktnu klasu *Aktivnost* i dvije bazne klase *Student* i *NastavnoOsoblje*. Očito je da je na mjestima gdje se koristi osnovni objekat može koristiti i izvedeni objekat po potrebi, ali slijedit ćemo pravilo da vazda vratimo pokazivač na osnovni tip iz metoda npr *List<Aktivnost>* a ne *List<Zadaća>* po konvenciji.

I princip – svaki interfejs mora obavljati samo jednu vrstu akcija

- Kako nemamo niti jedan interfejs u sistemu (za sada, no ne znači da u daljem razvoju neće doći do potrebe za nekim) ovaj princip ćemo preskočiti.

D princip – pri nasljeđivanju od više klasa, bazna klasa mora biti apstraktna

- Imamo jednu apstraktnu klasu *Aktivnost* i iz nje su izvedene klase *Ispit* i *Zadaća* odnosno *Aktivnost* se neće nigdje koristiti ni instancirati (jer će se koristiti samo ili *Ispit* ili *Zadaća*) pa je to ok da bude apstraktna klasa. No međutim imamo dvije bazne klase *Student* i *NastavnoOsoblje* koje su bazne klase a nisu apstraktne klase jer će se po defaultu podrazumijevati da je npr BSC student objekat tipa *Student* i moć će se instancirati dok će se moći i instancirati objekat tipa *MasterStudent*. Analogno je i za *NastavnoOsoblje* i *Profesor* (gdje će se bilo koja osoba iz ansambla moći instancirati kao objekat tipa *NastavnoOsoblje* ali će se moći posebno odvojiti i osobe iz ansambla koji su i *Profesor*-i). Tako da smatramo da je i ovaj princip ispoštovan, odnosno da je ovo osmišljeno na ispravan način.