**UNIVERZITET U BEOGRADU**

**FAKULTET ORGANIZACIONIH NAUKA**

**ZAVRŠNI RAD**

**naslov…..**

**Mentor Student**

**Dr… Ilija Košanin 205/2015**

**Beograd, 2022. godine**

**Apstrakt**

Ovde pišeš nešto tipa zašto si izabrao tu temu fbjerfbjfrbhfrhejfbhjhfgbrehjhj fbebesbjrbhjsbegfj

Nameštala sam prorede između paragrafa lol

**Sadržaj**

Contents

**No table of contents entries found.**

**1.Uvod**

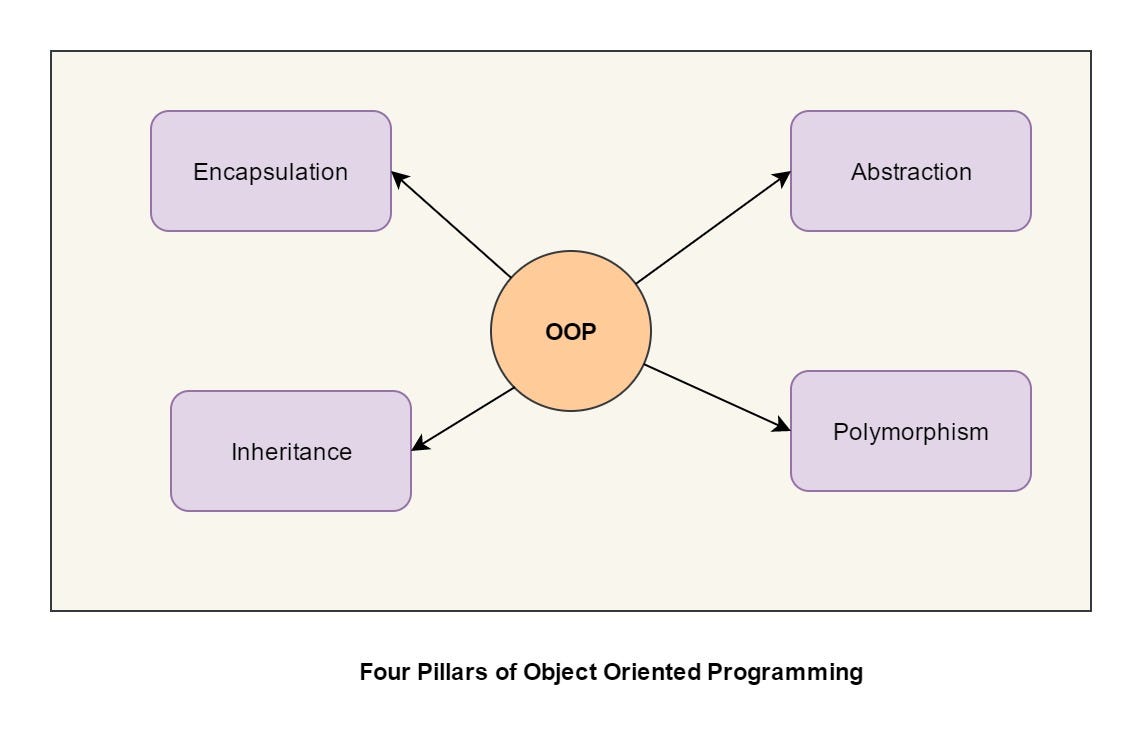
**Objektno orijentisan jezik**

Ovaj princip se zasniva na kombinaciji različitih tipova podataka, nad kojima se izvršavaju različite operacije. Ovo praktično znači da su kod i podaci spojeni u jedan koncept koji se naziva objekat.

* Prvenstveno ideja je odvajanje dela koda koji se menja od dela koji uvek ostaje isti. Ovo omogućava pouzdaniju osnovu za projektovanje softverskog sistema. Takođe, cilj je promovisanje upravljanja velikim softverskim projektima, povećanje kvaliteta i smanjenje broja neuspešnih projekata.
* Drugi važan deo je povećanje moguće ponovne upotrebe koncepta iz jednog projekta u drugom. Na primer, korisnik iz jednog softverskog sistema ne treba da se razlikuje od korisnika u drugom projektu.

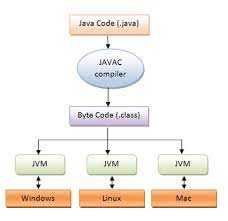
Četiri osnovna stuba objekto orijentisanog programiranja su:

* Enkapsulacija – skrivanje implementacije podataka pomoću privatnih I javnih metoda.
* Apstrakcija – implementira se korišćenjem apstraktnih klasa i interfejsa I podrazumeva ideju da znamo šta će neka klasa uradi bez da znamo kako će to da uradi.
* Nasledjivanje – Koristi se kada jedan objekat ima vezu sa drugim. Na primer, prica i patka. Patka će da bude objekat koji će da nasledi objekat ptica i koristiće osnovne funkcionalnsti iz tog objekta. Ono što je bitno jeste da sam objekat patka moze da se dopuni novim funcionalnostima koje neće da poremete objekat pritca na bilo koji nacin. Ovo praktično znaci da možemo da implementiramo veliki broj objekata koji će da koristi osnovne funkcionalnosti objekta koji nasleđuju i u isto vreme implementiraju svoje specifične funkcionalnosti.
* Polimorfizam – Podrazumeva koncent “Jedno ime više oblika”, Deli se na statični i dinamični. Usko je povezan sa nasleđivanjem i omogućava korišćenje različitih funkcionalnosti iz klase koja se nasleđuje i daje mogućnost promene tih funkcionalnsti ukoliko postoji potreba za tim.

****

Nezavisnost od platforme

Nezavisnost od platforme je koncept na osnovu kojeg je projektovan Java programski jezik. On predstavlja jednu vrstu filozofije i principa dizajniranja programskih jezika. Podrazumeva pravilo da kada se jednom napiše kod, taj isti kod može da se pokrene na bilo kojoj mašini, bez obzira na kojoj platformi je ta mašina. Kako Java programski jezik postiže ovo?



Postiže ga na dva načina. Prvi način je preko Java kompajlera koji prevodi Java kod u bajtkod Java, koji predstavlja skup pojednostavljenih mašinskih instrukcija specifičnih za Java platformu. Kada se ovaj prvi deo završi, na scenu stupa Java virtuelna mašina koja tumači i izvršava generički Java bajtkod u izvorni mašinski kod na hardveru domaćina. Takođe, uz ova dva mehanizma, dostupne su i standardizovane biblioteke koje omogućavaju unificiran pristup domaćinovim hardverskim funkcijama, kao što su grafika i niti.