

Svojstva OOP:

Apstrakcija podataka - Data Abstraction

Smanjuje se stupanj kompleksnosti uzimajući u obzir samo bitna svojstva i ponašanja (dinamiku)
Odnosi se na svaki entitet kojeg identifikirate kao bitan dio vašeg problema
Potrebno je pronaći minimalni skup svojstava i funkcionalnosti bitnih za aplikaciju koja je vezana uz zadani problem
Jedan od osnovnih mehanizama apstrakcije u većini OOP jest mehanizam sučelja (eng. Interface)
Tim mehanizmom omogućeno je pojednostavljenje kompleksnih sustava i razumnija implementacija

Učahurivanje - Encapsulation

Mehanizam koj osigurava da podaci ne budu krivo upotrijebljeni ili čak izmijenjeni na način koji mijenja smisao cijele klase ili bilo kojeg drugog dijela koda
Podaci se skrivaju na način da im korisnik ne može direktno pristupiti
Skrivaju se i informacije - maskiranje implementacijskih detalja
Direktno je u vezi s apstrakcijom

Nasljeđivanje - Inheritance

Omogućuje višestruku upotrebu istog koda (nema potrebe za ponavljanjem)
Parent Class ! Child Class
Postoji hijerarhija koja je vrlo bitna
Ekstenzivnost je povezano s nasljeđivanjem na način da se kreiraju nove klase iz postojećih i potom se dodaju nova svojstva ili ponašanja
Specijalizacija je veoma slično povezana s nasljeđivanjem na način da se mijenja neko ponašanje naslijeđeno iz polazne klase

Polimorfizam - Polymorphism

Odnosi se na sposobnost da različiti objekti odgovaraju na istu poruku na različite načine
Isti operator ili funkcija (istog imena/naziva) se može odnositi na različite akcije ovisno o podacima koji se dohvataju
Programski jezik JAVA za sada ne podržava preopterećenje (eng. overloading) operatora
Prema ovom mehanizmu objekti koji pripadaju istoj hijerarhiji klase mogu različito reagirati na istu pozvanu metodu
Drugacije kazano metode (funkcije) istog naziva mogu poprimiti različite forme
Koristimo isto ime za metodu (funkciju), a imamo na raspolaganju više različitih funkcionalnosti koje se nalaze pod tim imenom
Ispravnost korištenja baš željene funkcionalnosti vezana je uz tzv. dinamičko vezanje

Ustrajnost - Persistence

Ekstenzivnost - Extensibility

Delegiranje - Delegation

Konkurentnost objekata - Object Concurrency
Upravljanje događajima - Event handling
Prosljeđivanje poruka - Message Passing

Struktura dijagrama klasa ogleda se u određivanju svih klasa i njihovih relacija. Prema tome, elementi dijagrama klasa su cvorovi dijagrama koji predstavljaju klase i pripadajuće veze. Veze među klasama mogu biti različitog tipa, a jednu od njih smo već upoznali! Nasljeđivanje. Nasljeđivanje se ponekad u UML dijagramima klasa naziva generalizacija. Ostali tipovi veza su asocijacije (usmjerene i reflektivne), višestrukost, agregiranje, kompozicija, realizacija.

Kolekcije su karakterizirane većim brojem objekata /elemenata na koja se mogu primijeniti ista pravila

Liste moraju čuvati objekte/elemente u određenom redoslijedu (slučajno, kako su uneseni ili sortirani po određenom pravilu)

Skupovi su slični listama s tom razlikom što ne mogu da sadrže ponovljene elemente

Redovi redoslijed elemenata određuju pravilima cekanja

Mape su određene uređenim parom vrijednost - ključ. Opcenito i vrijednost i ključ mogu biti objekti bilo kojeg tipa