Method overloading = het gebruiken van dezelfde fucntienaam maar andere invoerswaarde

Oo programmeren = voordeel : hergebruiken van code

Associatie = ene object maakt gebruik van het andere

Aggregaties = ene object is eigenaar van het andere maar de objecten kunnen afzonderlijk bestaan.

Compisite = 2 objecten zijn onlosmakelijk met elkaar verbonden. Het ene object kan niet bestaan zonder het ander. Als het ene ophoudt te bestaan zal het ander ook ophouden te brestaan

High cohesion = een klasse moet gericht zijn op 1 kerntaak. Daardoor kan die klasse later makkelijk hergebruikt worden => zorgen voor hergebruik

Method overriding = word gebruikt in super en subklasses heeft dezelfde signatuur(kan gebruikt worden als er bv iets andere moet gebeuren in een subklassen dan in de superklassen.

Polymorfisme = een object die verschillende vormen kan aannamen.

Final = laatse versie hier kan niet meer een extends van gemaakt worden

Abstract = een klasse waar geen objecten van kunne gemaakt worden maar enkel voor bv functies voor andere klasse gemaakt worden.

Inctance of kijkt of het in een object past als het nie in de juiste hiërarchie is geeft het een fout melding.

Inheritance = het gebruik van code in een andere klasse

Signatuur van een methode = als je een functie overschrijft in een subklasse moet deze van hetzelfde type zijn

Late binding = als er gebruik gemaakt word van polimofisme dan kan je geen gebruik maken van een set van een sub klasse die je kiest omdat het pas geinaliseerd word als die bepaalde klasse na het uitvoeren van de klasse

Final Persoon p = new Persoon(); = slaagt op de p en niet op de waardes die je er aan geeft .

Nut van abstatcte funties in een abstrate klasse is dat zo deze geen compulatie fouten krijg als je een object aanmaakt door middel van polimorfisme

Autoboxing = automatisch van een primitief datatype naar een wrapper klasse steken( bv van een int i naar Integer getal)

Autounboxing = het resultaat van een wrapperklasse uit halen

Interfaces = verplicht een bepaald aantal dingen voor de gebruiker. Ook kunnen er meer interfaces wordne geimplimeteerd wordne tegen maar één abstracte klasse