

Апстракција

Човек користи рачунар као помоћно средство у решавању разних проблема.

Решавање проблема из људске околине се изводи креирањем модела људске околине.

У моделовању се користе разне интелектуалне операције а основно је апстракција.

Апстракција је контролисано укључивање и искључивање неких детаља о објектима.

Моделовање је успостављање аналогije између оригинала и модела.

Сваки виши (процедурални) програмски језик (FORTRAN, PASCAL, C) су апстракција машинског рачунарског језика, који је сам апстракција ниског нивоа.

Проблем у решавању проблема путем ових програмских језика је размишљање из угла структуре рачунара а не из угла проблема који се решава.

Код објектно оријентисаних програмских језика (ООП) (C++, C#, PYTHON) у моделовању се полази од реалног света чиме су апстракције много ближе човеку него рачунару.

Објекти реалног света имају статичке и динамичке особине: ауто има својства боју, број седишта али и изводи акције, стартовање, промена брзине, скретање; и ове акције чине саставни део самог објекта.

Сваки објекат у ООП има своја унутрашња стања, као и скуп метода или акција која може да изврши.

Употреба ООП чини да се решавање проблема више окреће на разматрање самог проблема а мање на програмски језик или сам рачунар.

Такође, сваки члан тима програмера може да користи све објекте који су развили други чланови тима.

Класа и објекат

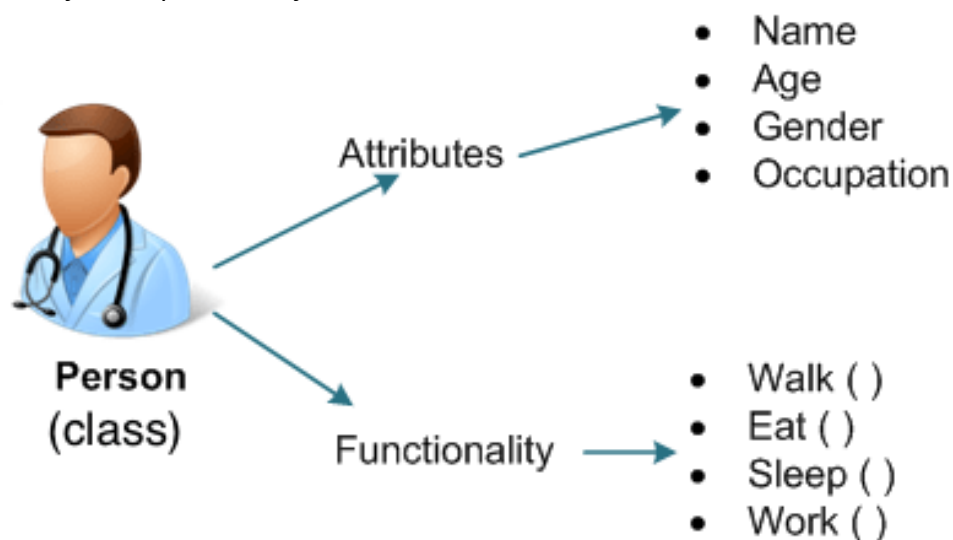
У програмирању ООП све је објекат; објекат је као променљива која чува податке али може да добије и захтеве које испуњава вршећи операције над тим подацима.

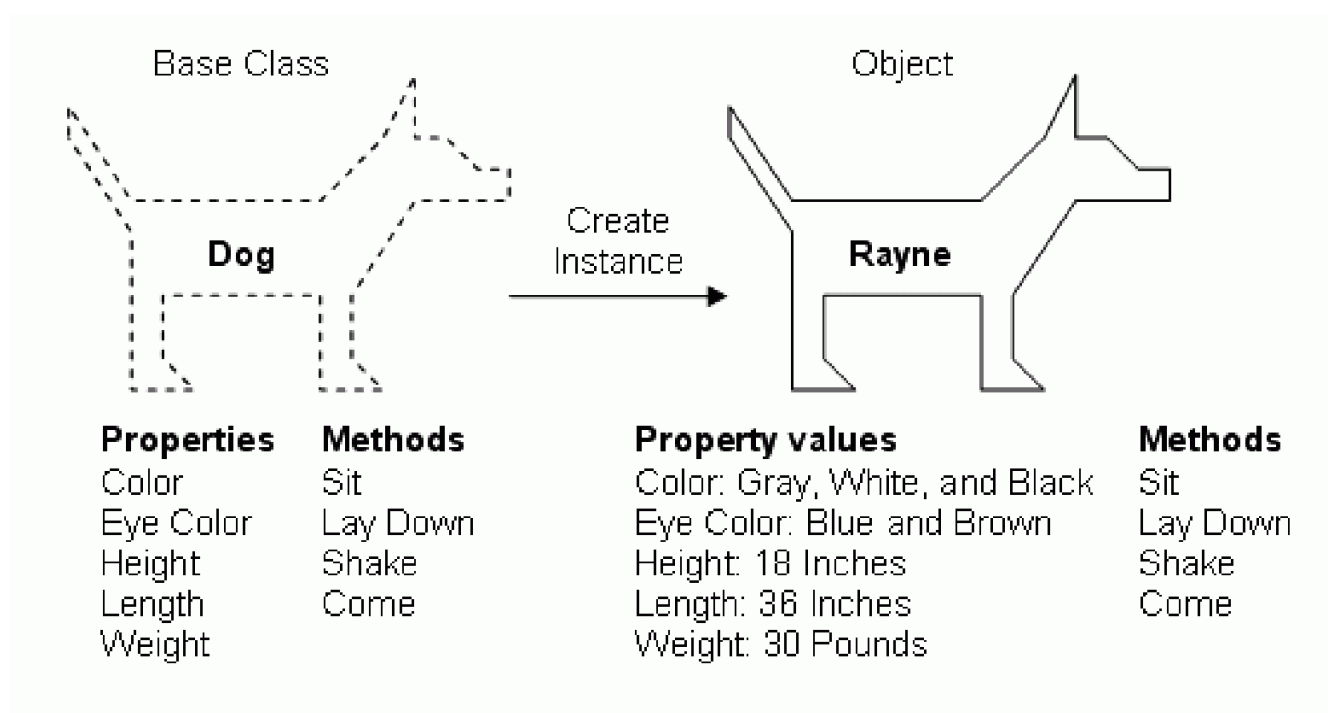
Сваки објекат има интерфејс и преко интерфејса се упућују захтеви објекту.

Програм је скуп објеката који међусобно комуницирају путем порука.

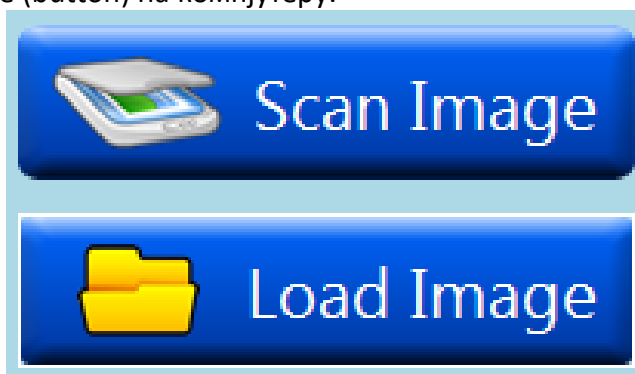
Сваки објекат је инстанца неке класе и може се схватити да је класа уопштавање типа података.

Класа је апстракција конкретних објеката.





Посматрамо класу Dugme (button) на компјутеру.



На слици су два таква објекта, кажемо да су они инстанце класе Dugme и сваки од њих може да прими исте поруке, на њих се примењују исти методи (притиснуто дугме), имају исту функционалност која је добијена из класе Dugme.

Објекти су основа ООП и представљају самосталне целине које се могу склапати са другим објектима и тако решавати сложенији програмерски проблеми.

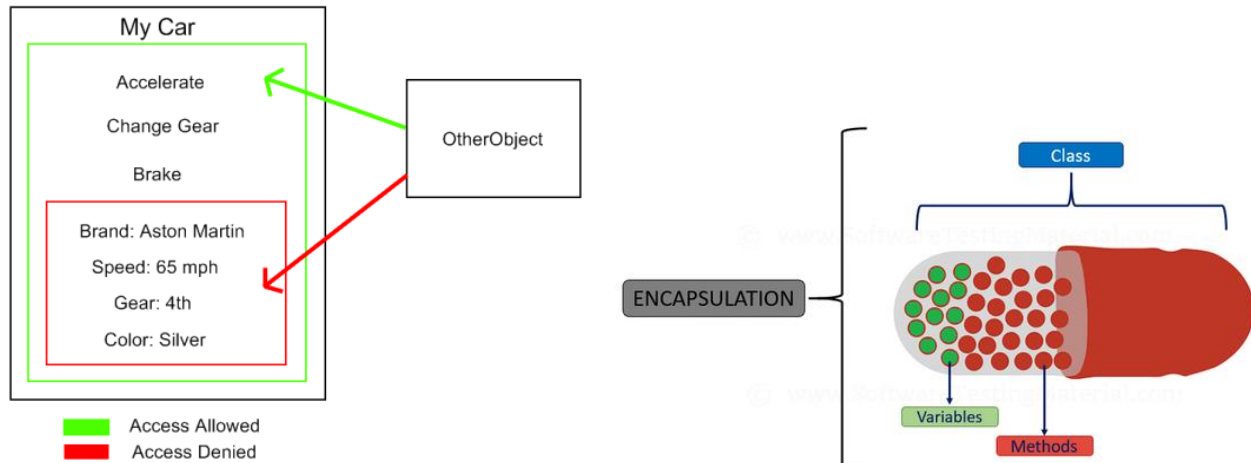
Сваки објект може примети и слати поруку другим објектима; објект класе Dugme може примети поруку да је миш у његовом фокусу или да је мишем притиснуто дугме (један објект шаље поруку другом објекту којим се спроводи акција).

Енкапсулација

Енкапсулација подразумева да корисници неког објекта не могу мењати унутрашња стања и методе објекта.

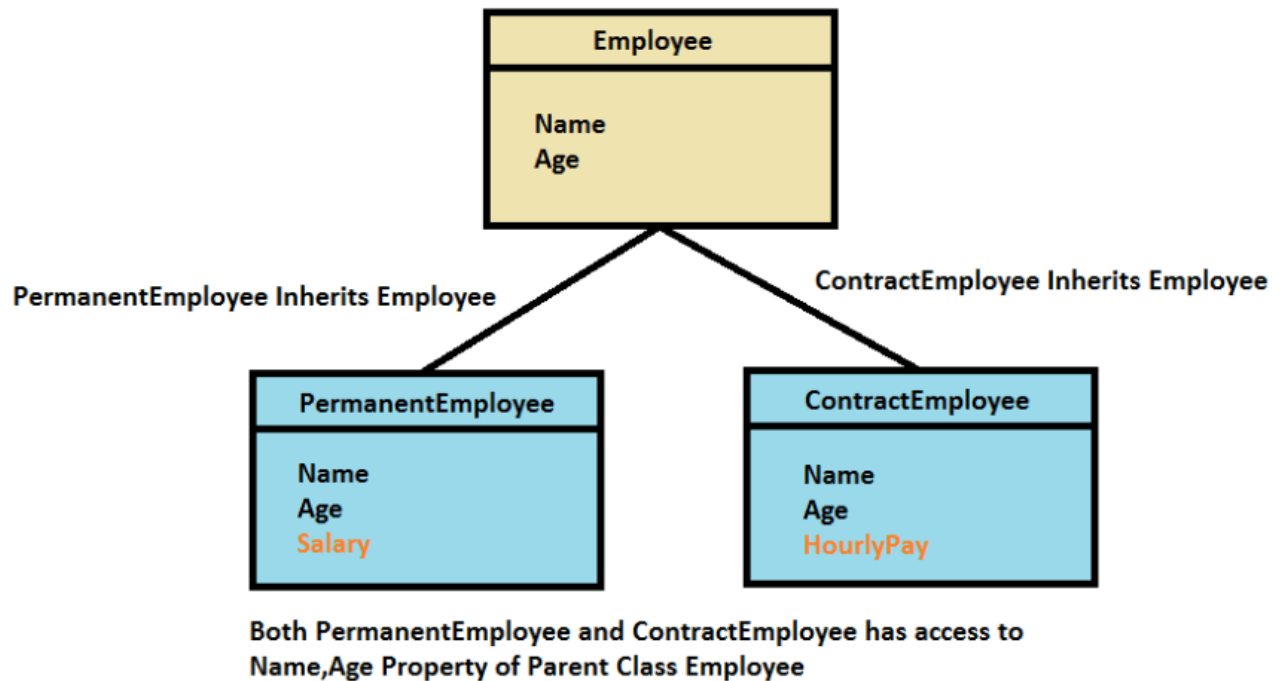
Само интерним методама објекта омогућено је да приступе тим стањима.

Сваки објект омогућава спољним корисницима да приступе и користе његове јавне методе помоћу интерфејса.



Наслеђивање

Наслеђивање се користи у ООП као метода за моделовање објеката.



Може се рећи да се класа Employee специјализује у класе PermanentEmployee и ContractEmployee. Може се рећи да класе PermanentEmployee и ContractEmployee генерализацијом праве класу Employee.

Полиморфизам

Обе класе наслеђују особине Name и Age а може се креирати метода nacinZarade која ће деловати на објекте класе PermanentEmployee тако што ће се обрачунавати као месечна плата а на објекте класе ContractEmployee тако што ће се обрачунавати као зарада по сату.

Особина да исти метод делује различито у зависности од објекта на којем се примењује, назива се полиморфизам.

Методе и атрибути

У ООП постоје програми као скупови објеката и скупови догађаја (event) и порука између њих.

Објекти реагују на догађаје а редослед реакција зависи од догађаја и поруке, што је у складу са реалним светом.

Посматрањем свих објеката једног система долази се до њихових особина – атрибута и акција које ти објекти могу увести – методе.

Пример:

```
class Ucenik
{
    String ime, prezime;
    DateTime datumRodjenja;
    int razred, SkolskaGodina;
    char odeljenje;
    int[] ocene;
    double prosek()
    {
        ...
    }
    void uci(Lekcija)
    {
        ...
    }
}
```

Атрибутима описујемо особину објекта (ime, prezime, datumRodjenja, razred, odeljenje).

Вредности атрибута дефинишу стање објекта.

Методом описујемо понашање објекта у одређеној ситуацији и под одређеним условима (uci) али и одређујемо нове вредности на основу особина које објекат поседује (prosek).

На овај начин се описује функционалност објекта.