class & object - Mario Example 1) Classes & objects 2) Emapsulation 3) Abstraction 4) Imheritamie 5) Polymosphison 000 -> object -> pormitue datatype why we need oop? Mario Game: 3) Char 1) pleyer 4) Boolean 2) position 3) Sie, Health. Access through methods what is an object? (mistame of class)

templete of assict radity tolo object dat smerodox

Bluepoint

Actual Home.

class & object \_ Mario Example

Enemy

Class

Sing

Postbron

more ()

objects

Enemy !

Enemy 2

Eherry 3

cop -> Object -> Parinitive 1) Encapsulation :

Encapsulation is grouping data with

methods in a class.

Hedring data prevent direct access

from outside.

3) Size Health Access through methods

Rebrieve Into Modern Settler Into

Actual House.

radity Info.

Cheek corrector ht

2) Abstraction Super class only show essential detail Hide all other detail keybard tours ] monitor usp somer face Madeless. Hode Rem Hard dak partery sev 2) Banicole Tomplementation Interfere Hidden from orbeide show orberde chase company ( keyboard Herdu-Moul PAM ( wondfor Il you white be subclim too only get position 2) Statu Polymerphusin 3) Inheritane: Allow classes to us data & mellod from another clase. mony method with some nam ? Tarke all mediad overleadons Mary

Superclase industrial (a enty show exentered delaste & Subclassi Subclass 2 subclass 3 Subclassy no Subclass interit from Superclass Forteritare - Access Modifiers. 1) Public uss produced 2) Printale 3) protiected 4) Polymorphoms (Rime time) 1) Pyromi'l (Both Superclar & Sub'class chance Serme modeled). 2) Statu It you write a Sub class for object motor de la will work 2) Statu Polymorphism 3) Tookertone: accuss during compile time many methods with some ram & different approment is the same class Function overloading mested overloading move limite buchter) algunit

## OBTECT ORIENTED PROGRAMMINGS

- 1) Classes and objects
- 2) Ensapsulation
- shired 3) Abstraction assob assess and adds probable
  - 4) Inheritence
  - 5) Polymor phism

00 PS -> Object promote roll) rolled

Primitive data type;

Complex data Group similar pomnitare data together.

class & object:

Class Objects

Size Eveny!

[Osition Eveny 2

movel)

Class Objects

Eveny 3

dead:

class -> Template (Home bluepoint)
object -> Actual home (used for object)

object -> object is an istance of class class -> is a template for object

encapsulation

(生物)

Encapsulation: 3 MIMMAS 10049 COTTO TOSTED Encapsulation is grouping claba with methods in a class. Hidring data. Prevent direct access from outside Access through methods Class Enemy & Setter Retrieve Information Modify Intornation (Change Enemy) position) Checkcorrector not Eveny ! class Hayer class themy class norme class Item Class Background object - object is an istance of class we can find essor easily through

encapsulation

Abstraction or only Show ossential detail - Hide all other detail key Board Trankpad USB Implementation Interface Hidden toom outside Shown outside class class tremy: Modera no need to show get position () Impletation Inheritance: Allows Classes to use date 68 method form 1) Srigle Inheritance another class (Ossan) Class enemy 3) Multiple & class monkey class Octopus class dragon Super class Sub class: 1 Sub Class 2 Subclass 3 Sub class 4

modifiers : Access change where class data com be oucemed form, 1) Public ( we can used is all) & private (only is superclass) D) protected (we can use in subclass), Polymosphism: (wany forms) O Dyramic polyonophism 2) Statie Holymorphiston. ) occurs during own time and sould dear Both Superlass and Subclass Lave Some method ale position class Brenny move () Class montey class octobre (morel) and ottom one more is Statec Polyonoophism. Ocurs during complile time () wany meturals with some name & differen organients in the Some days called or method or function

class - Enemy more (mario\_ location): detects the location and have to more differ towards marios to attack X THE CER SQL Queries: select - select data from database 6 town - which table weave pulling town a where - Condition we can spenty a condition as - alian - Revand Column or table with en alian 3 join - add two or more rows form subserent tables And - a Both conditions must be ment only one andition is amough consit-limit. roows & returned in - sperified values when we use whole case - return value of the sourspectived and is will- Bow rathe us null value · Like - sporch for pattern in column commit - write tradition to dataford Roubant winds the tornants block ide date in the loading database

Object Oriented Programming with python: 1) Abstract class 2) Attributes & Objects 3) polymorphism ton is ( 130 4) Abstraction of a = ( 131 mans) 310720 5) Encapsulation 6) netwood kinds self name = name Methods -> "def bunctions" inside class are methods. Formated strong: Historia & disc 6" A instance created & rame? I pormated storing. assert -> defines wheather positive or negative or ranges the value. Augument >? (1/601 "onong") moss = 1 moss instances -> ? devotators >? ) our later rotal price () = souther substitute of class -> Destign Objects -> Instances Next to 20% discount: Magric attribute: mit (clery - dict - - ) # All attarbites to class level

int literal -- dell --) # 14 attributes for tristance lovel.

Class Item: Atten paramoneral destricted to the det \_ mit\_ (self, name: str, paice: float, quantity=0) assert price >= 0, t" price { price} is not greater than or equal to zero!"

assert Quantity >= 0 6" Quantity & quantity is ust greater on equal to sero! "

Augment - 0

self name = name shower self. Price = price routing job & should Selb - Quantity = Quantity

# calculation for price (function or method).

det calculate total price (self): assest - defi seturn selb. Price \* selb. Quantity

item = item ("phone," 100, 1) item 2 = item ( Laptop", 1000, 3)

Point (item). Calculate Total-price ()) print literir calculate Total-prine(1) objects > Instances

Next to 20% discount: Magic attribute:

Point (item - dict -- +) # All outtoibules box class level Print literut. -- dict -- ) # All attributes for Instance level 1.40

-- seps-- to show the attributes and mistorices.

CSV -> Comma Separated value

class software engineer:

# class attribute

alias = "keyboard Majiciam" housemakers is

deb - mit (Selb, name, age, level, Salary):

Selb name = hame # mistance attainmes:

Selb age = age

Selb level = level

Selb Salary = Salary

# instance (object)

Sel = Softwareenginee ("max", 20," "junios", 5000)

Point (sel name, age)

1) instance can't be used in the whole class
2) But, class attribute can be used in whole class
Eunction in classes: (method) or (instance method)

def add (a, b): parameters and the second of the second of

add (2,3)

Arguments

deb -str\_ (selb): Information of object d under method sular distances aminos & vas deb -- eq (selb, other): compare two objects class softwark engineer: # class attailede decorator -> ) @ statismethod " mosson broodpost" - souls Inheritence Gustes, Isual ages lamon also) time\_ doo one class takes the attributes and methods of other class. The newly created class is called cs child dass and pre-enisting one is parent class # inherit, override, entend. probas probas also Super (). \_\_init \_ Cname, age, Salary) C) cused to Applicable the inheritance.

override entends. Parit (see name, age) # inherit, override, extends 1) instante can't be used in the whole class class employee: deb - init\_ (selb, name, age, salary): Self-age = age

self. Salary = salary (dis) bbs job

det work (self):

point Lf"[self: name3 is working...")

Agaments ...

class sobtuare Engineer (Employee): deb\_intit-- (selb, name, age, salary, level): Super() \_ init \_ (name, age, salary) self level = level def work (self): prot (+" { self name 3 is coding -- ") deb debug (selt): 2) inhappy be overed extend point 16" Eselb name 3 is debugging. class pessames (Employee): mosting countries to desired to def work (self): Print ( t" E selb name 3 is designing ") intologous deb draw (selb): lob set prishit to resurction ( point (b" Eselb name y is draw ring ") se = Sobtware engineer ("Max", 25, 6000, "Junior") se-work () should platagnos d= designer ("Philipp", 27, 7000) Single understore (protected) (1 strow. b double undersure (private) detrils are valed -

Abstraction is tru natural extension of the Emaperhation."

Polymorphism: - (Many borns)

Polymorphism gries is a way to use a Class emactly as a parent class , but still each child class keeps its method.

- 1) inhoritance: child class (Base class)
- 2) inherit, overside, extend (disc) goden don
- 3/ Super() init\_() when when we change from Base clarge
- 4 loncept of Polymosphiton. ( )

Encapsulation: - Gring 3286 or Esman dos? " & I time?

- 1) Mechanism of Hiding the data implemention
- 2) instance the methods are kept poirate.
- 3) Single underscore () to create private instance attribute
- 4) double underscore (--) for no attribute or

Logou never see tuis in real obten.

single underscore (protected)
double underscore (private)

Bonk details are valid

"Abstraction" is the natural extension of the Emapsulation."

mey don't case about a interal calculation ob the calary. They only care about : (OF) diese in 1 est the output.

THAT HERE IS IN

class Software Engineer ():

deb \_ inita (self, name, age):

selb name = name

self rangue = age

self. - salary = Nome # No need to detrine in Sett. \_mm\_bigs\_solved = 0

det Code (self):

Self: inum-bugs-solved+=1 masses malages del solary (self, value):

# getter

def get\_salary (Selt):

return & Selt-Salary eleb calling (self):

# setter

male - 6193 100 ... # wheck values, constraints enforcement, check details.

deb Set-Salary (self, base value):

Sebb. - Selary = Selb: - Calculate Salary (tase-value)

det calculate salary (self, base value):

of sell-num-bugs solved closing seturn base value

is selb \_nm\_ bugs \_ solved <100:

return terse value + 2

return bese value # 3

se = softwarenginess () Print (se age, se name) too 1 in sange (70): se code () se-set-salary (6000) Point (se. get-salary ()) x - dewratore self acione = age in @ property Joseph on # smon = produce - flos det salary (selt): = bonde quid man door Return Selt salary det code (self): @ Salary. Setter 1=+ bould good man' the def Salary (self, value): self. solary = value (1) 2) produce too tob - + getlor @ solary deleter produce des que mondas det saldry (self): del Selb--salary # wheele values, constraints enforcement, theek details Set salary liets tesse relied: -Stole - sclary = telb - Calculate to lary (tase-value) Cretter & Setter are only may to access the private and protected values in code 1) getter -> @ proposty + 2) setter - X x setter 2000 menter neturn tens value of 3

+1+

DOP'S PRINCIPLES :

Inhoritance + 10 HARMAN 30 MM CONTINUE TO THE TOTAL OF THE PARTY OF TH Inheritance is the process of by which one class takes on the attributes and methods of others. newly formed classes are called child class, and the classes that child classes are derival boom are caued parent classes.

crechan half

child classes inherit all of the parent's attributes and methods but can also extend and oversites attributes and methods that are unique to thenselves.

Polymosphism:

" Many Shapes"

we can write a code that works on the superclass and it will work with any subclass type as well

Gives us a way to use a class exactly like its parent but each child class keeps it's own method as they are,

Encapsulation:

Encapsulation is the mechanism of hiding of data implementation.

Instances variables are rept private and accessor methods are made Public to achieve this with this , we restrict access to public methods cgetter / setter).

kept private Instance methods can also

Abstraction :-

Abstraction can be thought up as an natural extension of encaysulation. Applying abstraction meons that such object should only expose a high-level

mechanism for asing it.

The mechanism Should hide to internal implementation details. It should only reveal operations relevant for the other objects. child classes where

attributes and metacis but the other extend and overlites attailectes and methods their are unique

DOP S PRINCIPLE

we can write a cade that works on the superclass

Gives us a way to use a class exadely like its pasent but out thild class leaps it's own

Enlapsulation is the mechanism of hiding of date inflamentation.

accessor methods are made pather to achieve this with this pure restrict acress to reduce methods ighter / subject).

Instance instructe an also rept parato