26/07/2022

Java Notes Day4

Day 4
1. Association in Inheritance
2. static, final and abstract keywords
3. Types of methods

```
                                Association
                isA         hasA        usesA       producesA
                |           |           |           |
                Aggregation     Composition Object as   Object as
                                            method arg  method return
type
```

Class ---> data + functions
Initializing an object of another class as a global variable of the class.
Can be initialized only using reference
The object contains null as the values within

isA - extends
- >Employee isA Student, Person
- >Student isA Person

hasA - containment, consists of within
- >hasA is non linear, falls under one parent horizontally
- >Employee hasA salary, job
- >Student hasA grade, qualification

class 0
class1 > objRef(class 0) >

Passing an object as an argument for amother object
Stack memory
Heap memory (Free store)

```
                                    FIELD AREA(100)
FIELD AREA(200)

ref to an obj (obj1) is created -------->accNumber  accHolderName
balance  ref1(DL) ---> name  type  issueDate
100                                                              |200
                                                  This creates a
reference to another object
BankAccount obj1 = new BankAccount();             METHOD AREA
 |         |         |      |

 |         |         |      |                     BankAccount()
Type      ref  allocator constructor              withdraw()
                                    deposit()
                                    etc
```

---

	static	final	abstract
field	shared across all objs of the class	cannot be changed immutable	NA
method	is used to refer other static data /method members by the child	cannot be overriden It is the partial contract	of a class
class	NA	cannot be extended	It contains the partial contract of class

---

Class data - Static data which can be accessed with class name since it belongs to the class
- Preloaded in the memory
- Stored in the Stack memory

Object data - Can only be accessed through instance of a class(object creation)
- Stored in heap storage
- Loaded in the memory only after object creation

Static - static int - can be accessed multiple times
private static int - can be accessed only within the class
void showDets() - can only be accessed through object
static void showDets() - can be accessed multiple times w/o object
Static function cannot access non-static data, but non-static functions can access static data

static object reference

final - data member value cannot be changed - but can be created multiple times
final - function is created and it cannot be overridden, only inherited
final - class is final, and it cannot be extended by another class, last hierarchy of the project
static - data member can obly be created once

abstract = incomplete = partial
The method declared in the parent class using the abstract keyword, is a mandate for the child class to implement a method of the same name only

4 types of methods
> Exclusive
> Inherited
> Overriding
> Implemented

```
  parent      child       child
abstract class1 > abstract class2 > class > contract on this class for
implementation of both abstract classes 1 & 2
      |          |
abstract method1 abstract method2
```

abstract class is only a mansate and it does not contain any code
only reference of an abstract class can be created, and not the object

Super class reference can point to a child class

Early binding - compile time binding - objReference.methodOfSameClass()
Late binding(lazy loading)(polymorphism) - run time binding - objReference.methodOfParentClass() / parentReference

```
                                    |-------------------------------------
------------> At run time the decision changes
```

->Super class reference can point to any one of the child class object - eg - BankAccount ba = new BankAccount(); SavingsBankAccount extends BA

similarly, it works for abstract classes, since only ref can..
SavingsBankAccount sa = new SavingsBankAccount();

ba = sa;

-> only works for inherited methods

-> not exclusive methods of the class - Doctor knows diagnose, but not HeartSurgery exclusive methods