***Difference between Array and ArrayList with example.  
Solution:>***

Array:

· An array is a dynamically-created object. It serves as a container that holds the constant number of values of the same type. It has a contiguous memory location.

· Array is static in size.

· We can add elements in an array by using the assignment operator.

public class ArrayExample

{

public static void main(String args[])

{

//creating an array of integer type

int arr[]=new int[4];

//adding elements into array

arr[0]=12;

arr[1]=2;

arr[2]=15;

arr[3]=67;

for(int i=0;i<arr.length;i++)

{

System.out.println(arr[i]);

}

}

}

ArrayList:

The ArrayList is a class of Java Collections framework. It contains popular classes like Vector, HashTable, and HashMap.

ArrayList is dynamic in size.

Java provides the add() method to add elements in the ArrayList.

import java.util.\*;

public class ArrayListExample

{

public static void main(String args[])

{

//creating an instance of ArrayList

List<Float> list = new ArrayList<Float>();

//adding element to arraylist

list.add(12.4f);

list.add(34.6f);

list.add(56.8f);

list.add(78.9f);

//iteration over ArrayList using for-each loop

for(Float f:list)

{

System.out.println(f);

}

}

}

***Draw the class diagram and develop the logic for the following scenario. Consider the class Cuboid with length, breadth, and height as private attributes. Code the parameterized constructor and toString() method Develop main() method that stores data of Cuboids as a collection and must be menu driven with the following operations: 1. Add new Cuboid, 2) Sort by length, 3) Sort by area [HINT: Area of cuboid is length "breadth\*height]***

**package** p1;

**public** **class** Cuboid

{

**private** **int** length,breadth,height;

Cuboid(**int** l,**int** b,**int** h)

{

**this**.length=l;

**this**.breadth=b;

**this**.height =h;

}

**public** **int** getl() {**return** **this**.length;}

**public** **int** getb() {**return** **this**.breadth;}

**public** **int** geth() {**return** **this**.height;}

**public** **int** getA() {**return** (getl()\*getb()\*geth());}

**public** String toString()

{

String str="L="+ getl()+ "\tB=" + getb()+ "\tH=" + geth();

str= str +"\tArea="+getA();

**return** str;

}

}

**package** p1;

**import** java.util.Comparator;

**import** java.util.Collections;

**public** **class** SortByArea **implements** Comparator<Cuboid>

{

**public** **int** compare(Cuboid c1, Cuboid c2)

{

**return** (**int**)(c1.getA()-c2.getA());

}

}

package p1;

import java.util.Comparator;

import java.util.Collections;;

public class SortByLength implements Comparator<Cuboid>

{

public int compare(Cuboid c1, Cuboid c2)

{

return (int)(c1.getl()-c2.getl());

}

}

package p1;

import java.util.Collections;

import java.util.ArrayList;

import java.util.Scanner;

public class Mainmenu

{

public static Scanner sr=new Scanner(System.in);

public static void main(String args[])

{

ArrayList<Cuboid> al=new ArrayList<Cuboid>();

while(true)

{

System.out.println("\n\*\*\*Main menu\*\*\*");

System.out.println("1)Add new Cuboid\n2)Display all\n

3)Sort by Length\n4)Sort by Area\n5)Exit");

System.out.println("Enter your choice");

int cho=sr.nextInt();

switch(cho)

{

case 1:

System.out.println("Enter the Length, Breadth and Height");

Cuboid c= new Cuboid(sr.nextInt(),sr.nextInt(),sr.nextInt());

al.add(c);

System.out.println("Cuboid added");

break;

case 2:

for(int i=0 ;i < al.size();i++)

System.out.println(al.get(i));

break;

case 3:

System.out.println("Sorted by Length");

Collections.sort(al, new SortByLength());

for(int i=0 ;i < al.size();i++)

System.out.println(al.get(i));

break;

case 4:

System.out.println("Sorted by Area");

Collections.sort(al, new SortByArea());

for(int i=0 ;i < al.size();i++)

System.out.println(al.get(i));

break;

case 5: System.exit(0);

default: System.out.println("Please enter 1 to 5");

}

}

}

}

***Consider a set of car names given as inputs through console and store in an ArrayList. The task is to sort and print them using iterator.   
Solution:>***

package seven;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Scanner;

import java.util.Collections;

public class Eight\_Key {

public static void main (String[] args) {

ArrayList<String> a1=new ArrayList<String> ();

Scanner sc=new Scanner (System.in);

for (int i=0;i<5;i++)

a1.add (sc.next());

sc.close();

Collections.sort (a1);

Iterator itr=a1.iterator ();

while (itr.hasNext()){

System.out.println(itr.next());

}

}

}

***Develop an abstract class Employee with ID, name and address as attributes and an abstract method computeSalary():double. The classes ContractEmployee and RegularEmployee inherit Person. ContactEmployee has hoursWorked and hourlyWage as its attributes and the RegularEmployee has basic and allowance as attributes. The rules for the computation of salary are as follows:***

1. ***For Contract employees the salary is product of hoursWorked and hourlyWage.***
2. ***For Regular employee the salary is basic + HRA(20 % of basic) + allowance.***

***Write the Demo class with main() method to create objects of both ContractEmployee and RegularEmployee and compute the salary. Store the data in an ArrayList .***

***Solution:>***

import java.util.Scanner;

abstract class Employee ------------🡪2.5

{

public int id;

public String name;

public String address;

Employee(int id,Stringname,String address)

{

this.id=id;

this.name=name;

this.address=address;

}

public abstract double computeSalary();

}

class ContractEmployee extends Employee ------------🡪2.5

{

 private int hoursworked;

 private int hourlywage;

 ContractEmployee(int id,Stringname,Stringaddress,inthoursworked,inthourlywage)

 {

super(id,name,address);

this.hoursworked=hoursworked;

this.hourlywage=hourlywage;

 }

 public  double computeSalary()

 {

return hoursworked\*hourlywage;

 }

}

class RegularEmployee extends Employee   ------------🡪2.5

{

public int basic;

public int allowance;

RegularEmployee(int id,Stringname,Stringaddress,intbasic,int allowance)

{

super(id,name,address);

this.basic=basic;

this.allowance=allowance;

}

public  double computeSalary()

{

return basic+ (basic\*20/100)+allowance;

}

}

public class Demo ------------🡪2.5

{

public static void main(String args[])

{

Employee e;

Scanner  sc=new Scanner(System.in);

System.out.println("Enter id and name and address of ContractEmployee:");

int id=sc.nextInt();

String name=sc.next();

String address=sc.next();

System.out.println("enter hoursworked and hourlywage :");

int hoursworked=sc.nextInt();

int hourlywage=sc.nextInt();

e=new ContractEmployee(id,name,address,hoursworked,hourlywage);

System.out.println("salary is:"+e.computeSalary());

System.out.println("Enter id and name and address of RegularEmployee:");

id=sc.nextInt();

name=sc.next();

address=sc.next();

System.out.println(" enter basic and allowance");

int basic=sc.nextInt();

int allowance=sc.nextInt();

e=new RegularEmployee(id,name,address,basic,allowance);

System.out.println("salary is:"+e.computeSalary());

sc.close();

}

}

Class Diagram -------------🡪2.5

|  |
| --- |
| P1 |
| Employee |
| 1. ID : long 2. name : String 3. gender : String 4. salary : double |
| 1. getID():long 2. getName():String 3. getGender(): String 4. getSalary():double   + Employee()  + Employee(i:long,n:String,g:String,s:double)  + toString():String  + main(args[] : String) : void |

package p1;

public class Employee

{

private String id;

private String name;

private String gender;

private double salary;

Employee()

{

this.id="#";

this.name="#";

this.gender="#";

this.salary=0.0;

}

Employee(String i,Stringn,Stringg,double s)

{

this.id=i;

this.name=n;

this.gender=g;

this.salary=s;

}

private String getID()

{

return this.id;

}

private String getName()

{

return this.name;

}

private String getGender()

{

return this.gender;

}

private double getSalary()

{

return this.salary;

}

public String toString()

{

return String.format("%nID : %s%nName : "

+ "%s%n"

+ "gender : %s%n"

+ "Salary : %f", getID(),

getName(),getGender(),getSalary());

}

}

package p1;

import java.util.ArrayList;

import java.util.Scanner;

public class EmployeeDetails

{

private static ArrayList<Employee> al=

new ArrayList<Employee>();

public static void main(String[] args)

{

boolean repeat = true;

Scanner s = new Scanner(System.in);

while(repeat)

{

System.out.println("Empolyee Database");

System.out.println("1. Create an New "

+ "Employee record");

System.out.println("2. Print All Employees");

System.out.println("3. Exit Employees "

+ "Database");

System.out.println("Enter U r choice");

switch(s.nextInt())

{

case 1:

System.out.println("Enter Employee ID,"

+ "Employee Name\n, Employee "

+ "Gender, Employee Salary:");

Employee st = new Employee(

s.nextLine(),

s.nextLine(), s.nextLine(),

s.nextDouble());

al.add(st);

break;

case 2:

System.out.println("All Employee "

+ "Details");

for(Employee e: al)

System.out.println(e);

break;

default:

repeat=false;

}

}

}

}

***Draw the class diagram and develop a class Employee (with ID, name, gender, salary as private instance members), define the parameterized constructor and format the output as***

***ID: 1234***

***Name: ABC***

***Gender: Male***

***Salary: $2400***

***The main () method must be menu driven with the following options:***

1. ***Add New Employee***
2. ***Print all Employee data***
3. ***Sort based on salary***

***Store the data in an ArrayList.   
Solution:>***

package Test\_2\_ArrayList;

import java.util.Comparator;

public class SortbySalary implements Comparator<Employee>

{

public int compare(Employee e1, Employee e2)

{

return (int)(e1.getsal()-e2.getsal());

}

}

------------------

package Test\_2\_ArrayList;

public class Employee

{

private long id;

private String name;

private char gen;

private double sal;

Employee(long i,Stringna,charg,double s)

{

this.id=i;

this.name=na;

this.gen=g;

this.sal=s;

}

public long getid() {return this.id;}

public String getname() {return this.name;}

public char getgen() {return this.gen;}

public double getsal() {return this.sal;}

public String toString()

{

return "Id ="+getid()+"\tName="+getname()+

"\tGender="+getgen()+ "\tSalary="+getsal();

}

}

---------------------------

package Test\_2\_ArrayList;

import java.util.Scanner;

import java.util.ArrayList;

import java.util.Collections;

public class Employee\_Menu

{

public static Scanner sr=new Scanner(System.in);

public static void main(String[] args)

{

ArrayList<Employee> al= new ArrayList<Employee>();

while(true)

{

System.out.println("\n\*\*\*Main Menu\*\*\*");

System.out.println("1) Add Employee ");

System.out.println("2) Display all Employee");

System.out.println("3) Sort wrt salary");

System.out.println("4) Exit");

System.out.println("Enter your choice");

int cho=sr.nextInt();

switch(cho)

{

case 1:System.out.println("Enter an employee detail");

Employee e=new Employee(sr.nextLong(), sr.next(),

sr.next().charAt(0), sr.nextDouble());

al.add(e);

break;

case 2:System.out.println("The lsit of Employee are:");

for(int i=0;i<al.size();i++)

System.out.println(al.get(i));

break;

case 3:System.out.println("The sorted lsit based on salary:");

Collections.sort(al, new SortbySalary());

for(int i=0;i<al.size();i++)

System.out.println(al.get(i));

}

}

}

}

**A website has data of movies. Every movie has name, yearOfRelease and rating as attributes. The tasks are to 1) sort by yearOfRelease 2)Sort by rating. (8M)**

package mytes2;

import java.util.Comparator;

import java.util.\*;

class Movie

{

String name;

int year,rating;

Movie(String name,intyear,int rating)

{

this.name=name;

this.year=year;

this.rating=rating;

}

public String toString()

{

String msg;

msg=String.format("name=%s yearOfRelease=%d Rating=%d",name,year,rating);

return msg;

}

}

class sortByYear implements Comparator<Movie>

{

public int compare(Movie a,Movie b)

{

return a.year-b.year;

}

}

class sortByRating implements Comparator<Movie>

{

public int compare(Movie a,Movie b)

{

return a.rating-b.rating;

}

}

public class MovieDetails {

public static void main(String args[])

{

ArrayList<Movie> list=new ArrayList<Movie>();

int count=0;

Scanner s=new Scanner(System.in);

boolean repeat=true;

while(repeat)

{

System.out.println("1.new Details 2. Display details 3.sortbyyear 4.sortbyrating 5.Exit");

System.out.println("enter your choice");

int ch=s.nextInt();

if(ch==1)

{

System.out.println("enter movie name,yearofrelase and rating");

String name=s.next();

int year=s.nextInt();

int rating=s.nextInt();

list.add(new Movie(name,year,rating));

}

else if(ch==2)

{

System.out.println("details are");

for(int i=0;i<list.size();i++)

{

System.out.println(list.get(i));

}

}

else if(ch==3)

{

Collections.sort(list,newsortByYear());

System.out.println("details are");

for(int i=0;i<list.size();i++)

{

System.out.println(list.get(i));

}

}

else if(ch==4)

{

Collections.sort(list,newsortByRating());

System.out.println("details are");

for(int i=0;i<list.size();i++)

{

System.out.println(list.get(i));

}}

else if(ch==5)

{

repeat=false;

}

}

}

}

***Sample Questions:-***

* 1. ***The Shape Demo class with main () method which is menu driven has the following operations- Add New Geometric Shape, Print details of all Geometric Shapes. (Data to be stored in an ArrayList)***
  2. ***The setAge() throws an InvalidAgeException when age is not within the range 18-120. The VoterDemo class has main() method to store data of 5 voters in an ArrayList and the exception must be handled in main() method. Print the data of voters.***
  3. ***Develop a student class with the following fields: ID, name, marks of 6 subjects(array) as a private instance member and corresponding mutator and accessors and toString() method.***

Write a Menu Driven Java Program to

1. Add a new student and
2. Display all Student Details.

The class diagram is as follows:

***Diagram

Description automatically generated***

***OUTPUT :  
Press 1 for Add New Student 2 for Print details of all students 3 for Exit***

***1***

***enter id, name***

***1***

***sarvani***

***enter marks of subject 0 50***

***enter marks of subject 1 49***

***enter marks of subject 2 45***

***enter marks of subject 3 40***

***enter marks of subject 4 50***

***enter marks of subject 5 50***

***Press 1 for Add New Student 2 for Print details of all students 3 for Exit***

***2***

***Student [id=1, name=sarvani, marks=[50, 49, 45, 40, 50, 50], total=284]***

***Press 1 for Add New Student 2 for Print details of all students 3 for Exit***

***3***

***Exiting the application***

1. ***Enhance the main () method of Demo class to display a menu of operations as follows:***

***1. Add new Student***

***2. Print details of all students***

***3. Search a student based on ID***

***4. Search based on name***

***5. Modify name based on ID***

***6. Sort based on ID***

***7. Sort based on total***

***8. Exit***

***The program must store details of 10 students***

Diagram

Description automatically generated