Class Projects path:

C:\Users\megan\myeclipseprojects

<https://www.simplilearn.com/ice9/pdfs/agenda/masters-program/Full_Stack_Java_Developer_Masters_Program_Brochure.pdf>

<https://community.simplilearn.com/threads/phase-1-implement-oops-using-java-with-data-structures-and-beyond-july-2021-batch-1-meganadha-reddy-july-10-aug-07.66829/>

==================================================================

<https://github.com/meganadh>

==================================================================

Oracle Documentation:

<https://docs.oracle.com/en/java/javase/16/docs/api/index.html>

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

package mypackage;

public class MyClass

{

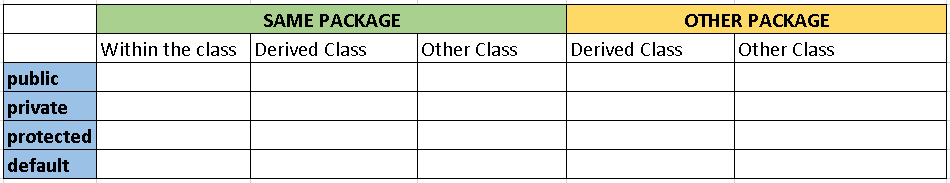
public static void main(String[] args)

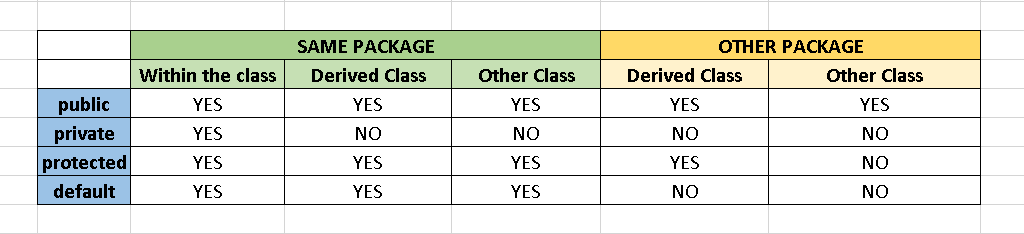
{

System.out.println("Hello, World");

}

}





package package1;

public class MyPackage1BaseClass

{

public int a;

private int b;

protected int c;

int d;

public void myPackage1BaseClassMethod()

{

a=5;

b=10;

c=11;

d=12;

}

}

class MyPackage1DerivedClass extends MyPackage1BaseClass

{

public void myPackage1DerivedClassMethod()

{

a=5;

b=10;

c=11;

d=12;

}

}

class MyPackage1OtherClass

{

public void myPackage1OtherClassMethod()

{

MyPackage1BaseClass obj = new MyPackage1BaseClass();

obj.a=10;

obj.b=11;

obj.c=12;

obj.d=13;

}

}

package package2;

import package1.MyPackage1BaseClass;

public class MyPackage2DerivedClass extends MyPackage1BaseClass

{

public void myPackage2DerivedClassMethod()

{

a=5;

b=10;

c=11;

d=12;

}

}

class MyPackage2OtherClass

{

public void myPackage2OtherClassMethod()

{

MyPackage1BaseClass obj=new MyPackage1BaseClass();

obj.a=10;

obj.b=11;

obj.c=12;

obj.d=13;

}

}

===================

CLASS-3

===================

Object Oriented Programming

Methods

Constructors

Pillars of Object Oriented Programming

======================================

1. Encapsulation

2. Abstraction

3. Inheritance

4. Polymorphism

package companyDetails;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

//Variable declaration

String name;

Scanner obj = new Scanner(System.in);

//Read data from user

System.out.println("Enter your name:");

name=obj.nextLine();

System.out.println("Your name is "+name);

}

}

package companyDetails;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

//Variable declaration

String name;

int age;

float height;

Scanner obj = new Scanner(System.in);

//Read data from user

System.out.println("Enter your name:");

name=obj.nextLine();

System.out.println("Enter your age:");

age=obj.nextInt();

System.out.println("Enter your height:");

height=obj.nextFloat();

System.out.println("Your name is "+name);

System.out.println("Your age is :"+age);

System.out.println();

}

}

===================================================

package companyDetails;

import java.util.Scanner;

public class MyClass

{

static int factorial(int n)

{

int i,fact=1;

for(i=1;i<=n;i++)

fact=fact\*i;

return fact;

}

public static void main(String[] args)

{

int n1=4,n2=5,n3=7;

System.out.println(factorial(n1));

System.out.println(factorial(n2));

}

}

==========================================================

Normally variables will be private

Method will be public

Note:

This is not mandatory. Depends on projects architecture.

package companyDetails;

import java.util.Scanner;

public class Employee

{

private String name;

private int age;

public static String company="Microsoft";

public void readData()

{

Scanner obj = new Scanner(System.in);

System.out.println("Enter Name:");

name=obj.next();

System.out.println("Enter age:");

age=obj.nextInt();

}

public void printData()

{

System.out.println("Name : "+name);

System.out.println("Age: "+age);

System.out.println("Company: "+company);

}

}

package companyDetails;

public class MyClass

{

public static void main(String[] args)

{

int a,b,c;

Employee emp1 = new Employee();

Employee emp2 = new Employee();

//Employee emp2 = new Employee();

emp1.readData();

emp2.readData();

emp1.printData();

emp2.printData();

System.out.println(Employee.company);

}

}

package companyDetails;

public class Maths

{

private int factorial(int n)

{

int i,fact=1;

for(i=1;i<=n;i++)

fact=fact\*i;

return fact;

}

public int nCr(int n, int r)

{

return factorial(n)/(factorial(n-r)\*factorial(r));

}

public int nPr(int n,int r)

{

return factorial(n)/factorial(n-r);

}

}

package companyDetails;

public class MyClass

{

public static void main(String[] args)

{

int n=5,r=2;

Maths obj = new Maths();

System.out.println(obj.nCr(5, 2));

}

}

Inheritance

===================

1. ENCAPSULATION:

        CREATING A CLASS WITH VARIABLES METHODS

   IN IT.

2. Inheritance:

       Re-usability

3. Abstract class:

      Re-usability and enforcing derived class to implement

  the abstract methods

     Template

4. Interface:

      pure abstract class.

      Interface is called - Contract

      keyword that we use - implements

5. Polymorphism

1. Method Overloading

         within the same class

            1. same method name

            2. irrespective of return type, parameters must be different.

        2. Method Overriding

         in base class and derived class, if you have

   same function, you can override your functionality in

derived class.

**=============================================**

**CLASS-4  (18th July 2021)**

**=============================================**

ARRAYS:

int a=5;

int[] data = {4,5,6,7};

int[] data = new int[]{4,5,6,7};

int[] data = {4,5,6,7};

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

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

for(int d:data)

System.out.println(d);

1. Array is collection of similar data types

2. Array index starts from zero.

3. Array requires sequential memory

4. Array size cannot be increased dynamically

5. Size(memory of array) = array size x data type size

short[] data = new short[20];

package mypackage;

import java.util.Scanner;

public class MyClass

{

//Write a program in Java to declare an array and read 5 numbers from user

//and print sum and average

public static void main(String[] args)

{

//Declare an array

int MAX=5;

int[] data = new int[MAX];

int sum=0, average;

Scanner obj = new Scanner(System.in);

//Read numbers from user

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

{

System.out.println("Enter any number:");

data[i]=obj.nextInt();

}

//Find sum

for(var d:data)

sum=sum+d;

//Find average

average=sum/MAX;

System.out.println("Sum="+sum);

System.out.println("Average="+average);

}

}

package mypackage;

import java.util.Scanner;

class Employee

{

private String name;

private int salary;

private int age;

Employee()

{

this.name=null;

this.salary=0;

this.age=0;

}

Employee(String name, int salary, int age)

{

this.name=name;

this.salary=salary;

this.age=age;

}

public void readData()

{

Scanner obj = new Scanner(System.in);

System.out.println("Enter name:");

name=obj.next();

System.out.println("Enter Salary:");

salary=obj.nextInt();

System.out.println("Enter age:");

age=obj.nextInt();

obj.close();

}

public void printData()

{

System.out.println("Name = "+name);

System.out.println("Age = "+age);

System.out.println("Salary = "+salary);

}

}

public class MyClass

{

public static void main(String[] args)

{

Employee emp = new Employee();

emp.printData();

}

}

A constructor is used to initialize class variables

[ when creating an object ]

The moment you write your own constructor

default constructor will be gone.

You can write any number of constructors for the class.

A constructor name should be same as your class name.

constructor should not have any return type., not even void.

<https://medium.com/zero-equals-false/arraylist-vs-linkedlist-vs-vector-22e1721a66b0>

=========================================================

CLASS - 5

=========================================================

package mypackage;

public class MyClass {

public static void main(String[] args)

{

//1-D array

int[] data = new int[] {4,5,6,7,9};

//2-D array

int[][] matrix = new int[][] {{1,2},{3,4},{7,9}};

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

{

for(int j=0;j<matrix[i].length;j++)

{

System.out.print(matrix[i][j]+" ");

}

System.out.print("\n");

}

for(int i=1;i<=6;i++)

{

for(int j=1;j<=i;j++)

{

System.out.print("\* ");

}

System.out.print("\n");

}

}

}

==============================================

JAGGED ARRAY

============================================

package mypackage;

public class MyClass {

public static void main(String[] args)

{

//Jagged Array

char[][] names = new char[][]

{

new char[] {'r','a','m'},

new char[] {'m','e','g','a','n','a','d','h'},

new char[] {'o','m'}

};

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

{

for(int j=0;j<names[i].length;j++)

{

System.out.print(names[i][j]+" ");

}

System.out.printf("\n");

}

}

}

============================================================

INNER CLASSES

===============================================================

package mypackage;

public class Mathematics

{

public final float PI = (float)22/7;

class Algebra

{

public int add(int a, int b)

{

return a+b;

}

}

class Geometry

{

public float circleArea(int r)

{

return PI\*r\*r;

}

public float circlePerimeter(int r)

{

return 2\*PI\*r;

}

}

}

==========================================================

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

//Object for Mathematics - outer class

Mathematics m = new Mathematics();

//Object for Geometry - Inner class

Mathematics.Geometry g = m.new Geometry();

System.out.println(g.circleArea(7));

}

}

=============================================================

package mypackage;

public class Mathematics

{

public final static float PI = (float)22/7;

static class Algebra

{

public static int add(int a, int b)

{

return a+b;

}

}

static class Geometry

{

public static float circleArea(int r)

{

return PI\*r\*r;

}

public static float circlePerimeter(int r)

{

return 2\*PI\*r;

}

}

}

=====================================================

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

System.out.println(Mathematics.Algebra.add(5, 6));

System.out.println(Mathematics.Geometry.circleArea(7));

System.out.println(Mathematics.Geometry.circlePerimeter(7));

}

}

===================================================================

package mypackage;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

int age;

Scanner obj = new Scanner(System.in);

char ch='i';

do

{

System.out.println("Enter your age:");

age=obj.nextInt();

if(age<=0 || age>100)

{

System.out.println("Please enter valid age");

}

else

{

System.out.println("Valid age");

ch='v';

}

}while(ch!='v');

obj.close();

}

}

============================================================

package mypackage;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class MyClass

{

public static void main(String[] args)

{

Pattern p = Pattern.compile("meg[0-9]?");

Matcher m = p.matcher("meg25");

if(m.matches())

System.out.println("VALID CODE");

else

System.out.println("INVALID CODE");

}

}

=====================================================

=======================================

Homework

=======================================

1. declare an array and print values of the array

 4 ways as discussed in the class.

2. write simple code to illustrate the difference between

   print and println

3. write code for below pattern:

\*  \*  \*

\*  \*  \*

\*  \*  \*

\*  \*  \*

4. write code for below pattern

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

5. Declare a 2 dimentional array with size 3x2 and print the

values.

6. Write a java code to read age from user and check if it is valid or not

   0<age<=100 is valid age.

   Keep asking user as long as he enters valid age.

7. Please write example code to illustrage usage of static

inner class.

8. \w and \W

   research and come up with what are word characters in regex

9. Research on the 6 regex that we left today and try to put

   a simple example for it.

10. Create a runnable jar file with multiplication table

   and ask one of your family member to take double click on it

   and give input

================

=============================

DAY-6

================================

package mypackage;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

public class MyClass

{

public static void main(String[] args)

{

try

{

//Step-1 Create File Object

File fl = new File("H:\\Hello.txt");

//Step-2 Create File Reader Object

FileReader fr = new FileReader(fl);

int i=fr.read();

while(i!=-1)

{

System.out.print((char)i);

i=fr.read();

}

}

catch(Exception Ex)

{

}

}

}

=============================================================

package mypackage;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.nio.file.Files;

import java.nio.file.Paths;

import java.util.ArrayList;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

try

{

//Variable declaration

var allLines = Files.readAllLines(Paths.get("H://Hello.txt"));

boolean isFound=false;

Scanner obj = new Scanner(System.in);

String name;

int score=0;

ArrayList<Integer> studentScores = new ArrayList<Integer>();

//Read data from user

System.out.println("Enter name of student:");

name=obj.next();

//Logic

for(String a:allLines)

{

String[] data=a.split(",");

if(data[0].equals(name))

{

studentScores.add(Integer.parseInt(data[1]));

isFound=true;

}

}

//Print output

if(isFound)

{

for(int i:studentScores)

System.out.println(i);

}

else

System.out.println("Student is absent for exam");

}

catch(Exception Ex)

{

}

}

}

============================================================

package mypackage;

interface Animal

{

String Sound();

}

class Duck implements Animal

{

@Override

public String Sound() {

return "Quack Quack";

}

}

class Dog implements Animal

{

@Override

public String Sound() {

return "Bow Bow";

}

}

public class MyClass

{

public static void main(String[] args)

{

Animal a;

a=new Dog();

System.out.println(a.Sound());

a=new Duck();

System.out.println(a.Sound());

}

}

==================================================

package mypackage;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.Serializable;

class Product implements Serializable

{

String name;

int price;

Product(String name, int price)

{

this.name=name;

this.price=price;

}

}

public class MyClass

{

public static void main(String[] args)

{

try

{

/\*

\* The below code is for Serialization

\*/

/\*

Product p=new Product("Sansung TV",2000);

//Step-1

FileOutputStream fos = new FileOutputStream("H:\\product.txt");

//Step-2

ObjectOutputStream oos = new ObjectOutputStream(fos);

oos.writeObject(p);

System.out.println("Object stored in file");

oos.close();

fos.close();

\*/

FileInputStream fis = new FileInputStream("H:\\product.txt");

ObjectInputStream ois = new ObjectInputStream(fis);

Product p=(Product)ois.readObject();

System.out.println(p.name);

System.out.println(p.price);

ois.close();

fis.close();

}

catch(Exception ex)

{

}

}

}

========================================================

EXCEPTION HANDLING

================================================================

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

String p="meg";

String s = p.substring(6);

System.out.println(s);

}

}

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

String p=null;

String s = p.substring(6);

System.out.println(s);

}

}

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

String p="abc";

int a = Integer.parseInt(p);

System.out.println(a);

}

}

======================================

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

int[] data = new int[5];

System.out.println(data[7]);

}

}

======================================

package mypackage;

public class MyClass

{

public static void main(String[] args)

{

int p=6;

int q=0;

int r=p/q;

System.out.println(r);

}

}

======================================

================================

CLASS-7 NOTES

================================

LINEAR SEARCH

==============================================

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

// TODO Auto-generated method stub

List<Integer> data = Arrays.asList(78,56,92,43,77,99,12,14,17,24,45,59,94);

int searchElement,loc=0;

Scanner obj = new Scanner(System.in);

System.out.println("Enter search element:");

searchElement = obj.nextInt();

//foreach loop

for(int d:data)

{

if(d==searchElement)

{

System.out.println("Element found at "+loc);

break;

}

loc++;

}

if(loc==data.size())

System.out.println("Element not found");

}

}

=================================

LINEAR SEARCH - SECOND WAY

=======================================

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

// TODO Auto-generated method stub

List<Integer> data = Arrays.asList(78,56,92,43,77);

int searchElement,i;

Scanner obj = new Scanner(System.in);

System.out.println("Enter search element:");

searchElement = obj.nextInt();

for(i=0;i<data.size();i++)

{

if(data.get(i)==searchElement)

break;

}

if(i==data.size())

System.out.println("Element not found");

else

System.out.println("Element found at location : "+i);

}

}

=====================================================

=================================

BINARY SEARCH - 90% done code

========================================

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

// TODO Auto-generated method stub

List<Integer> data = Arrays.asList(12,14,17,24,43,45,56,59,77,78,92,94,99);

int searchElement=23;

int low, mid, high;

low=0;

high=data.size();

mid=(low+high)/2;

while(true)

{

System.out.println("Mid : "+mid);

if(searchElement==data.get(mid))

{

System.out.println("Found at : "+mid);

break;

}

if(searchElement>data.get(mid))

low=mid;

else

high=mid;

//Take new mid now

mid=(low+high)/2;

}

}

}

==============================================================

BINARY SEARCH - 100% done

==============================================================

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class MyClass

{

public static void main(String[] args)

{

// TODO Auto-generated method stub

List<Integer> data = Arrays.asList(12,14,17,24,43,45,56,59,77,78,92,94,99);

int searchElement=11;

int low, mid, high;

low=0;

high=data.size();

mid=(low+high)/2;

while(true)

{

//System.out.printf("Low : %d, High : %d, Mid : %d",low,high,mid);

System.out.println("Low : "+low+",High:"+high+",mid:"+mid);

if(searchElement==data.get(mid))

{

System.out.println("Found at : "+mid);

break;

}

if(mid==low || mid==high)

{

System.out.println("Element not found");

break;

}

if(searchElement>data.get(mid))

low=mid;

else

high=mid;

//Take new mid now

mid=(low+high)/2;

}

}

}

============================================================

SELECTION SORT

=============================================================

public class MyClass

{

public static void main(String[] args)

{

int[] data = new int[] {71,67,89,92,54};

int index,i,j;

for(i=0;i<data.length;i++)

{

index=i;

for(j=i+1;j<data.length;j++)

{

if(data[j]<data[index])

index=j;

}

int smallnumber = data[index];

data[index]=data[i];

data[i]=smallnumber;

}

for(i=0;i<data.length;i++)

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

}

}

=============================================================

JAVA DAY-8 NOTES [ 1-Aug-2021]

============================================================

PRINT ELEMENTS OF LIST USING FOREACH METHOD

============================================================

package mypackage;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

class Product

{

int id;

String name;

Product(int id,String name)

{

this.id=id;

this.name=name;

}

}

public class MyClass

{

public static void main(String[] args)

{

/\*

List<Integer> data = Arrays.asList(56,78,92,77,99);

//for

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

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

//hello

for(int d:data)

System.out.println(d);

//ForEach method with lambda expression ->

data.forEach(d->System.out.println(d));

\*/

//Declaring list of products

List<Product> products = new ArrayList<Product>();

//adding products

products.add(new Product(1,"TV"));

products.add(new Product(2,"Mobile"));

products.add(new Product(3,"Laptop"));

//For loop

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

{

System.out.println(products.get(i).name);

}

//foreach loop

for(Product p:products)

{

System.out.println(p.name);

}

//ForEach method and lambda

products.forEach(p->System.out.println(p.name));

}

}

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

JAVA-8 Interface concept

===========================================================

package mypackage;

interface ISalary

{

static void PrintCompany()

{

System.out.println("Microsoft");

}

default String DeveloperName()

{

return "Meganadh";

}

int getHRA(int basic);

int getBonus();

}

class Microsoft implements ISalary

{

@Override

public int getHRA(int basic) {

// TODO Auto-generated method stub

return 0;

}

@Override

public int getBonus() {

// TODO Auto-generated method stub

return 0;

}

}

public class MyClass

{

public static void main(String[] args)

{

int a=23\_5\_893;

System.out.println(a);

}

}

=================================================================

Most important and we understood

=================================================================

Implementing Linked List

=================================================================

package mypackage;

public class LinkedList

{

Node head;

static class Node

{

int data;

Node next;

Node(int d)

{

data = d;

next = null;

}

}

public static LinkedList insert(LinkedList list, int data)

{

//Create a new node with the data they are giving

Node new\_node=new Node(data);

//There are no elements in the list

if(list.head==null)

{

list.head=new\_node;

}

else

{

//travel through all elements till you get next as null

Node last=list.head;

while(last.next!=null)

{

last=last.next;

}

last.next=new\_node;

}

return list;

}

public static void printList(LinkedList list)

{

Node currNode=list.head;

System.out.println("Linked list is below:");

while(currNode!=null)

{

System.out.println(currNode.data);

currNode=currNode.next;

}

}

public static void main(String[] args)

{

// TODO Auto-generated method stub

LinkedList mydata = new LinkedList();

//ADD 4 VALUES FOR THE LIST

mydata=insert(mydata,45);

mydata=insert(mydata,75);

mydata=insert(mydata,98);

mydata=insert(mydata,88);

//PRINT THE LIST VALUES

printList(mydata);

}

}

==================================================

IMPLEMENTING STACK

==================================================

package mypackage;

public class Stack

{

static final int MAX = 10;

int top;

int a[] = new int[MAX];

Stack()

{

top=-1;

}

boolean isEmpty()

{

return (top<0);

}

boolean push(int data)

{

if(top>=(MAX-1))

{

System.out.println("Stack Overflow");

return false;

}

else

{

top=top+1;

a[top]=data;

System.out.println("Data added to stack");

return true;

}

}

int pop()

{

if(top==-1)

{

System.out.println("Stack is empty");

return 0;

}

else

{

int x=a[top--];

return x;

}

}

int peek()

{

if(top==-1)

{

System.out.println("Stack is empty");

return 0;

}

else

{

int x=a[top];

return x;

}

}

public static void main(String[] args)

{

Stack s=new Stack();

s.push(10);

s.push(25);

s.push(99);

System.out.println(s.peek());

System.out.println(s.pop());

System.out.println(s.pop());

System.out.println(s.pop());

System.out.println(s.pop());

}

}

====================================================

PriorityQueue

======================================================

package mypackage;

import java.util.PriorityQueue;

import java.util.Queue;

public class MyClass

{

public static void main(String[] args)

{

// TODO Auto-generated method stub

Queue<String> mq = new PriorityQueue<String>();

mq.add("Meg");//2

mq.add("Raj");//3

mq.add("Kiran");//1

mq.add("Shankar");//4

System.out.println(mq.poll());

}

}

==================================================================