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

{

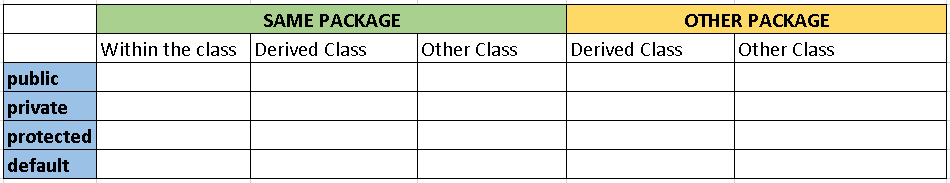
public static void main(String[] args)

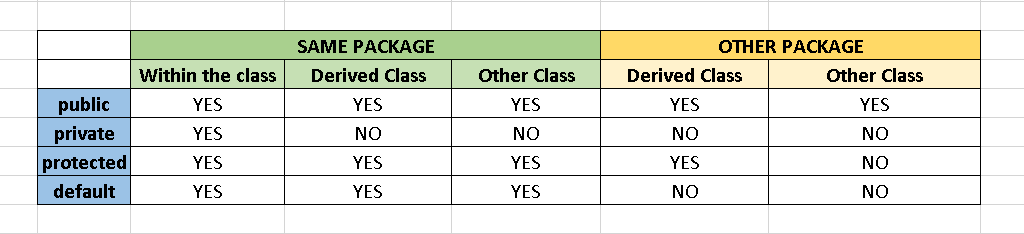
{

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

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