Notes: First you prepare how to write a code for syntax and write a code

Java: (day one)

install java

install eclipse

create workspace

create project

File -> Project

We call Project is program

create .java file/class

Project -> new class and give extension .java

Class Employee {

}

how to create packages and what is best way to give name

From solution explorer, select project, right click and select package

Ex: companyname.projectname.foldername (this is common naming standard)

what is main method will do?

Main method is starting point of program

what is data type and different data types

It will represents what type of

Int

Double

Float

creating property/data members

int salary

creating method with void

package selenium;

public class Methodwithvoid {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Learn Core java");

}

}

creating method with void and parameter

public class VoidwithParameter {

public static void main(String[] args) {

methodRankingMarks(255.7);

}

public static void methodRankingMarks(double marks) {

if (marks >= 202.5) {

System.out.println("Rank:A1");

}

else if (marks >= 122.4) {

System.out.println("Rank:A2");

}

}

}

creating method with return data type and parameter

public static int minFunction(int n1, int n2) {

int min;

if (n1 > n2)

min = n2;

else

min = n1;

return min;

}

creating variable

package selenium;

public class Methodwithvoid {

int a=4;

int b=7;

int c=a+b;

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("c");

}

}

creating static property

creating static method

creating object

package selenium;

public class Methodwithvoid {

Methodwithvoid obj=new Methodwithvoid();

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Learn Core java");

}

}

calling method with no return

calling method with no return and parameter

calling method with return and no parameter

calling method with return and parameter

calling method with return and storing the return data

package com.basics.corejava;

public class BasicsOOPS {

private int a, b;

public int getA() {

return a;

}

public void setA(int a) {

this.a = a;

}

public int getB() {

return b;

}

public void setB(int b) {

this.b = b;

}

private static String s;

// creating method with void

public void display() {

System.out.println("Hello World!! " + "\n" + "This is method with void return type");

}

// creating method with void and parameters

public void sum(int a, int b) {

int c = a + b;

System.out.println("result =" + c + "This is the method with void and parameters");

}

// creating method with return datatype

public boolean displayValue() {

System.out.println("This is the method with return datatype");

return false;

}

// creating method with return datatypes and parameters

public int addition(int a, int b) {

int result = a + b;

System.out.println("Hello you are calling sum method with  return type and has  input parameters!!");

System.out.println("" + result);

return result;

}

// creating variable

public int sumWithVariable() {

int count = 0;// count is a variable with method scope

System.out.println("Hello you are calling method with  return type and has  input parameters!!");

for (int i = 0; i < 3; i++) {

count = count + 1;

System.out.println(" value of count = " + count);

}

System.out.println(" value of count = " + count);

BasicsOOPS obj = new BasicsOOPS();

return count;

}

//static method creation

public static void message() {

System.out.println("This is a static method!!");

}

//method with return type and no parameters

// calling static property and static method

public String displayName() {

BasicsOOPS.message();// calling static method

BasicsOOPS.s = "Tejaswini Vodapally";

return BasicsOOPS.s ;// calling static property

}

public static void main(String[] args) {

BasicsOOPS obj = new BasicsOOPS();

        obj.display();

        obj.sum(4, 6);

        obj.addition(63, 33);

        obj.displayValue();

        int result = obj.sumWithVariable();

}

}

calling static method

class Calculate{

static int cube(int x){

return x\*x\*x;

}

public static void main(String args[]){

int result=Calculate.cube(5);

System.out.println(result);

}

}

using static property

The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class.

Java: Day2

create classes under multiple packages

class Computer {

Computer() {

System.out.println("Constructor of Computer class.");

}

void computer\_method() {

System.out.println("Power gone! Shut down your PC soon...");

}

public static void main(String[] args) {

Computer my = new Computer();

Laptop your = new Laptop();

my.computer\_method();

your.laptop\_method();

}

}

class Laptop {

Laptop() {

System.out.println("Constructor of Laptop class.");

}

void laptop\_method() {

System.out.println("99% Battery available.");

}

}

write code to handle exceptions with try/catch/finally:

class TestFinallyBlock{

public static void main(String args[]){

try{

int data=25/5;

System.out.println(data);

}

catch(NullPointerException e){System.out.println(e);}

finally{System.out.println("finally block is always executed");}

System.out.println("rest of the code...");

}

}

what is final keyword:

we use final keyword to variables to make its values not to be changed

write code for interface and create class to implement that interface

class Employee{

float salary=40000;

}

class Programmer extends Employee{

int bonus=10000;

public static void main(String args[]){

Programmer p=new Programmer();

System.out.println("Programmer salary is:"+p.salary);

System.out.println("Bonus of Programmer is:"+p.bonus);

}

}

write code for creating abstract class

abstract class Bike{

abstract void run();

}

class Honda4 extends Bike{

void run(){System.out.println("running safely..");}

public static void main(String args[]){

Bike obj = new Honda4();

obj.run();

}

}

implement method overloading:

class Calculate{

static int cube(int x){

return x\*x\*x;

}

public static void main(String args[]){

int result=Calculate.cube(5);

System.out.println(result);

}

}

implement method overriding

class Vehicle{

void run(){System.out.println("Vehicle is running");}

}

class Bike extends Vehicle{

public static void main(String args[]){

Bike obj = new Bike();

obj.run();

}

}

implementing polymorphism:

implementing interface:

interface printable{

void print();

}

class A6 implements printable{

public void print(){System.out.println("Hello");}

public static void main(String args[]){

A6 obj = new A6();

obj.print();

}

}

write a code to save data into excel file and read from excel file (POI and jexcel API)

how to update the data into XML file and read data from XML file

Java: Day3

write code to add items to integer, string array:

package selenium;

public class ArrayInt {

public static void main(String[] args) {

// TODO Auto-generated method stub

int arr[] = new int [7];

for(int i=0;i<7;i++){

arr[i]= i+3;

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

}

}

}

write code to retrieve items from integer, string array

public class Program {

public static void main(String[] args) {

String[] elements = { "cat", "dog", "mouse" };

System.out.println(elements.length);

System.out.println(elements[0]);

for (String element : elements) {

System.out.println(element);

}

String[] elements2 = new String[] { "bird", "fish", "cow" };

System.out.println(elements2.length);

for (int i = 0; i < elements2.length; i++) {

System.out.println(elements2[i]);

}

}

}

write code to add items to ArrayList collection:

import java.util.\*;

public class ArrayListExample {

public static void main(String args[]) {

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

obj.add("Ajeet");

obj.add("Harry");

obj.add("Chaitanya");

obj.add("Steve");

obj.add("Anuj");

System.out.println("Currently the array list has following elements:"+obj);

obj.add(0, "Rahul");

obj.add(1, "Justin");

obj.remove("Chaitanya");

obj.remove("Harry");

System.out.println("Current array list is:"+obj);

obj.remove(1);

System.out.println("Current array list is:"+obj);

}

}

write code to retrieve items from arraylist:

public class Arrayltol {

public static void main(String a[]){

ArrayList myar=new ArrayList();

myar.add("dsgfd");

myar.add("sds");

myar.add("sf");

myar.add(0,0);

System.out.println(myar);

List list=new ArrayList();

list.add("zsfs778549748");

list.addAll(myar);

System.out.println(list);

}

}

write code to add items HashMap

public class Player {

public Player() {

}

public void setScore(int score) {

this.score = score;

}

public void setName(String name) {

this.name = name;

}

private String name;

private int score;

public Player(String name, int score) {

this.name = name;

this.score = score;

}

public String getName() {

return name;

}

@Override

public String toString() {

return "Player{" + "name=" + name + "score=" + score + '}';

}

public int getScore() {

return score;

}

write code to retrieve items HashMap:

ublic static void main(String[] args) {

HashMap<String, String> hashmap = new HashMap<String, String>();

hashmap.put("one", "1");

hashmap.put("two", "2");

hashmap.put("three", "3");

hashmap.put("four", "4");

hashmap.put("five", "5");

hashmap.put("six", "6");

Iterator<String> keyIterator = hashmap.keySet().iterator();

Iterator<String> valueIterator = hashmap.values().iterator();

while (keyIterator.hasNext()) {

System.out.println("key: " + keyIterator.next());

}

while (valueIterator.hasNext()) {

System.out.println("value: " + valueIterator.next());

}

}

write code to connect to JDBC to get rows from employee table

package selenium;

import java.sql.\*;

import oracle.jdbc.driver.\*;

import oracle.sql.\*;

public class JBBCConnection

{

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection conn = DriverManager.getConnection("jdbc:oracle:oci8:@oracle.world", "oratest", "oratest");

Statement sql\_stmt = conn.createStatement();

ResultSet rset = sql\_stmt.executeQuery("SELECT empno, ename, sal, deptno FROM emp ORDER BY ename");

String sql = "SELECT empno, ename, sal, deptno FROM emp ORDER BY ename";

ResultSet rset = sql\_stmt.executeQuery(sql);

}

create Employee class

Add employee class to list collection

create method that return list of employee collection