Solution 29:

package com.hsbc.shapes; //shapes package created

class Square{

public void display(){

System.out.println(" Object of class Square ");

}

}

class Triangle{

public void display() {

System.out.println(" Object of class Triangle ");

}

}

class Circle{

public void display() {

System.out.println(" Object of class Circle ");

}

}

public class TestShapes { //main method

public static void main(String[] args) {

Square s = new Square();

Triangle t = new Triangle();

Circle c = new Circle();

s.display();

t.display();

c.display();

}

}

Solution 30:

package com.hsbc.pack;

public class Lion { // animal 1 class

String color;

int weight,age;

public Lion(String color, int weight, int age) { //parameterized constructor

super();

System.out.println("This is a lion");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Lion is not vegetarian");

}

public void canClimb() {

System.out.println("Lions cannot climb trees");

}

public void sound() {

System.out.println("Lion roars");

}

}

File 2: Tiger.java

package com.hsbc.pack.animals;

public class Tiger {

String color;

int weight, age;

public Tiger(String color, int weight, int age) { //parameterized constructor

super();

System.out.println("This is a tiger");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Tiger is not vegetarian");

}

public void canClimb() {

System.out.println("Tiger can climb trees");

}

public void sound() {

System.out.println("Tigers growls");

}

}

File 3 : Deer.java

package com.hsbc.pack.animals;

public class Deer {

String color;

int weight, age;

public Deer(String color, int weight, int age) { //parameterized constructor

super();

System.out.println("This is a deer");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Deers are vegetarians");

}

public void canClimb() {

System.out.println("Deers cannot climb trees");

}

public void sound() {

System.out.println("Deer grunts");

}

}

File 4 : VandalurZoo.java

// driver class

package com.hsbc.zoo;

import com.hsbc.org.pack.Deer;

import com.hsbc.org.pack.Lion;

import com.hsbc.org.pack.Tiger;

public static void main(String[] args) {

Lion lion =new Lion("ABC",140,19);

System.out.print("Color Information : ");

lion.getColor();

System.out.print("Age Information : ");

lion.getAge();

System.out.print("Weight Information : ");

lion.getWeight();

System.out.print("Capability to climb : ");

lion.canClimb();

System.out.print("Vegetarian or not : ");

lion.isVegetarian();

System.out.print("Animal sound : ");

lion.sound();

Tiger tiger =new Tiger("Orange",150,19);

System.out.print("Color Information : ");

tiger.getColor();

System.out.print("Age Information : ");

tiger.getAge();

System.out.print("Weight Information : ");

tiger.getWeight();

System.out.print("Capability to climb : ");

tiger.canClimb();

System.out.print("Vegetarian or not : ");

tiger.isVegetarian();

System.out.print("Animal sound : ");

tiger.sound();

Deer deer =new Deer("Ochre",40,22);

System.out.print("Color Information : ");

deer.getColor();

System.out.print("Age Information : ");

deer.getAge();

System.out.print("Weight Information : ");

deer.getWeight();

System.out.print("Capability to climb : ");

deer.canClimb();

System.out.print("Vegetarian or not : ");

deer.isVegetarian();

System.out.print("Animal sound : ");

deer.sound();

}

}

Solution 31:

package com.hsbc.pack;

import java.util.Properties;

public class JVM {

public static void main(String[] args) {

Properties property= System.getProperties();

property.list(System.out); //Listing the JVM properties

}

}

Solution 32:

// taking input from scanner class

package com.hsbc.pack;

import java.util.Scanner;

public class Student {

String name;

String degree;

int age;

float totalmarks;

float percentage;

public Student(String name, String degree, int age, float totalmarks, float percentage) {

super();

this.name = name;

this.degree = degree;

this.age = age;

this.totalmarks = totalmarks;

this.percentage = percentage;

}

public Student() {

super();

}

public String toString() {

return "Student has folloeing data\*\*\*\*\*name=" + name + ", degree=" + degree + ", age=" + age + ", totalmarks=" + totalmarks+ ", percentage=" + percentage + "";

}

public static void main(String[] args) {

Student stud=new Student();

Scanner sc=new Scanner(System.in); // creating scanner class object

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

stud.name=sc.next(); // input using scanner class object

System.out.println("Enter degree");

stud.degree=sc.next();

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

stud.age=sc.nextInt();

System.out.println("Enter total marks");

stud.totalmarks=sc.nextFloat();

System.out.println("Enter percentage");

stud.percentage=sc.nextFloat();

stud=new Student(stud.name,stud.degree,stud.age,stud.totalmarks,stud.percentage);

System.out.println(stud.toString());

}

}

Solution 33:

package com.hsbc.pack;

import java.io.FileDescriptor;

import java.io.FileOutputStream;

import java.io.PrintStream;

public class House {

public static void main(String[] args) {

Hall h = new Hall();

System.out.println();

Kitchen k = new Kitchen();

}

}

class Hall{

Hall(){

PrintStream myout = new PrintStream(new FileOutputStream(FileDescriptor.out));

myout.print("This is the first room while entering the house");

}

}

class Kitchen{

String[] appliance = {"Mixer","Oven","Gas","OTG","Toaster"};

Kitchen(){

System.out.println("The appliances found the kitchen are as follows : ");

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

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

}

String[] copy = appliance;

System.gc(); //Invoking the garbage collector explicitly

}

}

Solution 50:

package com.hsbc.pack;

import java.util.InputMismatchException;

import java.util.Scanner;

public class Calculator {

int num1,num2,res;

// functions of operands

void add(int num1,int num2)

{

res=num1+num2;

System.out.println("Result:" +num1 +"+"+num2+ "=" +res);

}

void sub(int num1,int num2)

{

res=num1-num2;

System.out.println("Result:" +num1 +"-"+num2+ "=" +res);

}

void mult(int num1,int num2)

{

res=num1\*num2;

System.out.println("Result:" +num1 +"\*"+num2+ "=" +res);

}

void div(int num1,int num2)

{

try

{

res=num1/num2;

}

catch(ArithmeticException e )

{

System.out.print("Arithmetic Exception at division");

}

finally

{

System.out.println("Result:" +num1 +"/"+num2+ "=" +res);

}

}

public static void main(String[] args) {

int firstNumber,secondNumber;

char ch;

Calculator c=new Calculator();

Scanner sc=new Scanner(System.in);

try

{

System.out.println("Enter your first number: ");

firstNumber = sc.nextInt();

System.out.println("Enter your second number: ");

secondNumber = sc.nextInt();

System.out.println("Enter the operation");

ch=sc.next().charAt(0);

switch(ch) //input going through switch

{

case '+' : c.add(firstNumber, secondNumber);

break;

case '-' : c.sub(firstNumber, secondNumber);

break;

case '\*' : c.mult(firstNumber, secondNumber);

break;

case '/' : c.div(firstNumber, secondNumber);

break;

default: System.out.println("Not valid");

}

}

catch (InputMismatchException e) // exception raised

{

System.out.println("Wrong input");

}

}

}

Solution 51:

package com.hsbc.pack;

public class StringPractice

{

public static void main(String[] args) {

try{

String s="The quick brown fox jumps over the lazy dog";

System.out.println("Character at 12th index "+ s.charAt(11));

String word="";

boolean flag=false;

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("is"))

{

System.out.println("It contains is");

flag=true;

break;

}

else

{

word="";

}

}

}

if (flag==false)

System.out.println("It does not contain is");

String a=s+" and killed it"; //adding 'and killed it' to existing string

System.out.println("After appending: "+a);

word=""; //checking if string ends with dogs

for(int i=s.length()-1;i<s.length();i--)

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=ch+word;

}

else

{

if(word.equals("dog"))

{

System.out.println("Ends with 'dog'");

break;

}

else

{

System.out.println("Does not end with 'dog'");

}

}

}

// Checking whether the String is equal to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Checking whether the String is “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

if(s.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Finding the index position of the character a

System.out.println(s.indexOf("a"));

//Finding the last index position of the character “e”

System.out.println(s.lastIndexOf("e"));

//Finding the length of the String.

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

//Checking whether the String matches to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Replacing the word “The” with the word “A”

word="";

String the="";

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("The")||word.equals("the"))

{

the=the+" A";

word="";

}

else

{

the=the+" "+word;

word="";

}

}

}

System.out.println("After replacing "+ the);

String d[]=s.split(" ",s.length());

System.out.println("After splitting");

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

{

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

}

the=s+" ";

word="";

System.out.println("Animal Names");

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

{

char ch=t.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("fox")||word.equals("dog"))

{

System.out.println(word);

word="";

}

else

{

word="";

}

}

}

//Printing the above string in completely lower case

System.out.println("String in lowercase: "+s.toLowerCase());

//Printing the above string in completely upper case

System.out.println("String in uppercase: "+s.toUpperCase());

}

catch(NullPointerException e) //Null pointer Exception

{

e.printStackTrace();

}

}

}

Solution 52:

package com.hsbc.pack;

public class StringPractice

{

public static void main(String[] args) {

try{

String s="The quick brown fox jumps over the lazy dog";

System.out.println("Character at 12th index "+ s.charAt(11));

String word="";

boolean flag=false;

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("is"))

{

System.out.println("It contains is");

flag=true;

break;

}

else

{

word="";

}

}

}

if (flag==false)

System.out.println("It does not contain is");

String a=s+" and killed it"; //adding 'and killed it' to existing string

System.out.println("After appending: "+a);

word=""; //checking if string ends with dogs

for(int i=s.length()-1;i<s.length();i--)

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=ch+word;

}

else

{

if(word.equals("dog"))

{

System.out.println("Ends with 'dog'");

break;

}

else

{

System.out.println("Does not end with 'dog'");

}

}

}

// Checking whether the String is equal to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Checking whether the String is “THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG”

if(s.compareTo("THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Finding the index position of the character a

System.out.println(s.indexOf("a"));

//Finding the last index position of the character “e”

System.out.println(s.lastIndexOf("e"));

//Finding the length of the String.

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

//Checking whether the String matches to “The quick brown Fox jumps over the lazy Dog”

if(s.compareTo("The quick brown Fox jumps over the lazy Dog")==0)

System.out.println("Equal");

else

System.out.println("Not Equal");

//Replacing the word “The” with the word “A”

word="";

String the="";

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

{

char ch=s.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("The")||word.equals("the"))

{

the=the+" A";

word="";

}

else

{

the=the+" "+word;

word="";

}

}

}

System.out.println("After replacing "+ the);

String d[]=s.split(" ",s.length());

System.out.println("After splitting");

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

{

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

}

the=s+" ";

word="";

System.out.println("Animal Names");

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

{

char ch=t.charAt(i);

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("fox")||word.equals("dog"))

{

System.out.println(word);

word="";

}

else

{

word="";

}

}

}

//Printing the above string in completely lower case

System.out.println("String in lowercase: "+s.toLowerCase());

//Printing the above string in completely upper case

System.out.println("String in uppercase: "+s.toUpperCase());

}

catch(StringIndexOutOfBoundsException e) //Exception 1

{

e.printStackTrace();

}

catch(Exception e) //General Exception

{

e.printStackTrace();

}

}

}

Solution 53:

package com.hsbc.pack;

import java.util.Scanner;

class SquareMatrix  
{  
  public static void main(String args[])  
  {

try{   
    int row, column, c, d;  
    Scanner in = new Scanner(System.in);

    System.out.println("Enter the number of rows and columns of matrix");  
    row = in.nextInt();  
    column = in.nextInt();

    int first[][] = new int[row][column]; //creating matrix  
    int square[][] = new int[row][column];

    System.out.println("Enter the elements of first matrix"); //input of matrix

    for (c = 0; c < row; c++)  
      for (d = 0; d < column; d++)  
        first[c][d] = in.nextInt();

    for (c = 0; c < row; c++)  
      for (d = 0; d < column; d++)  
        square[c][d] = square[c][d] \* square[c][d]; //squaring each element through loop

    System.out.println("Square of the matrices:");

    for (c = 0; c < row; c++)  
    {  
      for (d = 0; d < column; d++)  
        System.out.print(square[c][d] + "\t");

      System.out.println();  
    }  
  }

}

}

catch(ArrayIndexOutOfBoundsException e)

{

e.printStackTrace();

}

}

}

Solution 54:

package com.hsbc.pack;

public class AllException {

public static void main(String[] args) {

try {

Scanner sc = new Scanner(System.in);

int l = sc.nextInt();

int[] arr = new int[l];

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

arr[i] = sc.nextInt();

}

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

String n = "Aman Saraff";

System.out.println(n.charAt(6));

String s = null;

s.toString();

int a = 6,b =0;

int c = a/b;

System.out.println(c);

}

catch(NegativeArraySizeException e) {

e.printStackTrace();

}

catch(ArrayIndexOutOfBoundsException e) { //IndexOutOfBoundsException

e.printStackTrace();

}

catch(StringIndexOutOfBoundsException e) { //String out of bounds exception

e.printStackTrace();

}

catch(NullPointerException e) { // null pointer exception

e.printStackTrace();

}

catch(ArithmeticException e) { // arithmetic exception

e.printStackTrace();

}

}

}

Solution 55:

package com.hsbc.pack;

import java.util.Scanner;

import bankImpl.Savings;

public class Bank {

public static void main(String[] args) {

Savings savings = new Savings();

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

Scanner scnr = new Scanner(System.in);

savings.create(scnr.next());

savings.withdraw(14000);

}

}

Savings.java

package bankImp;

import bank.IDeposit;

import bank.MyException;

public class Savings implements IDeposit {

public static void main(String[] args) {

}

public void create(String name) {

System.out.println("Savings Account -> Create Account");

if(name.length() < 3) {

try {

throw new MyException(); // Name Exception

} catch (MyException me) {

me.nameException();

}

}

}

public void withdraw(double amt) {

if(amt > getBalance()) // condition if amt is more than balance

{

try {

throw new MyException();

}

catch (MyException me) {

me.withdrawalException();

}

}

}

public void deposit(double amt) {

}

public double getBalance() {

return 50000.00;

}

}

IDeposit.java

package bank;

public interface IDeposit extends IAccount {

void withdraw(double amt);

void deposit(double amt);

double getBalance();

}

MyException.java

package bank;

// custom exception

public class MyException extends Exception {

public void nameException() {

System.out.println("Name invalid/ very short");

}

public void withdrawalException() {

System.out.println("Withdrawal Amount is greater than Available Balance");

}

public void getBalanceException() {

System.out.println("Account number is not there");

}

}

Solution 56:

package com.hsbc.pack;

public class ToNull {

int num1;

int num2;

public ToNull() {

super();

}

public ToNull(int num1, int num2) {

super();

this.num1 = num1;

this.num2 = num2;

}

@Override

public String toString() {

return "ToNull [num1=" + num1 + ", num2=" + num2 + "]";

}

void div(int num1,int num2)

{

int res;

try

{

res=num1/num2;

System.out.println("Result:" +res);

}

catch(ArithmeticException e)

{

System.out.println("Arithmetic Exception at divison");

}

}

void add(int num1,int num2)

{

int res;

res=num1+num2;

System.out.println("Result:" +res);

}

void sub(int num1,int num2)

{

int res;

res=num1-num2;

System.out.println("Result:" +res);

}

void mult(int num1,int num2)

{

int res;

res=num1\*num2;

System.out.println("Result:" +res);

}

public static void main(String args[]) {

ToNull s=new ToNull(); //creating object

try

{

s.add(100,10);

s.sub(100,10);

s.mult(100,10);

s.div(100,10);

}

catch(ArithmeticException e)

{

System.out.println("Exception occurs");

}

finally

{

System.out.println("Setting object to null");

s=null; // setting null to object

s.toString(); // checking to see if it is null by printing

}

}

}

Solution 57:

package com.hsbc.pack;

public class UsingFinally {

public UsingFinally() {

super();

}

public void val()

{

String s = null; //string is null

try

{

System.out.println(s);

}

catch (NullPointerException e) // flow comes in this block

{

s = s + "Adding a string";

System.out.println(s);

}

Finally //finally block executes even if there is or isnt any exception

{

s = s + " returning the value";

System.out.println(s);

}

}

public static void main(String[] args) {

UsingFinally s = new UsingFinally();

s.val();

}

}

Solution 58:

package com.hsbc.pack;

import java.util.\*;

public class Employee {

static Scanner sc = new Scanner(System.in);

static ArrayList<String> names = new ArrayList<String>();

public Employee() {

super();

}

public void getName() throws MyException {

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

String name = sc.next();

if(names.contains(name)) //checking if the name entered exists already or not

throw new MyException();

else {

System.out.println("Hi " + name);

names.add(name); // adding the name if it is not there

}

}

public void getAge() throws AgeLimitException, NegativeAgeException {

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

int age = sc.nextInt();

if(age<0)

throw new NegativeAgeException();

else if(age<18 && age>60)

throw new AgeLimitException();

else

System.out.println("Your age is in range");

}

public static void main(String[] args) {

Employee e = new Employee();

try {

e.getName();

} catch (MyException e1) {

e1.printStackTrace();

}

try {

e.getAge();

} catch (AgeLimitException || NegativeAgeException e1) {

e1.printStackTrace();

}

}

}