**Rohitkaran Adusumalli**

**Packages**

**Lab Exercise No:**29

**Exercise Objective(s):***Package*

**Exercise:***Create a package called shapes. Create some classes in the package representing some*

*common geometric shapes like Square, Triangle, Circle and so on. Create a class called*

*TestShapes and create objects for all the shapes and print corresponding messages.*

*Execute the TestShapes class.*

**Recommended duration:** *20 Mins*

**Solution :**

Square.Java :

package com.hsbc.shapes;

//The sqaure class in shapes package

public class Square {

Square(){

System.out.println("This is the Square class in shapes com.hsbc.package.");

}

}

Circle.java :

package com.hsbc.shapes;

//The circle class in the same package

public class Circle {

Circle(){

System.out.println("This is the circle class in shapes com.hsbc.package.");

}

}

Triangle.java :

package com.hsbc.shapes;

//The triangle class in the same package

public class Triangle {

Triangle(){

System.out.println("This is the Triangle class in shapes com.hsbc.package.");

}

}

TestShapes.java :

package com.hsbc.shapes;

public class TestShapes {

public static void main(String[] args) {

// TODO Auto-generated method stub

//Creating the objects of various classes

Square s = new Square();

Circle c = new Circle();

Triangle t = new Triangle();

}

}

**Lab Exercise No:**30

**Exercise Objective(s):***Jar*

**Exercise:**

1. *Create a new project in which create a package named org.animals. In that create various classes like Lion, Tiger, Deer, Monkey, Elephant and Giraffe. In each class create data members like color, weight,age etc. Create methods like isVegetarian, canClimb, sound etc*
2. *Create another project and in that create a package called zoo and create a class called VandalurZooand create objects for the animals that are existing in zoo and print the characteristic of each animal.*

**Recommended duration:***15Mins*

**Solution :**

**Deer.java :**

package org.animals;

public class Deer {

int weight = 80,age = 20;

String color = "Brown";

public Deer() {

super();

}

public void displayInfo() {

System.out.println("The weight of deer is:" + weight);

System.out.println("The age of deer is:" + age);

System.out.println("The color of deer is:" + color);

}

public void isVegeterian() {

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

}

public void canClimb() {

System.out.println("Deers can't climb.");

}

}

**Monkey.java:**

package org.animals;

public class Monkey {

int weight = 90,age = 10;

String color = new String();

public Monkey() {

super();

}

public Monkey(int weight, int age, String color) {

super();

this.weight = weight;

this.age = age;

this.color = color;

}

public void displayInfo() {

System.out.println("The weight of monkey is:" + weight);

System.out.println("The age of monkey is:" + age);

System.out.println("The color of monkey is:" + color);

}

public void isVegeterian() {

System.out.println("Monkeys are vegeterian.");

}

public void canClimb() {

System.out.println("Monkeys can climb.");

}

}

**Lion.java:**

package org.animals;

public class Lion {

int weight = 100,age = 15;

String color = "Orange";

public Lion() {

super();

}

public Lion(int weight, int age, String color) {

super();

this.weight = weight;

this.age = age;

this.color = color;

}

public void isVegeterian() {

System.out.println("Lions are not vegeterian.");

}

public void displayInfo() {

System.out.println("The weight of lion is:" + weight);

System.out.println("The age of lion is:" + age);

System.out.println("The color of lion is:" + color);

}

public void canClimb() {

System.out.println("Lions can't climb.");

}

}

**Tiger.java:**

package org.animals;

public class Tiger {

int weight = 150,age = 12;

String color = "Orange";

public void displayInfo() {

System.out.println("The weight of tiger is:" + weight);

System.out.println("The age of tiger is:" + age);

System.out.println("The color of tiger is:" + color);

}

public void isVegeterian() {

System.out.println("Tigers are not vegeterian.");

}

public void canClimb() {

System.out.println("Tigers can't climb.");

}

public static void main(String[] args) {

// TODO Auto-generated method stub

}

}

**Vandalur Class :**

package zoo;

import org.animals.Deer;

import org.animals.Lion;

import org.animals.Monkey;

import org.animals.Tiger;

public class VandalurZoo {

public static void main(String[] args) {

// TODO Auto-generated method stub

//Lion instance

Lion l = new Lion();

System.out.println("The info abt Lion is as follows : ");

System.out.println();

l.displayInfo();

l.isVegeterian();

l.canClimb();

//Tiger instance

Tiger t = new Tiger();

System.out.println("The info abt tiger is as follows : ");

System.out.println();

t.displayInfo();

t.isVegeterian();

t.canClimb();

//New Deer Instance

Deer d = new Deer();

System.out.println("The info abt deer is as follows : ");

System.out.println();

d.displayInfo();

d.isVegeterian();

d.canClimb();

//New MOnkey INstance

Monkey m= new Monkey();

System.out.println("The info abt monkey is as follows : ");

System.out.println();

m.displayInfo();

m.isVegeterian();

m.canClimb();

}

}

**Lab Exercise No:**31

**Exercise Objective(s):***System class*

**Exercise:***Create a class which displays the following about the JVM.*

1. *Version of Java*
2. *Vendor for Java*
3. *Class Path*
4. *Installed home directory*
5. *OS name on which it is installed with version*

**Recommended duration:***10Mins*

**Solution :**

/\*\*

\* This is a java program to display JVM information.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day3;

import java.util.Properties;

public class ThirtyOneSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

Properties p = System.getProperties();

//Listing the JVM properties

p.list(System.out);

}

}

**Lab Exercise No:**32

**Exercise Objective(s):***Scanner class*

**Exercise:***Create a class called Student. Get the details like name, degree, age, total marks and*

*percentage from the user and display the same.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a java program to understand the scanner class.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day3;

import java.util.Scanner;

public class ThirtyTwoSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scan = new Scanner(System.in);

//Taking inputs from the user

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

String name = scan.nextLine();

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

String degree = scan.nextLine();

System.out.println("Enter the Total marks of the student : ");

int totalMarks = scan.nextInt();

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

int age = scan.nextInt();

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

double percentage = scan.nextDouble();

System.out.println();

//Printing out the details taken from the user

System.out.println("The Name of the student is : " + name);

System.out.println("The degree of the student : " + degree);

System.out.println("The Total Marks of the student : " + totalMarks);

System.out.println("The age of the student : " + age);

System.out.println("The percentage of the student : " + percentage);

scan.close();

}

}

**Lab Exercise No:**33

**Exercise Objective(s):***Systemclass,usingstaticimport*

**Exercise:***Create a Package called house. Create 2 classes namely Hall and Kitchen.*

1. *In the Hall class print the message “This is the first room while entering the house” without using the class name System explicitly in the println statement.*
2. *In the Kitchen class create an array called appliances and initialize with values and print the same.*
3. *After printing copy that array into a different array.*
4. *Invoke garbage collector explicitly for the Kitchen class.*

**Recommended duration:***20Mins*

**Solution :**

package house;

import java.io.FileDescriptor;

import java.io.FileOutputStream;

import java.io.PrintStream;

public class ThirtyThreeSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

Hall h1 = new Hall();

System.out.println();

Kitchen k1 = 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[] appliances = {"Fridge","Oven","Toaster","Cooker","Grinder"};

Kitchen(){

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

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

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

}

String[] copy = appliances;

//Invoking the garbage collector explicitly

System.gc();

}

}

**JAVA**

**Exception Handling**

**Lab Exercise No:**50

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 14, change the code such that the numbers are taken as input from the*

*user. Handle the appropriate exceptions.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a program to a class with 4 methods and overloaded methods with different inputs with

\* exception handling.

\* @author Rohitkaran

\*/

package com.hsbc.day2;

import java.util.Scanner;

public class Calculator{

//Method to add

public void add(int num1,int num2)

{

int num3=num1+num2;

System.out.println("Addition is=" + num3);

}

//Method to subtract

public void sub(int num1,int num2)

{

int num3=num1-num2;

System.out.println("Subtraction is="+ num3);

}

//Method to Multiply

public void mul(int num1,int num2)

{

int num3=num1\*num2;

System.out.println("Multiplication is="+ num3);

}

//Method to Divide

public void div(int num1,int num2)

{

int num3 = 0;

//Try and catch block to handle arithmetic exceptions

try {

num3=num1/num2;

}catch (ArithmeticException e){

System.out.println("The num2 variable is 0. Please try with a different input.");

}finally {

System.out.println("Division is="+ num3);

}

}

public static void main(String args[])

{

//Creating an object of class calculator

Calculator ob=new Calculator();

boolean cont = true;

Scanner scan = new Scanner(System.in);

while(cont) {

//Taking inputs from user

System.out.println("Enter the two number :");

int num1 = scan.nextInt();

int num2 = scan.nextInt();

ob.add(num1,num2);

ob.sub(num1,num2);

ob.mul(num1,num2);

ob.div(num1,num2);

System.out.println("Do you wish to continue? If yes the type 1 ,else 0");

int temp = scan.nextInt();

cont = (temp == 1) ? true : false;

}

scan.close();

}

}

**Lab Exercise No:**51

**Exercise Objective(s):***syntax*

**Exercise:***In the Lab Exercise 17, handle the scenarios if the String variable is not initialized.*

**Recommended duration:** *20 Mins*

**Solution** *:*

/\*\*

\* This is a program to manipulate a given string and print it in different ways

\* with exception handling (NullPointerException)

\* @author Rohitkaran

\*/

package com.hsbc.day2;

public class SeventeenSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

String s = new String();

try {

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

}catch (NullPointerException e){

System.out.println("The string has no value.");

}

//Printing the character at 12th position

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

System.out.println();

//Checking if 'is' is present in s

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("Contains 'is'");

flag=true;

break;

}

else

{

word="";

}

}

}

if (flag==false)

System.out.println("Doesn not contain 'is'");

System.out.println();

//Adding "and killed it" to the original string

String a=s+" and killed it";

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

System.out.println();

//To check if string ends with dog

word="";

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'");

}

}

}

System.out.println();

// 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");

System.out.println();

//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");

System.out.println();

//Finding the index position of the character a

System.out.println("The index position of character 'a' is :"+s.indexOf("a"));

System.out.println();

//Finding the last index position of the character e

System.out.println("The last index position of character 'e' is :"+s.lastIndexOf("e"));

System.out.println();

//Finding the length of the String.

System.out.println("The length of the string is :" + s.length());

System.out.println();

//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");

System.out.println();

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

word="";

String t="";

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"))

{

t=t+" A";

word="";

}

else

{

t=t+" "+word;

word="";

}

}

}

System.out.println("After replacing :"+ t);

System.out.println();

//Printing the animal names alone separately from the above string

t=s+" ";

word="";

System.out.println("The animals names are :");

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="";

}

}

}

System.out.println();

//Printing the above string in completely lower case

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

System.out.println();

//Printing the above string in completely upper case

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

}

}

**Lab Exercise No:**52

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 17, catch and demonstrate the required exceptions.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a program to manipulate a given string and print it in different ways

\* with exception handling (NullPointerException)

\* @author Rohitkaran

\*/

package com.hsbc.day2;

public class SeventeenSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

String s = new String();

try {

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

}catch (NullPointerException e){

System.out.println("The string has no value.");

}

//Printing the character at 12th position

try {

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

}catch (StringIndexOutOfBoundsException e) {

System.out.println("You are trying to access out of bounds index");

}

System.out.println();

//Checking if 'is' is present in s

String word="";

boolean flag=false;

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

{

char ch = 0;

try {

ch=s.charAt(i);

}catch (StringIndexOutOfBoundsException e) {

System.out.println("You are trying to access out of bounds index");

}

if(Character.isLetter(ch))

{

word=word+ch;

}

else

{

if(word.equals("is"))

{

System.out.println("Contains 'is'");

flag=true;

break;

}

else

{

word="";

}

}

}

if (flag==false)

System.out.println("Doesn not contain 'is'");

System.out.println();

//Adding "and killed it" to the original string

String a=s+" and killed it";

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

System.out.println();

//To check if string ends with dog

word="";

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

{

char ch = 0;

try {

ch=s.charAt(i);

}catch (StringIndexOutOfBoundsException e) {

System.out.println("You are trying to access out of bounds index");

}

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'");

}

}

}

System.out.println();

// 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");

System.out.println();

//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");

System.out.println();

//Finding the index position of the character a

System.out.println("The index position of character 'a' is :"+s.indexOf("a"));

System.out.println();

//Finding the last index position of the character e

System.out.println("The last index position of character 'e' is :"+s.lastIndexOf("e"));

System.out.println();

//Finding the length of the String.

System.out.println("The length of the string is :" + s.length());

System.out.println();

//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");

System.out.println();

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

word="";

String t="";

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"))

{

t=t+" A";

word="";

}

else

{

t=t+" "+word;

word="";

}

}

}

System.out.println("After replacing :"+ t);

System.out.println();

//Printing the animal names alone separately from the above string

t=s+" ";

word="";

System.out.println("The animals names are :");

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="";

}

}

}

System.out.println();

//Printing the above string in completely lower case

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

System.out.println();

//Printing the above string in completely upper case

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

}

}

**Lab Exercise No:**53

**Exercise Objective(s):***syntax*

**Exercise:***Using Lab Exercise 22, catch and demonstrate the required exceptions.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a program to display the square of the elements of a two dimensional array

\* with exception handling.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day2;

import java.util.Scanner;

public class TwentyTwo {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Enter the no. of rows and columns : ");

Scanner scan = new Scanner(System.in);

int row = scan.nextInt();

int col = scan.nextInt();

int arr[][] = new int[row][col];

int square[][] = new int[row][col];

System.out.println("Enter the value of the array : ");

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

for(int j=0;j<col;j++) {

arr[i][j] = scan.nextInt();

//Checking if array index is within bounds

try {

square[i][j] = arr[i][j] \* arr[i][j];

}catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Array index is out of bounds. PLease check again.");

}

}

}

//Displaying the square matrix

System.out.println("The final squared matrix is : ");

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

for(int j=0;j<col;j++) {

//Checking if array index is within bounds

try {

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

}catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Array index is out of bounds. PLease check again.");

}

}

System.out.println();

}

}

}

**Lab Exercise No:**54

**Exercise Objective(s):***Exception class methods*

**Exercise:***By using multiple catch blocks, write a class to demonstrate the order of the execution of the*

*catch blocks usingNegativeArraySizeException,ArrayIndexOutOfBoundsException,*

*StringIndexOutOfBoundsException, IndexOutOfBoundsException, NullPointerException,*

*ArithmeticException and print the stack trace for each exception.*

**Recommended duration:** *20 Mins*

**Solution** *:*

/\*\*

\* This is a program to display the execution of multiple catch blocks.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day4;

public class FiftyFourSol {

public static void main(String[] args) {

// TODO Auto-generated method stub

int a=10,b=0,result;

//Try block with multiple exceptions

try {

result = a/b;

int[] arr = new int[-6];

}

//Multiple catch blocks

catch(NegativeArraySizeException e1){

System.out.println("The array size is negative.");

e1.printStackTrace();

}

catch(ArrayIndexOutOfBoundsException e2) {

System.out.println("Array index is out of bounds.");

e2.printStackTrace();

}

catch(StringIndexOutOfBoundsException e3) {

System.out.println("String index is out of bounds.");

e3.printStackTrace();

}

catch(IndexOutOfBoundsException e4) {

System.out.println("Index is out of bounds.");

e4.printStackTrace();

}

catch(NullPointerException e5) {

System.out.println("The variable has null value.");

e5.printStackTrace();

}

//The program checks all the catch blocks but ends at arithmetic exception block

catch(ArithmeticException e6) {

System.out.println("This is an arithmetic exception.");

e6.printStackTrace();

}

}

}

**Lab Exercise No:**56

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that it resets the value of the objects it used to null after its usage in all*

*cases.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a program to display the use of finally block.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day4;

public class FiftySixSol {

int num1,num2;

public int mul(int num1,int num2) {

int result = num1 \* num2;

return result;

}

public double div(int num1, int num2) {

int result = num1/num2;

return result;

}

public FiftySixSol() {

super();

}

public FiftySixSol(int num1, int num2) {

super();

this.num1 = num1;

this.num2 = num2;

}

public static void main(String[] args) {

// TODO Auto-generated method stub

FiftySixSol c = new FiftySixSol();

try {

System.out.println("The multiplication is : "+ c.mul(10, 20));

System.out.println("The dividsion is : " + c.div(25, 0));

}catch(ArithmeticException e) {

System.out.println("Divide by zero error");

e.printStackTrace();

}finally {

//The object is made null

c=null;

}

}

}

**Lab Exercise No:**57

**Exercise Objective(s):***finally keyword*

**Exercise:***Create a class such that a method uses the try catch block with the return type of String.*

**Recommended duration:** *20 Mins*

**Solution :**

/\*\*

\* This is a program to create class such that a method uses the try catch block with the return type of String.

\*

\* @author Rohitkaran

\*/

package com.hsbc.day4;

public class FiftySevenSol {

public FiftySevenSol() {

// TODO Auto-generated constructor stub

super();

}

public void sample()

{

String s = null;

try

{

System.out.println(s.charAt(0));

}

//Adding string in catch block

catch (NullPointerException e)

{

s = s + " Catch block add ";

System.out.println("String has not been initialized");

}

//Finally block adds string content

finally

{

s = s + " This is final block";

System.out.println(s);

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

FiftySevenSol s = new FiftySevenSol();

s.sample();

}

}

**Lab Exercise No:**58

**Exercise Objective(s):***User-defined exceptions*

**Exercise:***Create a class called Employee which asks the user to input the name and the age of a*

*employee. Raise a custom defined exception when the user enters an employee name*

*that has been already entered and raise another exception if the age is negative or less*

*than 18 or greater than 60.*

**Recommended duration:** *20 Mins*

**Solution :**

package com.hsbc.day4;

import java.util.Scanner;

public class Employee {

int[] age = new int[100];

String[] name = new String[100];

int tempCnt=0;

Scanner scan = new Scanner(System.in);

public void getName() throws InvalidNameException{

System.out.println("Enter the names of the employee :");

boolean cont = true;

for(int i=0;i<this.name.length && cont;i++) {

this.name[i] = scan.nextLine();

for(int j=i-1;j>=0;j--) {

if (this.name[i]==this.name[j]) {

throw new InvalidNameException("Name is already present");

}

}

System.out.println("Do you want to add more names? If yes then 1 else 0");

int temp = scan.nextInt();

cont = (temp==1) ? true : false;

String temp1 = scan.nextLine();

tempCnt++;

}

}

public void getAge() throws InvalidAgeException{

boolean cont = true;

for(int i=0;i<tempCnt && cont;i++) {

System.out.println("Enter the age of " + this.name[i] + " :");

this.age[i] = scan.nextInt();

if(this.age[i]<18 || this.age[i]>60) {

throw new InvalidAgeException("Age is not valid as per given conditions.");

}

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Employee e1 = new Employee();

try {

e1.getName();

} catch (InvalidNameException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

try {

e1.getAge();

} catch (InvalidAgeException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

class InvalidNameException extends Exception

{

public InvalidNameException(String s)

{

// Call constructor of parent Exception

super(s);

}

}

class InvalidAgeException extends Exception

{

public InvalidAgeException(String s)

{

// Call constructor of parent Exception

super(s);

}

}