COMSATS UNIVERSITY ISLAMABAD, LAHORE CAMPUS



Name: Abdul Wahab

Registration No: FA22-BSE-160

Class: Object Oriented Programming

Assignment: Lab Task 09

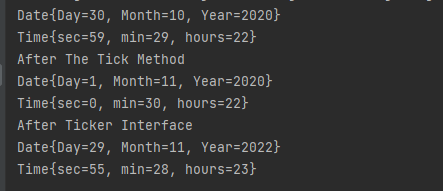
Teacher: Mam Mamoona Tassaduq

Date: 16th May 2023

Task 1:

public class Lab09\_Task1 {  
 public static void main(String[] args) {  
  
 Date date = new Date(30,10,2020);  
  
 Time time = new Time(59,29,22);  
  
 System.*out*.println(date);  
 System.*out*.println(time);  
 //Using Ticker Method  
 System.*out*.println("After The Tick Method");  
 date.tick();  
 time.tick();  
 System.*out*.println(date);  
 System.*out*.println(time);  
  
  
 //Creating Polymorphic Reference of Ticker Interface  
 System.*out*.println("After Ticker Interface");  
 Ticker dateTime = null;  
 dateTime = new Date(28,11,2022);  
 dateTime.tick();  
 System.*out*.println(dateTime);  
 dateTime = new Time(54,28,23);  
 dateTime.tick();  
 System.*out*.println(dateTime);  
  
  
  
  
  
  
 }  
}  
  
  
  
interface Ticker{  
 void tick();  
  
}  
  
  
class Date implements Ticker{  
  
 private int Day;  
 private int Month;  
 private int Year;  
  
 public Date(int day, int month, int year) {  
 Day = day;  
 Month = month;  
 Year = year;  
 }  
  
 public int getDay() {  
 return Day;  
 }  
  
 public void setDay(int day) {  
 Day = day;  
 }  
  
 public int getMonth() {  
 return Month;  
 }  
  
 public void setMonth(int month) {  
 Month = month;  
 }  
  
 public int getYear() {  
 return Year;  
 }  
  
 public void setYear(int year) {  
 Year = year;  
 }  
  
 @Override  
 public String toString() {  
 return "Date{" +  
 "Day=" + Day +  
 ", Month=" + Month +  
 ", Year=" + Year +  
 '}';  
 }  
  
 public void tick(){  
  
 Day++;  
  
 if (Day == 31) {  
 Day=1;  
  
 Month++;  
  
 if (Month==13){  
 Month=1;  
  
 Year++;  
 }  
 }  
  
 }  
  
}  
  
class Time implements Ticker{  
  
 private int sec;  
 private int min;  
 private int hours;  
  
 //Constructor  
 public Time(int sec, int min, int hours) {  
 this.sec = sec;  
 this.min = min;  
 this.hours = hours;  
 }  
  
 //Getter Setters  
  
  
 public int getSec() {  
 return sec;  
 }  
  
 public void setSec(int sec) {  
 this.sec = sec;  
 }  
  
 public int getMin() {  
 return min;  
 }  
  
 public void setMin(int min) {  
 this.min = min;  
 }  
  
 public int getHours() {  
 return hours;  
 }  
  
 public void setHours(int hours) {  
 this.hours = hours;  
 }  
  
 @Override  
 public String toString() {  
 return "Time{" +  
 "sec=" + sec +  
 ", min=" + min +  
 ", hours=" + hours +  
 '}';  
 }  
  
 public void tick(){  
  
 sec++;  
 if(sec==60){  
  
 sec = 0;  
  
 min++;  
  
  
 if (min==60){  
  
 min = 0;  
  
 hours++;  
  
 if(hours==24){  
  
 hours=0;  
 }  
 }  
 }  
  
  
  
 }  
  
  
  
  
}

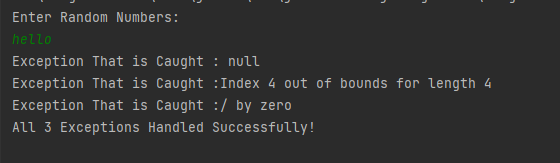
Output:



Task 3:

//Exception Handling  
  
import java.util.InputMismatchException;  
import java.util.Scanner;  
  
public class Lab09\_Task3 {  
  
 public static void main(String[] args) {  
  
 //InputMisMatch Exception :  
  
 try {  
 Scanner scn = new Scanner(System.*in*);  
 System.*out*.println("Enter Random Numbers: ");  
 int random = scn.nextInt();  
 System.*out*.println("The Given Input is : "+random);  
  
 }catch(InputMismatchException i){  
  
 System.*out*.println("Exception That is Caught : "+i.getMessage());  
  
 }  
  
 //Index-Out-Of-Bound Exception  
  
 try{  
 //Taking Array Here  
  
 int[] array = {1,2,3,4};  
 System.*out*.println(array[4]); //The Size of Array is from 0-3  
  
 }catch(IndexOutOfBoundsException i){  
  
 System.*out*.println("Exception That is Caught :"+i.getMessage());  
  
 }  
  
 //Arithmetic Exception  
  
 try{  
 //Dividing By Zero to Get Exception  
 double number = 100/0;  
 System.*out*.println("The Answer is: "+number);  
  
 }catch(ArithmeticException i){  
 System.*out*.println("Exception That is Caught :"+i.getMessage());  
  
 }  
  
 finally {  
 System.*out*.println("All 3 Exceptions Handled Successfully! ");  
 }  
  
  
  
 }  
  
  
  
}

Output:



Task 2:

import java.util.ArrayList;  
import java.util.Collections;  
  
public class Lab09\_Task2 {  
  
  
  
  
  
  
  
  
}  
  
class interface Stats{  
 double ComputeArea();  
 void reset();  
 String *units* = "Sq.m";  
  
}  
  
class Country implements Stats{  
  
 private String Name;  
  
 //Creating ArrayList Here  
 ArrayList<Integer> AreaOfProvince = new ArrayList<>(5);  
 ArrayList<Integer> AreaOfState = new ArrayList<>(4);  
  
 //Constructor of Class Country  
 public Country() {  
 AreaOfState.add(5);  
 AreaOfState.add(2);  
 AreaOfState.add(3);  
 AreaOfState.add(4);  
 }  
  
 public double ComputeArea(){  
  
 double Sum = 0;  
  
 //Using Enhanced For Loop  
  
 for(int a : AreaOfProvince)  
 {  
 Sum = Sum+a;  
  
 }  
 for(int a : AreaOfState)  
 {  
 Sum = Sum+a;  
  
 }  
  
 return Sum;  
  
 }  
  
 //Using Collection , we get  
  
 public void reset(){  
  
 Collections.*fill*(AreaOfProvince,0);  
 Collections.*fill*(AreaOfState,0);  
 }  
  
}  
  
  
class Shape {  
  
  
 private String Color = "red";  
 private boolean Filled= true;  
  
 public Shape() {  
 }  
  
 public Shape(String color, boolean filled) {  
 Color = color;  
 Filled = filled;  
 }  
 //Setter Getter  
  
  
 public String getColor() {  
 return Color;  
 }  
  
 public void setColor(String color) {  
 Color = color;  
 }  
  
 public boolean isFilled() {  
 return Filled;  
 }  
  
 public void setFilled(boolean filled) {  
 Filled = filled;  
 }  
  
 @Override  
 public String toString() {  
 return "Shape{" +  
 "Color='" + Color + '\'' +  
 ", Filled=" + Filled +  
 '}';  
 }  
}  
  
class Circle extends Shape{  
  
 private double radius = 1.0;  
  
 public Circle() {  
 }  
  
 public Circle(double radius) {  
 this.radius = radius;  
 }  
  
 public Circle(String color, boolean filled, double radius) {  
 super(color, filled);  
 this.radius = radius;  
 }  
  
 //Getter Setters  
  
  
 public double getRadius() {  
 return radius;  
 }  
  
 public void setRadius(double radius) {  
 this.radius = radius;  
 }  
  
 //Methods  
  
 public double Area(){  
 return 0;  
 }  
 public double Perimeter(){  
  
 return 0;  
 }  
  
 @Override  
 public String toString() {  
 return "Circle{" +  
 "radius=" + radius +  
 '}';  
 }  
}  
  
class Rectangle extends Shape{  
  
 private double width = 1.0;  
 private double length = 1.0;  
  
 //Constructors  
  
  
 public Rectangle() {  
 }  
  
 public Rectangle(double width, double length) {  
 this.width = width;  
 this.length = length;  
 }  
  
 public Rectangle(String color, boolean filled, double width, double length) {  
 super(color, filled);  
 this.width = width;  
 this.length = length;  
 }  
  
 //Getter Setter  
 public double getWidth() {  
 return width;  
 }  
  
 public void setWidth(double width) {  
 this.width = width;  
 }  
  
 public double getLength() {  
 return length;  
 }  
  
 public void setLength(double length) {  
 this.length = length;  
 }  
 public double ComputeArea(){  
 return length\*width;  
 }  
  
 public void reset()  
 {  
 width = 0;  
 length=0;  
 }  
  
 public double getArea(){  
 return ComputeArea;  
 }  
 public double getPerimeter(){  
 return 2\*(length+width);  
 }  
  
 @Override  
 public String toString() {  
 return "Rectangle{" +  
 "width=" + width +  
 ", length=" + length +  
 '}';  
 }  
  
  
  
  
}  
  
class Square extends Rectangle {  
  
 double side;  
 public double getSide() {  
 return getLength();  
 }  
  
 public void setSide(double side) {  
 setLength(side);  
 setWidth(side);  
 }  
 public Square()  
 {  
 super();  
 }  
  
 public Square(double side) {  
 super(side,side);  
 }  
  
 // public Square(double side, String color, boolean filled) {  
 // super(side,side,color,filled);  
 //}  
 public void setLength(double side)  
 {  
 super.setLength(side);  
 }  
 public void setWidth(double side)  
 {  
 super.setWidth(side);  
 }  
  
 @Override  
 public String toString() {  
 return "Square{" +  
 "side=" + side +  
 '}';  
 }  
}