Name:Keshav Nivrutti Yawale

Prn.number:220940325036

Git\_hub\_link: <https://github.com/keshavyawale/my_work/tree/main/220940325036>

Q1 : Write a Java program to create a new array list, add some elements (string) and print out the collection by using for-each loop. (10 Marks)

Answer:

import java.util.\*;

public class Q1 {

public static void main(String[] args) {

List <String> str = Arrays.asList("India","pakistan","bangladesh","England","Sri lanka","Iran");

ArrayList<String> ref=new ArrayList<>(str);

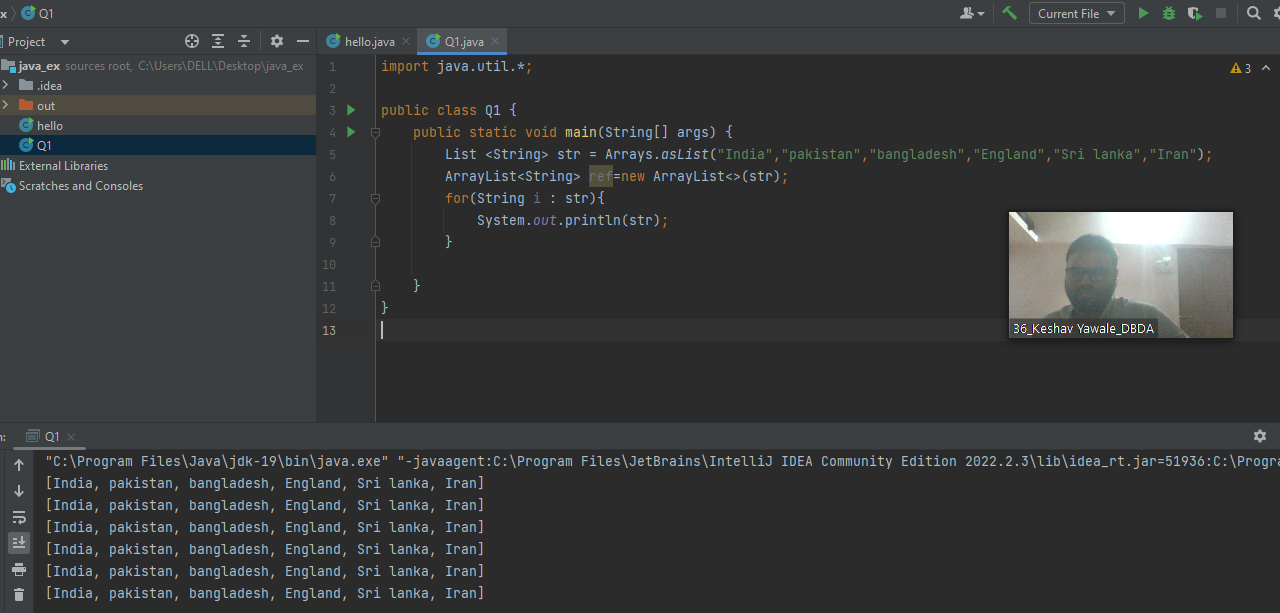
for(String i : str){

System.out.println(str);

}

}

}

****

Q3 : Write a program to create a class named shape. In this class we have three

sub classes circle, triangle and square, each class has two member function

named draw () and erase (). Create these using Runtime Polymorphism concepts. (10 Marks)

Answer:

public class Shape {

void draw(){

System.*out*.println("Drawing Shape");

}

void erase(){

System.*out*.println("Erasing Shape");

}

}

class Circle extends Shape{

void draw(){

System.*out*.println("Drawing circle");

}

void erase(){

System.*out*.println("Erasing circle");

}

}

class Triangle extends Shape{

void draw(){

System.*out*.println("Drawing Triangle");

}

void erase(){

System.*out*.println("Erasing Triangle");

}

}

class Square extends Shape{

void draw(){

System.*out*.println("Drawing Square");

}

void erase(){

System.*out*.println("Erasing Square");

}

}

class Abc{

public static void main(String[] args) {

Shape cir =new Circle();

Shape tri =new Triangle();

Shape squ=new Square();

cir.draw();

cir.erase();

tri.draw();

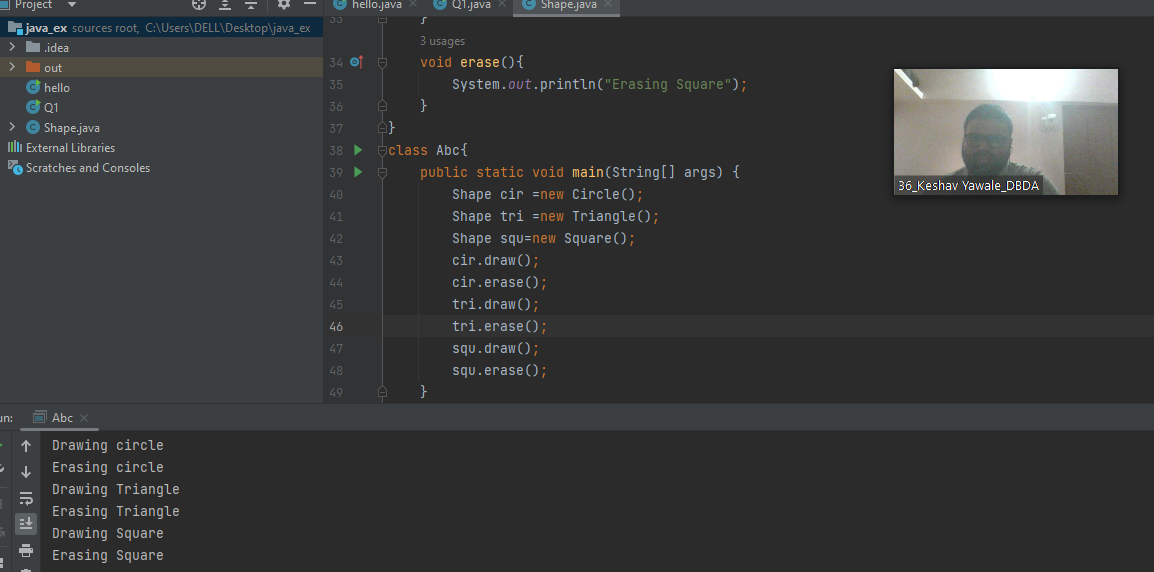
tri.erase();

squ.draw();

squ.erase();

}

}



Q2 : Develop a class BankAccount having following data members : (10 Marks)

int accno

double balance

Write appropriate constructors to initialize data members

Define the following functions :

withdraw : balance will reduce

deposit : balance will increase

show : display accno and balance

If user tries to withdraw more than the balance, use exception handling code. Demonstrate the concept of exception handling in main() function.

Answer:

import java.util.Scanner;

public class BankAccount {

int accno;

double balance;

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

BankAccount(int a , double b)

{

this.accno=a;

this.balance=b;

}

void withdraw() throws Exception

{

double amount;

System.*out*.println("enter withdraw amount");

amount=sc.nextInt();

if (balance>=amount)

{

balance-=amount;

System.*out*.println("remaining balance is "+balance);

}

else

{

throw new Exception("not sufficient balance");

}

}

void deposit() {

int amount;

System.*out*.println("enter deposit amount ");

amount = sc.nextInt();

if (amount >= 0) {

balance += amount;

System.*out*.println("balance is " + balance);

}

else

{

System.*out*.println("Invalid amount");

}

}

void show(){

System.*out*.println("account no is "+accno+" balance is "+balance );

}

}

class Ex{

public static void main(String[] args) {

BankAccount bank=new BankAccount(87654,6000);

bank.show();

bank.deposit();

bank.show();

try

{

bank.withdraw();

}

catch(Exception b)

{

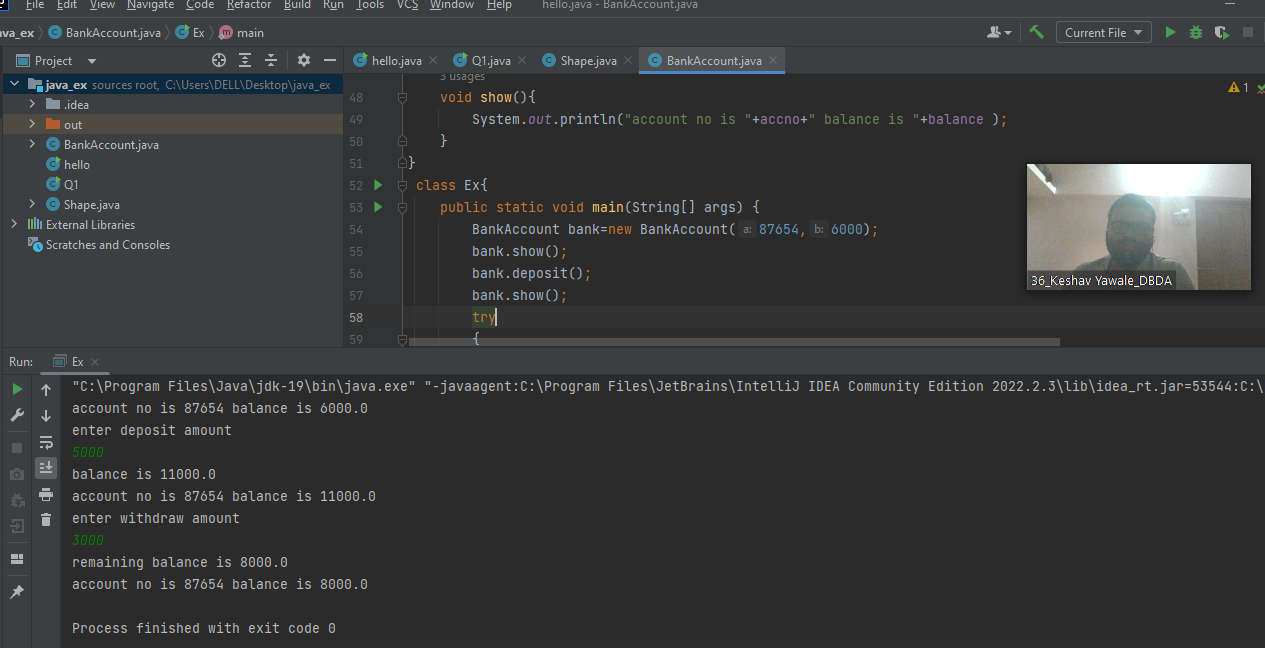
System.*out*.println(b);

}

bank.show();

}

}



Question 4.

Answer

public class GrandParent {

String grandfathername;

String grandmothername;

GrandParent(String a,String b )

{

this.grandfathername=a;

this.grandmothername=b;

System.*out*.println("name of grand father is "+grandfathername);

System.*out*.println("name of grand mother is "+grandmothername);

}

}

class Parent extends GrandParent{

String father\_name;

String mother\_name;

Parent(String c,String d,String grandfathername,String grandmothername)

{

super(grandfathername,grandmothername);

this.father\_name=c;

this.mother\_name=d;

System.*out*.println("father name "+father\_name);

System.*out*.println("mother name "+mother\_name);

}

}

class Child extends Parent{

public static void main(String[] args) {

Child ch= (Child) new GrandParent("Eknathrao","chandrabhaga");

}

}

