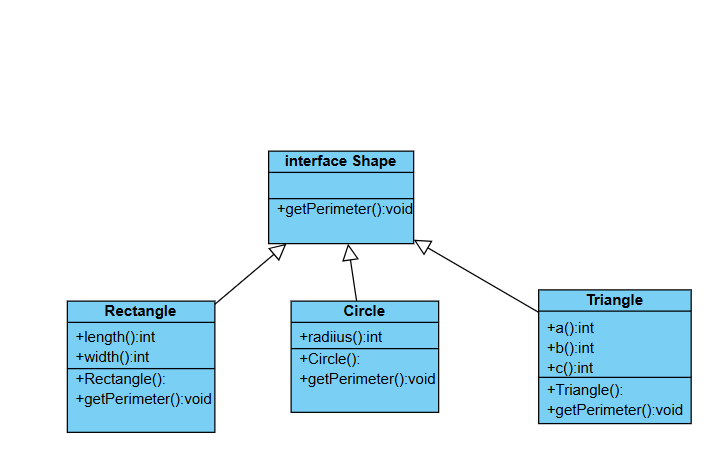
WEEK-8

1.AIM: Write a java program to create an interface Shape with the getPerimeter() method . Create three classes Rectangle, Circle, Triangle that implement the Shape interface ,Implement the getPerimeter() method for each of the three classes.

CLASS DIAGRAM:



CODE:

interface Shape{

    void getPerimeter();

}

class Rectangle implements Shape{

    int length;

    int width;

    public Rectangle(int length,int width){

        this.length=length;

        this.width=width;

    }

    public void getPerimeter(){

        System.out.println("perimeter of rectangle:"+2\*(length+width));

    }

}

class Circle implements Shape{

    int radius;

    public Circle(int radius){

        this.radius=radius;

    }

    public void getPerimeter(){

        System.out.println("perimeter of circle:"+2\*3.14\*radius);

    }

}

class Triangle implements Shape{

    int a;

    int b;

    int c;

    public Triangle(int a,int b,int c){

        this.a=a;

        this.b=b;

        this.c=c;

    }

    public void getPerimeter(){

        System.out.println("perimeter of triangle:"+a+b+c);

    }

}

Class interfaceShape{

    public static void main(String[] args){

        System.out.println("Name:B.SAM ROHITH");

        System.out.println("Roll.no:24031");

        System.out.println("Section:CSE-A");

        Triangle t=new Triangle(1,2,3);

        t.getPerimeter();

        Circle c=new Circle(5);

        c.getPerimeter();

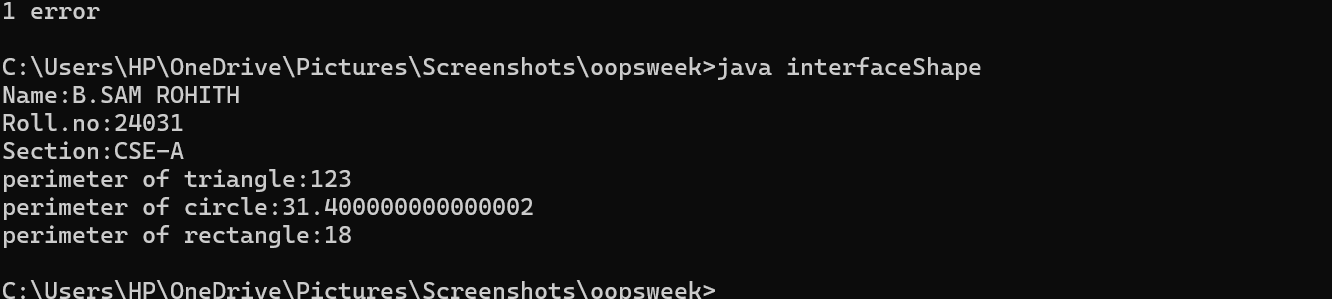
        Rectangle r=new Rectangle(4,5);

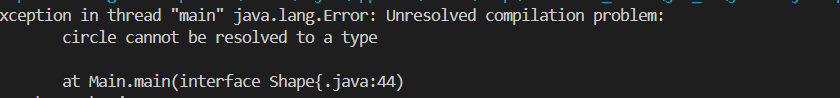
        r.getPerimeter();

    }

}

OUTPUT:

ERROR:



ERROR TABLE:

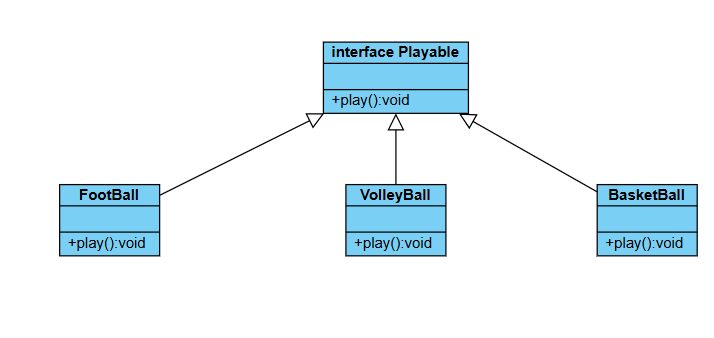
|  |  |  |  |
| --- | --- | --- | --- |
| Error type | Description | Impact | Solution |
| Syntax Error | circle is written in lowercase when creating object: circle c = new Circle(5); | Code will not compile — circle class not found | Correct to Circle c = new Circle(5); (capital C) |

IMPORTANT POINTS:

1. Shape is an interface with the method void getPerimeter();
2. All shapes (Rectangle, Circle, Triangle) implement the Shape interface.
3. Each class overrides getPerimeter() method from the Shape interface.
4. Each shape class has its own constructor to initialize dimensions.
5. Encapsulation: Each class handles its own data and behavior.
6. Abstraction: The Shape interface hides the specific implementation.

QUESTION-2 Write a java program to create an interface playable with a method play() that takes no arguments and return void. Create three classes Football, Volleyball, and Basketball that implement the playable interface and override the play() method to play the respective sports

CLASS DIAGRAM:



CODE:

interface Playable{

    void play();

}

class Football implements Playable{

    public void play(){

        System.out.println("playing a football");

    }

}

class Volleyball implements Playable{

    public void play(){

        System.out.println("playing a volleyball");

    }

}

class BasketBall implements Playable{

    public void play(){

        System.out.println("playing a basketball");

    }

}

class Main{

    public static void main(String[] args){

        System.out.println("Name:B.SAM ROHITH ");

        System.out.println("Roll.no:24031");

        System.out.println("Section:CSE-A");

        Football football = new Football();

        Volleyball volleyball = new Volleyball();

        BasketBall basketball = new BasketBall();

        football.play();

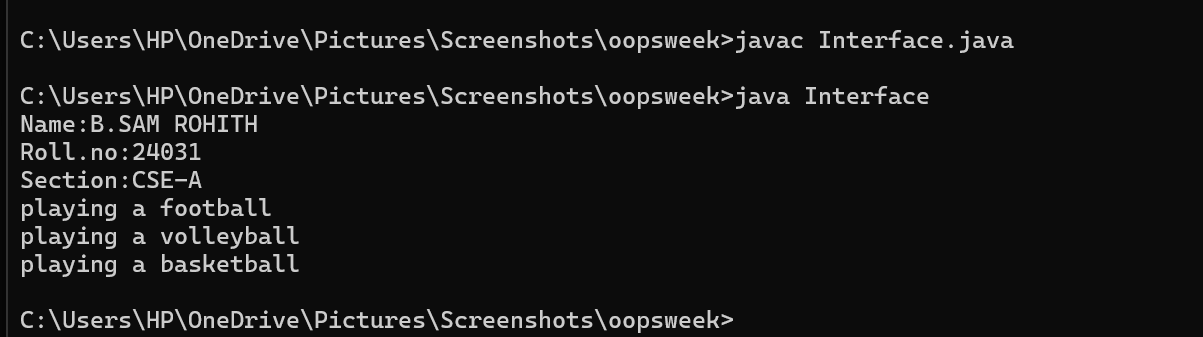
        volleyball.play();

        basketball.play();

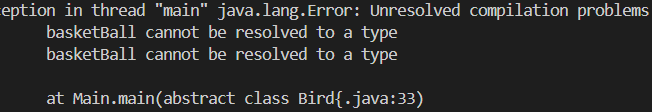
    }

}

OUTPUT:



ERROR:



ERROR TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| Errortype | Description | Impact | Solution |
| Basketball | Typo error mistake in calling an object | Leads to compilation error | Correcting the typo mistake. |

IMPORTANT POINTS:

1.In this code the main playable is an interface class .By default the method in the interface abstract.

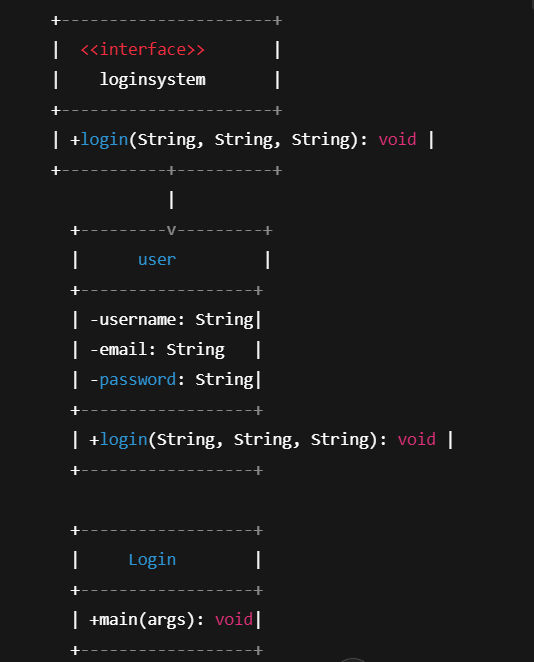
2.we use the key word implements that is used to access the Playable interface in a class.

3.We are creating objects for each class for calling the methods.

QUESTION-3

**Aim:** Write a Java program to implement a login system using interfaces.

**Class diagram:**

****

**Procedure**

**Step1:** Create a new file in Notepad name that file with the “.java” extension make sure that the class name and file name are the same for errorless code running

**Step2:** Write a Java program to implement a login system using interfaces

interface LoginSystem {

void login(String username, String email, String password);

}

class User implements LoginSystem {

String username;

String email;

String password;

// Constructor

User(String username, String email, String password) {

this.username = username;

this.email = email;

this.password = password;

}

// Implementation of login method

public void login(String inputUsername, String inputEmail, String inputPassword) {

if (inputUsername.equals(username) && inputEmail.equals(email) && inputPassword.equals(password)) {

System.out.println("Login successful!");

} else {

System.out.println("Login failed!");

}

}

}

public class Login {

public static void main(String[] args) {

User user = new User("john\_doe", "abd@gmail.com", "password123");

// Correct credentials

user.login("john\_doe", "abd@gmail.com", "password123");

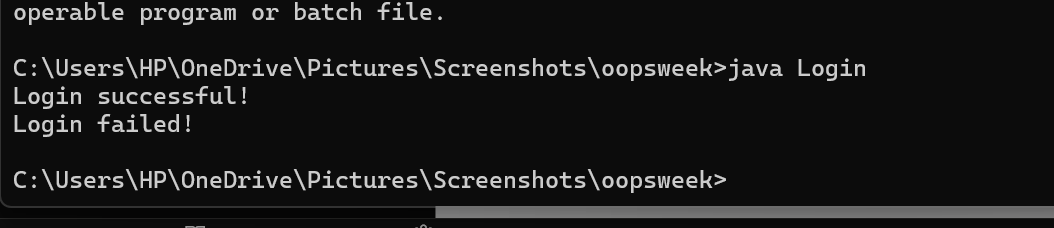
// Incorrect credentials

user.login("john\_doe", "wrongemail@gmail.com", "wrongpassword");

}

}

OUTPUT:



|  |  |  |
| --- | --- | --- |
| **S.No** | **Error** | **Solution** |
| **1.** | uequal | u.equal “.” Symbol is missing |
| **2.** | else{ | If class should be closed before starting else “}else“ |