Public class Estudiante {

Private String nombre;

Private int edad;

Private String curso;

Private double promedio;

Public Estudiante(String nombre, int edad, String curso, double promedio) {

This.nombre = nombre;

This.edad = edad;

This.curso = curso;

This.promedio = promedio;

}

// Métodos getter

Public String getNombre() {

Return nombre;

}

Public int getEdad() {

Return edad;

}

Public String getCurso() {

Return curso;

}

Public double getPromedio() {

Return promedio;

}

// Método para calcular si el estudiante está becado

Public boolean estaBecado() {

Return promedio >= 4.5;

}

// Método para calcular el nivel académico

Public String getNivelAcademico() {

If (promedio >= 4.5) {

Return “Excelente”;

} else if (promedio >= 4.0) {

Return “Muy Bueno”;

} else if (promedio >= 3.5) {

Return “Bueno”;

} else {

Return “Regular”;

}

Public class Operaciones {

Private double numero1;

Private double numero2;

Public Operaciones(double numero1, double numero2) {

This.numero1 = numero1;

This.numero2 = numero2;

}

Public double calcularOperacion(String tipoOperacion) {

Switch (tipoOperacion.toLowerCase()) {

Case “suma”:

Return numero1 + numero2;

Case “resta”:

Return numero1 – numero2;

Case “multiplicacion”:

Return numero1 \* numero2;

Case “division”:

If (numero2 ¡= 0) {

Return numero1 / numero2;

} else {

Throw new ArithmeticException(“No se puede dividir por cero”);

}

Default:

Throw new IllegalArgumentException(“Operación no válida”);

}

}

// Getters

Public double getNumero1() {

Return numero1;

}

Public double getNumero2() {

Return numero2;

}

Public class Figuras {

Private String tipo;

Private double lado1;

Private double lado2;

Public Figuras(String tipo, double lado1, double lado2) {

This.tipo = tipo;

This.lado1 = lado1;

This.lado2 = lado2;

}

Public double calcularArea() {

Switch (tipo.toLowerCase()) {

Case “circunferencia”:

Return Math.PI \* Math.pow(lado1, 2);

Case “cuadrado”:

Return lado1 \* lado1;

Case “rectangulo”:

Return lado1 \* lado2;

Case “triangulo”:

Return (lado1 \* lado2) / 2;

Default:

Return 0;

}

}

// Getters

Public String getTipo() {

Return tipo;

}

Public double getLado1() {

Return lado1;

}

Public double getLado2() {

Return lado2;