

Práctica Nº 15

Palabras Claves: **static** y **final**

1. ¿Cuál de los siguientes fragmentos de código compila? En aquellos casos en los que no compila. ¿Qué cambios habría que hacer para que compile?

```
public class Foo{
    static int x;
    public void go(){
        System.out.println(x);
    }
}
```

```
public class Foo4{
    static final int x = 12;
    public void go(){
        System.out.println(x);
    }
}
```

```
public class Foo2{
    static int x;
    public static void go(){
        System.out.println(x);
    }
}
```

```
public class Foo5{
    static final int x = 12;
    public void go(final int x){
        System.out.println(x);
    }
}
```

```
public class Foo3{
    final int x;
    public void go(){
        System.out.println(x);
    }
}
```

```
public class Foo6{
    int x = 12;
    public static void go(final int x){
        System.out.println(x);
    }
}
```

2. ¿Es posible crear una clase llamada **CadenaDeCaracteres** como subclase de la clase **String** del paquete **java.lang**? ¿Por qué? ¿Qué otras clases de la API de java tienen la misma característica?
3. Para el resto de esta práctica cree un proyecto llamado Practico15.

a. Cree una clase llamada **ComoAccederAMetodosDeClase** en la que se declaren una variable de instancia y una variable de clase, y un método de instancia y uno de clase. Intente desde ambos métodos acceder a las variables declaradas ¿Es posible? Escriba un método **main()** e intente invocar a ambos métodos sin crear ningún objeto. ¿Es posible? ¿Por qué?

b. Cree un paquete llamado **ej1** y codifique en él una clase llamada **Temperatura** con dos métodos de clase: **calcularCelsius(double temp)** y **calcularFahrenheit(double temp)**. El primero recibe como parámetro una temperatura en grados Fahrenheit, la convierte a grados Celsius y devuelve el valor convertido. El segundo recibe una temperatura en grados Celsius, la convierte a grados Fahrenheit y retorna el nuevo valor.

Pruebe la clase **Temperatura** escribiendo la clase **TestTemperatura** que en su método **main()** verifica que sus métodos realizan la conversión correctamente. Para ello, invoque al método **calcularCelsius()** con el valor 72, ¿Qué valor se obtiene como respuesta?. Invoque al método **calcularFahrenheit()** con el valor 30, ¿Qué valor se obtiene?

Sugerencia: para convertir valores **Fahrenheit** a valores **Celsius**:

- reste 32 al valor Fahrenheit, multiplique por 5 y divida entre 9.
- asegúrese que el resultado obtenido es correcto (por ejemplo realizando los cálculos en forma manual o con una máquina calculadora).

c. Escriba una clase llamada **ObjetosFelices** que declare un método de clase llamado **instancias()** que devuelva la cantidad de objetos que se crean. ¿Cómo mantiene la cantidad de objetos creados?

d. Codifique las siguientes clases:

```
public class Circulo{

    public static final double PI = 3.14159;
    protected double r;

    public Circulo(double r){
        this.r = r;
    }

    public static double radianesAgradados(double rads){
        return (rads * 180 / PI);
    }

    public double area() {
        return (PI * r * r);
    }

    public double circunferencia(){
        return (2 * PI * r);
    }

}

public class CirculoPlano extends Circulo {
    private double cx, cy;
    public static final double PI = 3;

    public CirculoPlano(double r, double x, double y){
        super(r);
        this.cx = x;
        this.cy = y;
    }

    public boolean pertenece(double x, double y){
        double dx, dy;
        dx = x - cx;
        dy = y - cy;
        double distancia = Math.sqrt(dx*dx + dy*dy);
        return (distancia < r);
    }

}
```

Use la siguiente clase de prueba y responda las preguntas que están a continuación:

```
public class TestOcultamiento {

    public static void main(String args[]){
        CirculoPlano cp=new CirculoPlano(10, 20, 10);
        System.out.println("Area : " + cp.area());
        System.out.println("Circunferencia: " + cp.circunferencia());
    }

}
```

i. ¿A qué PI hacen referencia los métodos **area()** y **circunferencia()** de TestOcultamiento?

- ii. Agregue un método estático llamado **descripción()** a las clases `Circulo` y `CirculoPlano` que retorne un `String` con una descripción general.
- iii. Luego, agréguele a la clase **TestOcultamiento** las siguientes líneas de código:

```
Circulo c= cp;  
System.out.println("Descripción : " + c.descripcion());
```

¿Es posible sobrescribir métodos de clase?