

Problema 3: Practicar con strings e instrucciones de control

Datos personales

Apellidos: Prieels

Nombre: Cedric

1 *Strings e instrucción condicional simple*

Objetivos

Practicar con los strings y la instrucción condicional simple (if)

Descripción

Se desea escribir el constructor de la clase EntradaFestival cuyo diagrama se muestra.

EntradaFestival
-String festival -double precio
+EntradaFestival(String descripcion) ...

Este constructor recibe como parámetro un texto llamado descripcion que tiene el nombre del festival y el precio de la entrada, con el formato de siguiente ejemplo:

Pirineos SUR Precio=22.00

El constructor debe depositar en el atributo festival los caracteres de la descripción anteriores a la palabra "Precio=", eliminando espacios en blanco iniciales y finales. Asimismo debe depositar en el atributo precio los caracteres posteriores a "Precio=" convertidos a número. Por ejemplo, en el caso de arriba quedará:

festival="Pirineos SUR"

precio=22.0

Si no se encuentra la palabra "Precio=" en la descripcion se pondrá festival al valor "XXX" y precio a valor Double.NaN, para indicar el error.

Métodos de la clase String que se pueden usar (buscar la documentación y ejemplos de uso en Internet). Hay que tener en cuenta que los caracteres de los strings se numeran desde 0 en adelante:

- `indexOf(String buscar)`: permite buscar un String en el String actual; retorna la posición en la que comienza el String buscar dentro del String actual, o -1 si no se encuentra
- `substring(int ini, int fin)`: permite obtener un trozo del String actual comprendido entre los caracteres ini (incluido) y fin (excluido)
- `length()`: permite obtener la longitud del String actual, en número de caracteres.
- `trim()`: retorna un String igual al original pero sin los espacios iniciales ni finales

Programación, Curso 2015-2016

Método de la clase Double (buscar la documentación y ejemplos de uso en Internet)

- static double parseDouble(String str): retorna el String str convertido a double

Respuesta:

```
public EntradaFestival (String descripcion) {

    String festival = "";
    double precio = -1.0;

    int posicion = -1;

    posicion = descripcion.indexOf("Precio=");
    if (posicion == -1) {
        festival = "XXX";
        precio = Double.NaN;
    } else {
        festival = descripcion.substring(0, posicion);
        festival = festival.trim();

        precio = descripcion.substring(posicion + 7, descripcion.length());
        // el +7 está aquí porque posicion indica el principio de "Precio=" y solo queremos el
        // número que sigue esta expresión
        precio = parseDouble(precio);
    }

    this.festival = festival;
    this.precio = precio;

}
```

2 Instrucción condicional múltiple

Objetivos

Practicar con la instrucción condicional múltiple (switch)

Descripción

Se desea escribir el constructor de la clase PasajeFerry cuyo diagrama se muestra y que contiene datos sobre una reserva de viaje en un Ferry.

PasajeFerry
+static final int ECONOMIE=1 +static final int CRUCERO=2 +static final int LUXURY=3 -int pasajeros -double coste

Programación, Curso 2015-2016

```
+Ferry (int pasajeros,int camarote)
...
```

Este constructor copia en el atributo pasajeros el parámetro del mismo nombre. Por otro lado calcula y guarda en el atributo coste el coste total del billete en función del número de pasajeros (dado por el parámetro pasajeros) y del tipo de camarote (dado por el parámetro camarote), según el precio en euros de la siguiente tabla:

Tipo de camarote	Primer pasajero	Segundo pasajero	Tercer y sucesivos pasajeros
ECONOMIE	186.00	173.00	162.00
CRUCERO	223.00	212.00	198.00
LUXURY	330.00	310.00	291.00

Por simplicidad no es preciso contemplar errores en las parámetros pasajeros y camarote. Se valorará la eficiencia de la implementación.

Respuesta:

```
public PasajeFerry (int pasajeros, int camarote) {

    double coste;
    switch(camarote) {

        case(1): // Economie

            if (pasajeros == 1) {
                coste = 186.0;
            } else if (pasajeros == 2) {
                coste = 186.0 + 173.0;
            } else if (pasajeros >= 3) {
                coste = 186.0 + 173.0 + (pasajeros-2) * 162.0;
            }

            break;
```

```
        case(2): // Crucero

            if (pasajeros == 1) {
                coste = 223.0;
            } else if (pasajeros == 2) {
                coste = 223.0 + 212.0;
            } else if (pasajeros >= 3) {
                coste = 223.0 + 212.0 + (pasajeros-2) * 198.0;
            }

            break;

        case(3): // Luxury

            if (pasajeros == 1) {
                coste = 330.0;
            } else if (pasajeros == 2) {
                coste = 330.0 + 310.0;
            } else if (pasajeros >= 3) {
                coste = 330.0 + 310.0 + (pasajeros-2) * 291.0;
            }

            break;

        default:

            coste = -1.0;
            break;

    }

    this.coste = coste;
    this.pasajeros = pasajeros;
}
```

3 Instrucciones de bucle

Objetivos

Practicar con instrucciones de bucle

Descripción

Se desea escribir un programa que lea del teclado unos números de valor creciente y sume los cuadrados de sus diferencias. El programa se detiene mostrando el resultado de la suma al encontrar

Programación, Curso 2015-2016

que dos números consecutivos no tienen valor creciente. Puede usarse este pseudocódigo transformado en un programa Java:

```
real num
real anterior
real suma=0
Lee de teclado num
hacer
    anterior=num
    Lee de teclado num
    si num>=anterior entonces
        suma=suma+(num-anterior)2
    fin si
mientras num>=anterior
Mostrar suma en pantalla
```

Respuesta

```
public static void main() {

double num;
double anterior;
double suma = 0.0;

Lectura miLectura = new Lectura("Ejercicio 3");
miLectura.creaEntrada("num",0.0);
miLectura.espera("Introduce tu numero y pulsa Aceptar");
num = miLectura.leeDouble("num");

do{

    anterior = num;

    miLectura.espera("Introduce tu numero y pulsa Aceptar");
    num = miLectura.leeDouble("num");

    if (num >= anterior) {
        suma += (num-anterior) * (num-anterior);
    }

} while (num >= anterior);

System.out.println("Suma = " + suma);

}
```

4 Atributos y métodos estáticos

Objetivo

Practicar con atributos y métodos estáticos

Descripción

Se dispone de una clase con atributos y métodos estáticos (de clase) y normales (de objeto) que obedece al siguiente diagrama:

Circulo
-static int numCirculosCreados -int xCentro, yCentro, radio +static final int grosorLapiz=2 +final int color
+Circulo (int x, int y, int radio, int color) +static int dameCirculosCreados() +pintaCirculo() -creaLienzo()

Escribir en otra clase aparte un main que a modo de ejemplo utilice todos los atributos y métodos públicos de la clase Circulo.

Respuesta

```
public static void main () {  
  
    Circulo miCirculo1 = new Circulo (1, 2, 3, "rojo");  
    Circulo miCirculo2 = new Circulo (4, 5, 6, "verde");  
  
    miCirculo1.pintaCirculo();  
    miCirculo2.pintaCirculo();  
  
    System.out.println("Numero de circulos creados = " + miCirculo2.dameCirculosCreados());  
    // Sin conocer la clase Circulo, solo puedo suponer que este método devuelve el  
    // número de circulos creados  
  
    System.out.println("Color del circulo 1 = " + miCirculo1.color);  
    System.out.println("Grosor del lapiz usado para el circulo 1 = " + miCirculo1.grosorLapiz);  
  
}
```