

Programación en Lenguaje Java

Práctica 3.1. Piscifactoría



Michael González Harbour
Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Práctica 3-1: Piscifactoría

Objetivos: Instrucciones de control, pseudocódigo y entrada/salida

Descripción: Se desea hacer una aplicación que nos permita realizar cálculos del crecimiento esperado de peces en una piscifactoría

Notación:

- p : peso del pez en gramos
- p_0 : peso inicial del pez, en gramos
- d : tiempo transcurrido en días
- $t(d)$: temperatura media de un día en grados centígrados
- $T(d)$: suma de las temperaturas medias de los días de crecimiento
- CCT : coeficiente de crecimiento térmico, en $\text{gramos}^{1/3}/(\text{día} \cdot ^\circ\text{C})$
- K : factor de crecimiento en $\text{gr}^{1/3}$

Cálculo del crecimiento y clases

El cálculo del peso se hace con la siguiente ecuación

$$p = (p_0^{1/3} + CCT \cdot T(d) + K)^3$$

Diseñaremos el software con una clase llamada **Pez** que hace los cálculos de crecimiento del pez, y una clase **Tanque** que contiene el **main**

Fuente: Modelo aleatorio de crecimiento CCT biparamétrico, Alamar, M.; Estruch, V.; Pastor, J. y Vidal, A.

<http://www.revistaaquatic.com/aquatic/art.asp?t=h&c=118>

Clase Pez

Los atributos de la clase **Pez** son:

- coeficientes de crecimiento (CCT y K): **cct, k**
- peso inicial (p_0): **p0**
- días transcurridos (d): **d**
- suma de temperaturas ($T(d)$): **sumT**

Constructor:

- se le pasan como parámetros los valores iniciales de los coeficientes y el peso inicial, en las unidades indicadas en la notación; pone **d** y **sumT** a cero
- si el peso inicial es incorrecto se pone al valor **Double.NaN**
 - el peso es correcto cuando está entre 0 y 50gr, ambos valores incluidos

Métodos de la clase Pez

Método `avanzaUnDia()`: simula el paso de un día. Para ello:

- el método recibe como parámetro la temperatura media del día `td`, en grados centígrados
- se calcula la temperatura eficaz `te`:
 - si $td < 10^\circ$, la eficaz es cero (por debajo de 10° el pez no crece)
 - si $td > 30^\circ$, la eficaz se hace `Double.NaN` (el pez se muere)
 - en otros casos, la eficaz es `td - 10`
- se incrementa el número de días transcurridos en uno, y se suma la temperatura eficaz a `sumT`

Método `peso()`: calcula y retorna el peso actual del pez, usando la fórmula del peso

Método `dias()`: retorna el número de días transcurridos: atributo `d`

Clase Tanque

Contiene el programa principal, en el que se simula un crecimiento inicial a temperatura constante (con calentadores en invierno) y luego a temperatura aleatoria en verano.

Usar el siguiente pseudocódigo:

```
método main
  Grafica g = nueva Grafica(títulos)
  // pez con CCT, K, p0
  Pez lubina = nuevo Pez
    (0.000897653, 0.116375, 38.97)
  real peso // en gr
  real temp; // temperatura en grados
```

Pseudocódigo (cont.)

```
para d desde 1 hasta 220
    si d<120 entonces
        temp=20º
    si no
        temp=15+número aleatorio entre 0 y 14
    fin si
    lubina.avanzaDia(temp)
    peso=lubina.peso()
    g.inserta(d,peso)
fin para
g.pinta()
fin método
```

Notas:

- en el constructor de la gráfica poner títulos apropiados a la gráfica y a los ejes
- elegir un color y atributos (símbolos, líneas) concretos para la gráfica

Parte avanzada

Crear otra clase similar a **Tanque** en la que sobre la misma ventana de la gráfica anterior se muestre otra gráfica del crecimiento de otro pez con $CCT=0.000815773$, $F=0.232579$, $p_0=31.86$. En este caso, las temperaturas se obtendrán de la siguiente tabla:

Num. días	Temp. media (°C)
72	23.0
62	25.0
52	26.0
34	29.0

Nota: no usar **switch**: saldría muy largo y ocuparía mucha memoria

Después de pintar la gráfica, añadir un día a 31° y comprobar que el peso sale con valor **Double.NaN** (pez muerto)

Entregar

Un *informe* conteniendo

- el diagrama de clases de la clase Pez (1 punto)
- el código de la clase **Pez** (3 puntos)
- el código de la clase **Tanque** (2.5 puntos)
- una captura de pantalla de la gráfica (1.5 puntos)

Si se ha hecho la parte avanzada

- el código de la nueva clase (1 punto)
- una captura de pantalla de la nueva gráfica (1 punto)

Notas

Tener cuidado con la aritmética de números enteros

- Recordar que en Java $1/3=0$

Consultar el pseudocódigo y las instrucciones de bucle en el capítulo 3 de la asignatura

- en particular el bucle `for`