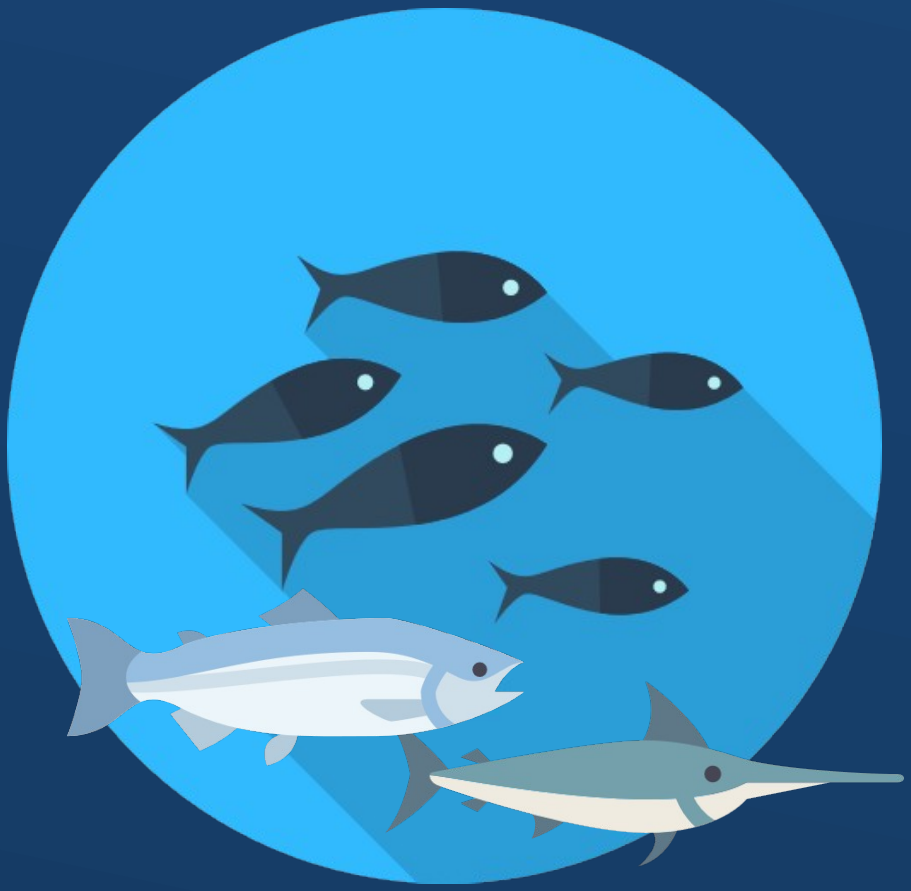
Adrián Fernández González



Índice

Proyecto I. Base

Introducción

La idea base del proyecto es crear un programa que simule una piscifactoría y toda la infraestructura que lo rodea, incluyendo el ciclo de vida de los peces, su alimentación, reproducción y gestión de piscifactorías y edificios.

A lo largo del módulo se **ampliará y modificará** cada vez que se vea algo nuevo, por lo que el proyecto ha de guardarse a buen recaudo, pues se usará la entrega anterior para crear la siguiente.

Es importante **corregir** los fallos detectados en cada entrega para que las siguientes funcionen correctamente.

El proyecto se realizará en parejas y todo el código ha de subirse a un **repositorio GIT** para su seguimiento.

Todas las entregas se realizarán mediante un **zip o rar** con los archivos del sistema y los documentos a modo de manual.

Debe **respetarse el nombre** de todo lo estipulado, tanto clases como paquetes.

Lee detenidamente este documento antes de empezar. Crear un diagrama en papel ayuda a visualizar las cosas.

Algunos métodos de la clase *Simulador* pueden necesitar la implementación de métodos adicionales en la clase *Piscifactoría* que no aparecen reflejados en este documento. También se pueden crear tantos métodos auxiliares como sea necesario.

Algunos datos se describen superficialmente, ya que su implementación puede realizarse de diversas formas y queda a tu disposición escoger la forma más oportuna. Ten en cuenta que, durante la corrección, has de justificar tu elección de forma razonable.

Ten en cuenta que, aunque se describa un proceso en una sección determinada, su implementación puede ser realizada en otro lugar.

Sistema de monedas

Dado que es un simulador, este cuenta con un sistema de monedas, de tal forma que ciertas acciones cuestan dinero y otras dan. En cada sección se especifica los costes y ganancias de cada acción.

En todo lugar en el que se pueda gastar monedas, se mostrará el número de monedas disponibles. Si no hay monedas suficientes, se mostrará un mensaje acorde y no se realizará la acción.

Comida

El sistema dispone de dos tipos de alimento, vegetal y animal. Esta está almacenada en las piscifactorías, que disponen de dos almacenes propios, uno para cada tipo de comida con igual capacidad.

Todas las cantidades que aparecen en el documento se refieren a ambas capacidades y estas aumentan juntas si se mejora el almacén.

Ciclo de vida

En términos generales, un pez realiza el siguiente ciclo de vida:

1. Se alimenta.
2. Si no está alimentado tiene, por defecto, un 50% de morir.
3. Aumenta en 1 su edad.
4. Se verifica su madurez y su fertilidad.
5. Se reproduce si es posible.

Aquellos peces que no han llegado a la madurez, tienen un 5% de morir cada **día par** antes de su madurez (incluido).

Reproducción

Un pez se reproduce si ha llegado a la madurez y es una hembra fértil y hay otro macho fértil en su mismo tanque. No es necesario emparejarlos, con que haya un macho en el tanque es suficiente.

Al reproducirse, aparecen los peces directamente, equilibrándose 50/50 entre machos y hembras presentes en el tanque.

Ten en cuenta que, tras la reproducción, tienen un número de días en los que no se reproduce.

Simulador

La clase que contiene el main, el menú y la lógica general.

Atributos

- El número de días que han pasado. Empieza en 0.
- Las piscifactorías que hay.
- El nombre de la entidad/empresa/partida.

init

Método que inicializa todo el sistema, pidiendo el nombre de la entidad/empresa/partida, pidiendo el nombre de la primera piscifactoría y creando todos los objetos necesarios.

Inicialmente el sistema cuenta con una piscifactoría de río con el almacén de comida lleno (25) y 100 monedas.

menu

Método que muestra el texto del menú.

1. Estado general
2. Estado piscifactoría
3. Estado tanques
4. Informes
5. Ictiopedia
6. Pasar día
7. Comprar comida
8. Comprar peces
9. Vender peces
10. Limpiar tanques
11. Vaciar tanque
12. Mejorar
13. Pasar varios días
14. Salir

menuPisc

Método que muestra la lista de piscifactorías actuales en forma de menú, más una opción 0 para cancelar de esta forma:

“Seleccione una opción:”
“---------- Piscifactorías ----------”

“[Peces vivos / Peces totales / Espacio total]” Esta línea es el texto tal cual.
“x.- Nombre [vivos/total/espacio]”

selectPisc
Método que muestra el menú de piscifactorías y permite seleccionar una de ellas, devolviendo la opción seleccionada.

selectTank
Método que muestra el menú de tanques de una piscifactoría y permite seleccionar uno de ellos, devolviendo la opción seleccionada. En cada tanque aparece el número del mismo y el tipo de pez que admite.

showGeneralStatus
Método que muestra el estado de las piscifactorías, la comida disponible en cada una y todos sus datos. También muestra el día actual y el número de monedas disponibles.

Si se dispone del almacén central, muestra la cantidad almacenada, el máximo y el % que representa.

showSpecificStatus
Método que muestra un menú para seleccionar una piscifactoría y luego muestra el estado de todos los tanques de dicha piscifactoría.

showTankStatus
Método que muestra un menú para seleccionar un tanque de una piscifactoría y luego muestra el estado de todos los peces de dicho tanque.

showStats
Método que muestra un desglose de los peces del sistema indicando su nombre, el número de ellos comprados, el número de ellos que han nacido, cuantos fueron vendidos y el dinero ganado con ello.

Por último, muestra un total general con las cuatro cantidades.

showIctio
Método que permite consultar la información de cada pez mostrando un menú de selección de los disponibles.

nextDay
Método que avanza un día en todas las piscifactorías, realizando el crecimiento, reproducción y venta de peces óptimos.

Luego, muestra el resultado “Piscifactoría X: Y peces vendidos por Z monedas” por cada piscifactoría y, luego, un resultado conjunto, “X peces vendidos por un total de Y monedas”.

addFood

Método que añade comida a una piscifactoría seleccionada, permitiendo escoger el tipo de comida y la cantidad a añadir entre 5, 10, 25 o llenar. Esto ha de solicitarse mediante un menú textual, no insertar un número de cantidad.

Cada 1 de comida cuesta 1 moneda, pero por cada 25 añadidos, se descuentan 5 monedas. Ten en cuenta de que, si se solicita una cantidad mayor de comida que el máximo, solo se cobra lo que se llena y el descuento se aplica solo por cada 25 añadidos.

Luego muestra la cantidad de comida añadida, y el estado del depósito de comida de ese tipo. “Añadida X de comida Y”, siendo Y el tipo de comida, “Depósito de comida X de la piscifactoría Z al X% de su capacidad. [comida/max]”.

En caso de disponer de la mejora de almacén central, se salta la selección de la piscifactoría, pues la comida va a ese almacén y se distribuye entre todas las piscifactorías de forma equitativa.

addFish

Método que añade un pez escogido por el usuario a una piscifactoría si hay sitio. Si no, mostrará un mensaje indicándolo. El sexo es el menor de los disponibles o, en su defecto, hembra.

Luego muestra un mensaje del estado del tanque de esa piscifactoría. “Tanque # de la piscifactoría x al y% de capacidad. [peces/espacios]”.

sell

Método que vende todos los peces adultos que estén vivos haciendo la lógica necesaria. Esto solo se realiza de una única piscifactoría seleccionada.

Dado que se realiza una venta prematura, darán la mitad del dinero de lo normal, redondeando hacia abajo. Los longevos darán la mitad de lo que iban a dar.

Luego, muestra el resultado “Piscifactoría X: Y peces vendidos por Z monedas”.

cleanTank

Elimina todos los peces muertos de los tanques de una piscifactoría.

emptyTank

Elimina todos los peces de un tanque concreto independientemente de su estado.

upgrade

Método que permite escoger qué mejorar.

1. Comprar edificios
    a. Piscifactoría. Pedirá el nombre a darle y el tipo.
    b. Almacén central.
2. Mejorar edificios
    a. Piscifactoría
        i. Comprar tanque.
        ii. Aumentar almacén de comida.
    b. Almacén central (Si se tiene)
        i. Aumentar capacidad.
3. Cancelar.

Piscifactoría de río

Comprar una piscifactoría en un río cuesta 500 x el número que haya ya (mínimo 500). Por ejemplo, si hay 3 de río, costará 1500 comprar una nueva.

Comprar un tanque para ella cuesta 150 por cada uno que ya tenga. Como máximo, una piscifactoría puede disponer de 10 tanques.

Aumentar el almacén de comida cuesta 50 monedas y la aumenta en 25 unidades ambas capacidades. Como máximo, el almacén puede disponer de una capacidad de 250 de cada tipo.

Piscifactoría de mar

Comprar una piscifactoría en el mar cuesta 2000 x el número que haya ya (mínimo 2000). Por ejemplo, si hay 3 en el mar, costará 6000 comprar una nueva.

Comprar un tanque para ella cuesta 600 por cada uno que ya tenga. Como máximo, una piscifactoría puede disponer de 10 tanques.

Aumentar el almacén de comida cuesta 200 monedas y la aumenta en 100 unidades ambas capacidades. Como máximo, el almacén puede disponer de una capacidad de 1000 de cada tipo.

Almacén central

El almacén central cuesta 2000 monedas y permite almacenar comida en él para ser distribuida equitativamente entre las piscifactorías cuando se compra o pasa un día.

Inicialmente tiene una capacidad de 200 unidades de cada tipo y puede ser mejorada infinitamente por 200 monedas cada 50 de capacidad de ambos tipos.

main

En el *main* se mostrará el menú y se pedirá la acción a realizar.

Si es la primera vez que entra, llamará al método *init*.

Luego, hará la lógica de cada día, primero se muestra el día actual y luego el menú, en el que el usuario podrá elegir múltiples opciones o salir.

Si se selecciona otra opción distinta de pasar día, realizará lo que tiene que hacer y volverá a mostrar el menú sin mostrar el estado.

Al seleccionar pasar de día, hará la lógica necesaria y volverá a mostrar el estado y el menú.

Se añaden dos opciones ocultas al menú, la 98 que añade cuatro peces al azar y de forma gratuita a una piscifactoría seleccionada y la 99, que añade 1000 monedas para cuestiones de pruebas.

Piscifactoría

La clase piscifactoría representa el lugar en el que se crían y cuidan los peces. Hay de dos tipos, de río y de mar, las cuales solo pueden albergar especies de dichos lugares.

Inicialmente tienen un tanque, con espacio para 25 peces las de río y 100 las de mar.

El almacén de comida inicial es de 25 para las de río y 100 para las de mar.

showStatus

Muestra toda la información de la piscifactoría:

"========== nombre =========="
"Tanques: numero"
"Ocupación: peces / max (x%)"
"Peces vivos: vivos / total (x%)"
"Peces alimentados: alimentados / vivos (x%)"
"Peces adultos: adultos / vivos (x%)"
"Hembras / Machos: H/M"
"Fértiles: fértiles / vivos"
"Almacén de comida: actual / max (x%)"

showTankStatus

Muestra la información de cada tanque:

"========== Tanque # =========="
"Ocupación: peces / max (x%)"
"Peces vivos: vivos / total (x%)"
"Peces alimentados: alimentados / vivos (x%)"
"Peces adultos: adultos / vivos (x%)"
"Hembras / Machos: H/M"
"Fértiles: fértiles / vivos"

showFishStatus

Muestra la información de todos los peces de un tanque determinado.

showCapacity

Muestra la ocupación de un tanque determinado. "Tanque # de la piscifactoría x al y% de capacidad. [peces/espacios]".

showFood

Muestra el estado del almacén de comida. “Depósito de comida de la piscifactoría Y al x% de su capacidad. [comida/max]”

nextDay

Hace la lógica de pasar de día de todos los peces de la piscifactoría.

A la hora de alimentarlos, cogerán la comida del almacén y, si esta se acaba, se intentará traer del almacén central.

sellFish

Vende todos los peces de la piscifactoría que sean adultos y estén vivos.

upgradeFood

Mejora el almacén de comida. Luego, mostrará un mensaje “Almacén de comida de la piscifactoría X mejorado. Su capacidad ha aumentado en Y hasta un total de Z”

Tanque

La clase tanque representa el lugar en el que están los peces. Cada uno solo puede albergar un único tipo de pez.

Esta clase dispone de todos los métodos necesarios para la gestión de los peces desde la clase piscifactoría, intentando conservar los nombres.

showStatus

Muestra la información del tanque:

"========== Tanque # =========="
"Ocupación: peces / max (x%)"
"Peces vivos: vivos / total (x%)"
"Peces alimentados: alimentados / vivos (x%)"
"Peces adultos: adultos / vivos (x%)"
"Hembras / Machos: H/M"
"Fértiles: fértiles / vivos"

showFishStatus

Muestra la información de todos los peces del tanque.

showCapacity

Muestra la ocupación del tanque. "Tanque # de la piscifactoría x al y% de capacidad. [peces/espacios]".

nextDay

Hace crecer todos los peces del tanque, luego realiza el proceso de reproducción y, por último, vende aquellos que hayan llegado a la edad óptima.

Pez

Clase padre de los peces.

Ten en cuenta que habrá múltiples peces, con características propias, por lo que, será necesaria la implementación de múltiples clases que hereden de esta clase padre. También, es posible que, para facilitar la creación de las clases, necesites u optes por crear otras clases intermedias que aúnen peces con cualidades similares.

También será necesaria la capacidad para diferenciar si un pez es de mar o de río para saber en qué piscifactoría podrá ser introducido.

Las características propias de cada pez serán descritas en ***Peces.xlsx*** y facilitadas en la librería ***Orca***, por lo que aquí se describen los elementos generales.

Atributos

- Nombre común del pez. No puede ser modificado.
- El nombre científico del pez. No puede ser modificado.
- La edad.
- El sexo. No puede ser modificado
- Si es fértil.
- Si está vivo

showStatus

Muestra el estado del pez:

"---------- Nombre ----------"
"Edad: x días"
"Sexo: H/M"
"Vivo: Si/No"
"Alimentado: Si/No"
"Adulto: Si/No"
"Fértil: Si/No"

grow

Hace crecer un día el pez, realizando toda la lógica.

Ten en cuenta que esto solo se realiza si el pez está vivo, en caso de estar muerto, no se realiza nada, ni aumentar su edad.

Si muere, recuerda cambiar los valores de aquellos atributos que puedan acarrear problemas.

reset

Reinicia el pez, estableciendo su edad a cero y el resto de atributos a sus estados iniciales.

Almacén central

Clase que representa el almacén central. Este permite almacenar comida en él para ser distribuida equitativamente entre las piscifactorías cuando se compra la comida o se pasa un día.

Inicialmente tiene una capacidad de 200 unidades de cada tipo y mejorable infinitamente.

Consideraciones adicionales

toString

Toda clase ha de tener el método *toString* sobrescrito para devolver la información relevante de la misma.

Seguridad

Ten en cuenta que, en todo el sistema, tienes que recoger posibles excepciones y salvaguardarlo de errores del usuario como campos mal introducidos.

Usabilidad

A la hora de diseñar los menús, opciones y demás interfaz, piensa en su fácil uso de cara al usuario, de tal forma que haya cosas como cancelaciones, vueltas atrás, etc.

Librería Orca

Los datos de los peces, así como sus características y parte del sistema de informes, viene dado por una librería, ***Orca_lib.jar***.

La librería puede no estar disponible inicialmente, por lo que realiza las otras partes primero.