Manual técnico

Piscifactoría

Lois Domínguez Domínguez, Manuel Bértolo Blanco, Miriam Betanzos Jamardo

Componentes **10**
Sistema de monedas 10
Clases 10
SistemaMonedas 10
Atributos 10
private int monedas 10
Métodos 10
SistemaMonedas() 10
SistemaMonedas(int monedas) 10
getMonedas() 10
setMonedas(int monedas) 11
toString() 11
Generador de menús 12
Clases 12
GeneradorMenus 12
Métodos 12
generarMenu(String[] opciones, int numeroOpcionInicial) 12
generarMenu(String[] cabecera, String[] opciones, int numeroOpcionInicial) 12
generarMenuOperativo(String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal) 12
generarMenuOperativo(String[] cabecera, String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal) 13
toString() 13
Sistema de entrada 14
Clases 14
SistemaEntrada 14
Atributos 14
private static final BufferedReader BUFER_ENTRADA 14
Métodos 14
entradaOpcionNumerica(int minimo, int maximo) 14
entradaOpcionNumerica(int[] opciones) 14
entradaTexto() 14
entradaOpcionNumericaEnteraPositiva() 15
close() 15
toString() 15
FechaTiempoLocal 16
Clases 16
FechaTiempoLocal 16
Métodos 16
obtenerFechaTiempoActual() 16
SistemaFicheros 17
Clases 17
SistemaFicheros 17

Métodos 17
crearArchivo(String rutaArchivo) 17
crearCarpeta(String rutaCarpeta) 17
crearEstructuraCarpetas(String estructuraCarpetas) 17
existeArchivo(String rutaArchivo) 18
existeDirectorio(String rutaDirectorio) 18
isDirectorioVacio(String rutaDirectorio) 18
borrarArchivo(String rutaArchivo) 18
LecturaEscrituraFicherosPlanos 19
Clases 19
LecturaEscrituraFicherosPlanos 19
Métodos 19
escrituraFicheroTextoPlano(File archivo, String texto, String codificacion) 19
escrituraFicheroTextoPlanoSinSobreescritura(File archivo, String texto, String codificacion) 19
lecturaFicheroTextoPlano(File archivo, String codificacion) 20
Clases propias **21**
Clase Pez **21**
Atributos 21
protected final String nombre 21
protected final String nombreCientifico 21
protected int edad 21
protected final boolean sexo 21
protected boolean fertil 21
protected boolean vivo 21
protected boolean alimentado 21
protected int diasSinReproducirse 21
Métodos 21
getNombre() 21
getNombreCientifico() 22
getEdad() 22
isSexo() 22
isFertil() 22
isVivo() 22
isAlimentado() 22
getDiasSinReproducirse() 22
setEdad(int edad) 23
setFertil(boolean fertil) 23
setVivo(boolean vivo) 23
setAlimentado(boolean alimentado) 23
setDiasSinReproducirse(int diasSinReproducirse) 23
Pez(String nombre, String nombreCientifico, boolean sexo) 23
showStatus() 24

grow() 24
reset() 24
comer() 24
isMaduro() 24
isEdadOptima() 24
obtenerPezHijo() 24
obtenerPezHija() 24
String toString() 25
Peces **26**
Clase Carnívoro **26**
Métodos 26
Carnivoro(String nombre, String nombreCientifico, boolean sexo) 26
Clase Filtrador **26**
Métodos 26
Filtrador(String nombre, String nombreCientifico, boolean sexo) 26
comer() 26
Clase Omnívoro **27**
Métodos 27
Omnivoro(String nombre, String nombreCientifico, boolean sexo) 27
comer() 27
Clase Caballa **28**
Métodos 28
Caballa(boolean sexo) 28
showStatus() 28
grow() 28
isMaduro() 28
isEdadOptima() 28
obtenerPezHijo() 28
obtenerPezHija() 29
Clase Pejerrey **30**
Métodos 30
Pejerrey(boolean sexo) 30
showStatus() 30
grow() 30
isMaduro() 30
isEdadOptima() 30
obtenerPezHijo() 30
obtenerPezHija() 31
Clase PercaEuropea **32**
Métodos 32
PercaEuropea(boolean sexo) 32
showStatus() 32
grow() 32
comer() 32

isMaduro() 32
isEdadOptima() 32
obtenerPezHijo() 33
obtenerPezHija() 33
Clase Robalo **34**
Métodos 34
Robalo(boolean sexo) 34
showStatus() 34
grow() 34
isMaduro() 34
isEdadOptima() 34
obtenerPezHijo() 34
obtenerPezHija() 35
Clase SalmonAtlantico **36**
Métodos 36
SalmonAtlantico(boolean sexo) 36
showStatus() 36
grow() 36
isMaduro() 36
isEdadOptima() 36
obtenerPezHijo() 36
obtenerPezHija() 37
Clase SalmonChinook **38**
Métodos 38
SalmonChinook(boolean sexo) 38
showStatus() 38
grow() 38
isMaduro() 38
isEdadOptima() 38
obtenerPezHijo() 38
obtenerPezHija() 39
Clase ArenqueDelAtlantico **40**
Métodos 40
ArenqueDelAtlantico(boolean sexo) 40
showStatus() 40
grow() 40
isMaduro() 40
isEdadOptima() 40
obtenerPezHijo() 40
obtenerPezHija() 41
Clase TilapiaDelNilo **42**
Métodos 42
TilapiaDelNilo(boolean sexo) 42
showStatus() 42

grow() 42
isMaduro() 42
isEdadOptima() 42
obtenerPezHijo() 42
obtenerPezHija() 43
Clase Abadejo **44**
Métodos 44
Abadejo(boolean sexo) 44
showStatus() 44
grow() 44
isMaduro() 44
isEdadOptima() 44
obtenerPezHijo() 44
obtenerPezHija() 45
Clase CarpinTresEspinas **46**
Métodos 46
CarpinTresEspinas(boolean sexo) 46
showStatus() 46
grow() 46
comer() 46
isMaduro() 46
isEdadOptima() 46
obtenerPezHijo() 47
obtenerPezHija() 47
Clase Dorada **48**
Métodos 48
Dorada(boolean sexo) 48
showStatus() 48
grow() 48
isMaduro() 48
isEdadOptima() 48
obtenerPezHijo() 48
obtenerPezHija() 49
Clase Sargo **50**
Métodos 50
Sargo(boolean sexo) 50
showStatus() 50
grow() 50
isMaduro() 50
isEdadOptima() 50
obtenerPezHijo() 50
obtenerPezHija() 51
AlmacenCentral **52**
Atributos 52

private int capacidadComida 52
private int cantidadComidaAnimal 52
private int cantidadComidaVegetal 52
Métodos 52
AlmacenCentral() 52
getCapacidadComida() 52
setCapacidadComida(int capacidadComida) 52
getCantidadComidaAnimal() 52
setCantidadComidaAnimal(int cantidadComidaAnimal) 53
getCantidadComidaVegetal() 53
setCantidadComidaVegetal(int cantidadComidaVegetal) 53
mejorar() 53
toString() 53
Clase Tanque **54**
Atributos 54
private final int numeroTanque 54
private int capacidadMaximaPeces 54
private ArrayList<Pez> peces 54
Métodos 54
getCapacidadMaximaPeces() 54
setCapacidadMaximaPeces(int capacidadMaximaPeces) 54
getPeces() 54
setPeces(ArrayList<Pez> peces) 54
getNumeroTanque() 55
Tanque(int numeroTanque, int capacidadMaximaPeces) 55
showStatus() 55
pecesVivos() 55
pecesAlimentados() 55
pecesAdultos() 55
pecesAdultosVivos() 55
pecesMacho() 56
pecesHembra() 56
pecesFertiles() 56
showFishStatus() 56
showCapacity(String piscifactoria) 56
alimentar(Piscifactoria.AlmacenComida almacenComida) 56
alimentarAleatorio(ArrayList<Integer> cantidadDeComidaNecesariaPorPez, Piscifactoria.AlmacenComida almacenComida, int comidaDisponible) 57
hayMachoFertil() 57
reproducir() 57
venderPecesOptimos() 57
venderPeces() 57
eliminarPecesMuertos() 57

nextDay() 57
vaciarTanque() 57
toString() 58
Clase Piscifactoría **59**
Clases 59
AlmacenComida 59
Atributos 59
private int capacidadMaximaComida 59
private int cantidadComidaAnimal 59
private int cantidadComidaVegetal 59
Métodos 59
AlmacenComida(int capacidadMaximaComida, int cantidadComidaAnimal, int cantidadComidaVegetal) 59
getCapacidadMaximaComida() 59
setCapacidadMaximaComida(int capacidadMaximaComida) 60
getCantidadComidaAnimal() 60
setCantidadComidaAnimal(int cantidadComidaAnimal) 60
getCantidadComidaVegetal() 60
setCantidadComidaVegetal(int cantidadComidaVegetal) 60
mejorar(int capacidadAAumentar) 60
toString() 61
Atributos 61
protected ArrayList<Tanque> tanques 61
protected Tanque tanqueInicial 61
protected AlmacenComida almacenInicial 61
protected String nombre 61
Métodos 61
Piscifactoria(String nombre) 61
getTanques() 61
setTanques(ArrayList<Tanque> tanques) 61
getTanqueInicial() 62
setTanqueInicial(Tanque tanqueInicial) 62
getNombre() 62
getAlmacenInicial() 62
setAlmacenInicial(AlmacenComida almacenInicial) 62
setNombre(String nombre) 62
showStatus() 62
datosTanques() 63
showTankStatus() 63
showFishStatus() 63
showCapacity() 63
showFood() 63
sellFish() 63
upgradeFood() 63

nextDay() 63
getPecesVivos() 63
getPecesTotales() 64
getEspacioPeces() 64
getIndiceTanqueVacio() 64
getIndiceTanqueConEspacioParaPez(String nombrePez) 64
isTodosLosTanqueLlenos() 64
toString() 64
Clase PiscifactoriaRio **65**
Métodos 65
PiscifactoriaRio(String nombre) 65
upgradeFood() 65
toString() 65
Clase PiscifactoriaMar **65**
Métodos 65
PiscifactoriaMar(String nombre) 65
upgradeFood() 65
toString() 66
Clase Simulador **67**
Atributos 67
private static int diasPasados 67
public static ArrayList<Piscifactoria> piscifactorias 67
private static String nombre 67
public static SistemaMonedas sistemaMonedas 67
public static AlmacenCentral almacenCentral 67
public static Estadisticas estadisticas 67
public static final File archivoLogsGeneral 67
public static File archivoLogPartida 67
public static File archivoTranscripcionesPartida 67
public static File archivoGuardadoPartida 67
Métodos 67
init() 67
menu() 68
menuPisc() 68
selectPisc() 68
selectTank(Piscifactoria piscifactoria) 68
showGeneralStatus() 68
showSpecificStatus() 68
showTankStatus(Piscifactoria piscifactoria) 68
showStats() 68
showIctio() 69
nextDay() 69
addFood() 69
menuTipoComida() 70

menuCantidadComida() 70
calcularCosto(int cantidad) 70
addFish() 70
addFishMar(Piscifactoria piscifactoria) 70
addFishRio(Piscifactoria piscifactoria) 70
crearPezMar(int pez, boolean sexo) 71
crearPezRio(int pez, boolean sexo) 71
mostrarInformacionPez(PecesDatos datosDelPez) 71
sell() 71
cleanTank() 71
emptyTank() 71
upgrade() 71
comprarEdificio() 72
comprarPiscifactoria() 72
mejorarEdificio() 72
mejorarPiscifactoria() 72
mejorarAlmacenComidaPiscifactoria(Piscifactoria piscifactoria) 72
comprarTanque(Piscifactoria piscifactoria) 72
aumentarCapacidadAlmacenCentral() 72
seleccionarTipoPiscifactoría() 72
calcularCostoPiscifactoría(int tipo) 72
numeroPiscifactoriasRio() 74
numeroPiscifactoriasMar() 74
mostrarEstadoTanque() 74
pasarDias() 74
anadirPezAleatorio() 74
repartirComida() 74
todasLasPiscifactoriasEnLaMediaComidaAnimal(int mediaCantidadComidaAnimal) 74
todasLasPiscifactoriasLlenasDeComidaAnimal() 75
mediaComidaAnimal() 75
repartirComidaAnimal() 75
todasLasPiscifactoriasEnLaMediaComidaVegetal(int mediaCantidadComidaVegetal) 75
todasLasPiscifactoriasLlenasDeComidaVegetal() 75
mediaComidaVegetal() 75
repartirComidaVegetal() 76
anadirMonedasOculto() 76
toString() 76
main(String[ ] args) 76
Interfaces **77**
Interfaz Mar **77**
Interfaz Río **77**

Componentes

Sistema de monedas

Componente que proporciona un sistema que gestiona las monedas de las que dispone el usuario.

Clases

SistemaMonedas

Clase pública que representa a un sistema en el que se dispone de una cierta cantidad de monedas que se puede establecer y obtener.

Atributos

private int monedas

Cantidad de monedas de las que se dispone.

Métodos

SistemaMonedas()

public SistemaMonedas()
Constructor de un objeto de la clase con una cantidad nula de monedas.

SistemaMonedas(int monedas)

public SistemaMonedas(int monedas)
Constructor de un objeto de la clase con una cantidad determinada de monedas.

Parámetros

monedas Cantidad de monedas con las que se inicializa el sistema de monedas.

getMonedas()

public int getMonedas()
Getter del atributo monedas.

devuelve Cantidad de monedas disponible en el sistema de monedas.

setMonedas(int monedas)

public void setMonedas(int monedas)
Setter del atributo monedas.

Parámetros

monedas Cantidad de monedas disponibles a establecer en el sistema de monedas.

toString()

public String toString()
Devuelve un string con información relevante del sistema de monedas. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante del sistema de monedas.

Generador de menús

Componente encargado de proporcionar herramientas para generar menús, este depende del componente del sistema de entrada.

Clases

GeneradorMenus

Clase pública que proporciona una serie de utilidades para la generación de menús.

Métodos

generarMenu(String[] opciones, int numeroOpcionInicial)

public static void generarMenu(String[] opciones, int numeroOpcionInicial)
Genera un menú con unas opciones.

Parámetros

opciones Opciones del menú a generar.
numeroOpcionInicial Número de la opción inicial del menú.

generarMenu(String[] cabecera, String[] opciones, int numeroOpcionInicial)

public static void generarMenu(String[] cabecera, String[] opciones, int numeroOpcionInicial)
Genera un menú con una cabecera y unas opciones.

Parámetros

cabecera Cabecera del menú a generar.
opciones Opciones del menú a generar.
numeroOpcionInicial Número de la opción inicial del menú.

generarMenuOperativo(String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal)

public static int generarMenuOperativo(String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal)
Genera un menú operativo con unas determinadas opciones.

Parámetros

opciones Opciones del menú a generar.
numeroOpcionInicial Número de la opción inicial del menú.
numeroOpcionFinal Número de la opción final del menú.

devuelve Opción seleccionada por el usuario.

generarMenuOperativo(String[] cabecera, String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal)

public static int generarMenuOperativo(String[] cabecera, String[] opciones, int numeroOpcionInicial, int numeroOpcionFinal)
Genera un menú operativo con una cabecera y unas opciones.

Parámetros

cabecera Cabecera del menú a generar.
opciones Opciones del menú a generar.
numeroOpcionInicial Número de la opción inicial del menú.
numeroOpcionFinal Número de la opción final del menú.

devuelve Opción seleccionada por el usuario.

toString()

public String toString()
Devuelve un string con información relevante de la clase. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante de la clase.

Sistema de entrada

Componente que se encarga de la gestión de la entrada de datos. Este es utilizado por el componente de generador de menús.

Clases

SistemaEntrada

Clase pública que se encarga de gestionar la entrada de datos introducidos por el usuario.

Atributos

private static final BufferedReader BUFER_ENTRADA

Búfer de entrada de datos introducidos por el usuario por teclado.

Métodos

entradaOpcionNumerica(int minimo, int maximo)

public static int entradaOpcionNumerica(int minimo, int maximo)
Gestiona la entrada de opciones numéricas siendo aceptables un cierto rango de opciones.

Parámetros

minimo Valor mínimo aceptado.
maximo Valor máximo aceptado.

devuelve Opción escogida por el usuario.

entradaOpcionNumerica(int[] opciones)

public static int entradaOpcionNumerica(int[] opciones)
Gestiona la entrada de opciones numéricas.

Parámetros

opciones Opciones numéricas aceptadas.

devuelve Opción numérica aceptada escogida por el usuario.

entradaTexto()

public static String entradaTexto()
Gestiona la entrada de texto introducido por el usuario.

devuelve Texto introducido por el usuario.

entradaOpcionNumericaEnteraPositiva()

public static int entradaOpcionNumericaEnteraPositiva()
Gestiona la entrada de un número entero positivo introducido por el usuario.

> **devuelve** Número entero positivo introducido por el usuario.

close()

public static void close()
Cierra el búfer de entrada de datos.

toString()

public String toString()
Devuelve un string con información relevante de la clase. Este método sobrescribe al método de la superclase Object.

> **devuelve** String con información relevante de la clase.

FechaTiempoLocal

Componente que contiene utilidades relativas a la fecha y a la hora.

Clases

FechaTiempoLocal

Clase pública que contiene utilidades relativas a la fecha y a la hora.

Métodos

obtenerFechaTiempoActual()

public static String obtenerFechaTiempoActual()
Permite obtener la fecha y tiempo del sistema en formato [aaaa-mm-dd hh:mm:ss].

> **devuelve** Fecha y tiempo del sistema en formato [aaaa-mm-dd hh:mm:ss].

SistemaFicheros

Componente que se encarga de la gestión del sistema de ficheros.

Clases

SistemaFicheros

Clase pública con utilidades relativas al sistema de ficheros.

Métodos

crearArchivo(String rutaArchivo)

public static void crearArchivo(String rutaArchivo) throws IOException
Crea un archivo en el sistema de ficheros a partir de su ruta.

Parámetros

rutaArchivo Ruta del archivo a crear.

Lanza IOException Cuando no se ha podido crear el archivo en el sistema de ficheros.

crearCarpeta(String rutaCarpeta)

public static void crearCarpeta(String rutaCarpeta) throws IOException
Crea una carpeta en el sistema de ficheros a partir de su ruta.

Parámetros

rutaCarpeta Ruta de la carpeta a crear.

Lanza IOException Cuando no se ha podido crear la carpeta en el sistema de ficheros.

crearEstructuraCarpetas(String estructuraCarpetas)

public static void crearEstructuraCarpetas(String estructuraCarpetas) throws IOException
Crea una estructura de carpetas en el sistema de ficheros a partir de la ruta de la estructura de carpetas.

Parámetros

estructuraCarpetas Ruta de las estructura de carpetas.

Lanza IOException Cuando no se ha podido crear la estructura de carpetas en el sistema de ficheros.

existeArchivo(String rutaArchivo)

public static boolean existeArchivo(String rutaArchivo) throws IOException
Permite comprobar la existencia de un archivo en el sistema de ficheros a partir de su ruta.

Parámetros

rutaArchivo Ruta del archivo a comprobar su existencia.

devuelve Si el archivo existe o no.

Lanza IOException Si el archivo del que se comprueba su existencia se trata de un directorio.

existeDirectorio(String rutaDirectorio)

public static boolean existeDirectorio(String rutaDirectorio) throws IOException
Permite comprobar la existencia de un directorio en el sistema de ficheros a partir de su ruta.

Parámetros

rutaDirectorio Ruta del directorio a comprobar su existencia.

devuelve Si el directorio existe o no.

Lanza IOException Si el directorio del que se comprueba su existencia se trata de un archivo.

isDirectorioVacio(String rutaDirectorio)

public static boolean isDirectorioVacio(String rutaDirectorio) throws IOException
Indica si un directorio del sistema de ficheros está vacío.

Parámetros

rutaDirectorio Ruta del directorio a comprobar si está vacío.

devuelve Si el directorio está vacío o no.

Lanza IOException Si el directorio no existe o si se trata de un fichero.

borrarArchivo(String rutaArchivo)

public static void borrarArchivo(String rutaArchivo) throws IOException
Permite borrar un archivo del sistema de ficheros.

Parámetros

rutaArchivo Ruta del archivo a borrar.

Lanza IOException Cuando el archivo no ha podido ser eliminado o no existe.

LecturaEscrituraFicherosPlanos

Componentes con utilidades para la escritura y lectura de ficheros de texto plano.

Clases

LecturaEscrituraFicherosPlanos

Clase con utilidades para la lectura y escritura de ficheros de texto plano.

Métodos

escrituraFicheroTextoPlano(File archivo, String texto, String codificacion)

public static void escrituraFicheroTextoPlano(File archivo, String texto, String codificacion) throws IOException
Permite la escritura de un texto en un archivo de texto plano.

Parámetros

archivo Archivo en el que se realizará la escritura del texto.
texto Texto que se escribirá en el archivo.
codificacion Codificación en la que se encuentra el texto a escribir en el fichero.

Lanza IOException Cuando surge un error al escribir en el archivo o cuando surge un error al cerrar el búfer de escritura.

escrituraFicheroTextoPlanoSinSobreescritura(File archivo, String texto, String codificacion)

public static void escrituraFicheroTextoPlanoSinSobreescritura(File archivo, String texto, String codificacion) throws IOException
Permite la escritura de un texto en un archivo de texto plano sin eliminar el contenido escrito anteriormente.

Parámetros

archivo Archivo en el que se realizará la escritura del texto.
texto Texto que se escribirá en el archivo.
codificacion Codificación en la que se encuentra el texto a escribir en el fichero.

Lanza IOException Cuando surge un error al escribir en el archivo o cuando surge un error al cerrar el búfer de escritura.

lecturaFicheroTextoPlano(File archivo, String codificacion)

public static String lecturaFicheroTextoPlano(File archivo, String codificacion) throws IOException
Permite la lectura de un fichero de texto plano.

Parámetros

archivo Archivo de texto plano del cuál se lee el texto.
codificacion Codificación del texto del archivo.

devuelve Texto que contiene el archivo de texto plano.

Lanza IOException Cuando surge un error al leer el texto o al cerrar el búfer de lectura.

LecturaEscrituraJSON

Componente encargado de la lectura y escritura de archivos JSON.

Clases

LecturaEscrituraJSON

Clase para gestionar la lectura y escritura de ficheros JSON.

Métodos

guardarJSON(File archivo, T objeto)

public static <T> void guardarJSON(File archivo, T objeto) throws IOException
Guarda un objeto en un archivo JSON.

Parámetros

archivo Archivo donde se guardará el objeto JSON.
objeto Objeto a guardar.

Lanza IOException Si ocurre un error al guardar el archivo.

cargarJSON(File archivo)

public static void <T> T cargarJSON(File archivo) throws IOException
Carga un objeto desde un archivo JSON.

Parámetros

archivo Archivo JSON desde donde se cargará el objeto.
<T> Tipo del objeto a cargar.
Class<T> Clase del objeto que se quiere cargar.

devuelve El objeto cargado.

lanza IOException Si ocurre un error al leer el archivo.

Clases propias

Clase Pez

Clase abstracta que representa la base de un pez del sistema.

Atributos

protected final String nombre

Nombre común del pez.

protected final String nombreCientifico

Nombre científico del pez.

protected int edad

Edad del pez en días.

protected final boolean sexo

Sexo del pez. True indica que el pez es hembra y false indica que el pez es macho.

protected boolean fertil

Indica si el pez es fértil y puede reproducirse.

protected boolean vivo

Indica si el pez está vivo.

protected boolean alimentado

Indica si el pez ha sido alimentado.

protected int diasSinReproducirse

Número de días en los que el pez no se ha reproducido desde la última vez que era fértil.

Métodos

getNombre()

public String getNombre()
Getter del nombre común del pez.

> **devuelve** Nombre común del pez.

getNombreCientifico()

public String getNombreCientifico()
Getter del nombre científico del pez.

devuelve Nombre científico del pez.

getEdad()

public int getEdad()
Getter de la edad del pez en días.

devuelve Edad del pez en días.

isSexo()

public boolean isSexo()
Getter del sexo del pez.

devuelve True si el pez es hembra y false si el pez es macho.

isFertil()

public boolean isFertil()
Getter que indica si el pez es fértil y puede reproducirse.

devuelve True si el pez es fértil y puede reproducirse.

isVivo()

public boolean isVivo()
Getter que indica si el pez está vivo o no.

devuelve True si el pez está vivo.

isAlimentado()

public boolean isAlimentado()
Getter que indica si el pez ha sido alimentado o no.

devuelve True si el pez ha sido alimentado.

getDiasSinReproducirse()

public int getDiasSinReproducirse()
Getter del número de días en los que el pez no se ha reproducido desde la última vez que ha sido fértil.

devuelve Número de días en los que el pez no se ha reproducido desde la última vez que ha sido fértil.

setEdad(int edad)

public void setEdad(int edad)
Setter de la edad del pez en días.

Parámetros
edad Edad del pez en días a establecer.

setFertil(boolean fertil)

public void setFertil(boolean fertil)
Setter de si el pez es fértil y puede reproducirse.

Parámetros
fertil Si es true se establece que el pez es fértil y puede reproducirse.

setVivo(boolean vivo)

public void setVivo(boolean vivo)
Setter de si el pez está vivo.

Parámetros
vivo Si es true se establece que el pez está vivo.

setAlimentado(boolean alimentado)

public void setAlimentado(boolean alimentado)
Setter de si el pez ha sido alimentado.

Parámetros
alimentado Si es true se establece que el pez ha sido alimentado.

setDiasSinReproducirse(int diasSinReproducirse)

public void setDiasSinReproducirse(int diasSinReproducirse)
Setter del número de días sin reproducirse desde que el pez era fértil.

Parámetros
diasSinReproducirse Número de días sin reproducirse desde que el pez era fértil.

Pez(String nombre, String nombreCientifico, boolean sexo)

protected Pez(String nombre, String nombreCientifico, boolean sexo)
Constructor de peces a partir de su nombre, nombre científico y sexo.

Parámetros
nombre Nombre del pez.
nombreCientifico Nombre científico del pez.
sexo Sexo del pez si es true el pez es hembra y si es false el pez es macho.

showStatus()

public abstract void showStatus()
Muestra el estado del pez.

grow()

public abstract void grow()
Hace que el pez crezca un día.

reset()

public void reset()
Reinicia el pez estableciendo sus atributos a sus estados iniciales.

comer()

public int comer()
Indica la cantidad de comida que consume el pez un determinado día.

devuelve Cantidad de comida que consume el pez un determinado día.

isMaduro()

public abstract boolean isMaduro()
Indica si el pez está maduro.

devuelve True si el pez está maduro.

isEdadOptima()

public abstract boolean isEdadOptima()
Indica si el pez está en la edad óptima para ser vendido.

devuelve True si el pez está en la edad óptima para ser vendido.

obtenerPezHijo()

public abstract Pez obtenerPezHijo()
Devuelve un pez recién nacido de sexo masculino.

devuelve Pez recién nacido de sexo masculino.

obtenerPezHija()

public abstract Pez obtenerPezHija()
Devuelve un pez recién nacido de sexo femenino.

devuelve Pez recién nacido de sexo femenino.

String toString()

public String toString()
Devuelve un string con información relevante del pez. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante del pez.

Peces

Clase Carnívoro

Clase abstracta pública que extiende de Pez la cual representa un pez con tipo de alimentación carnívora.

Métodos

Carnivoro(String nombre, String nombreCientifico, boolean sexo)

protected Carnivoro(String nombre, String nombreCientifico, boolean sexo)
Constructor de la clase carnívoro el cual llama al constructor de la superclase.

Parámetros

nombre Nombre del pez.
nombreCientifico Nombre científico del pez.
sexo Sexo del pez si es true es hembra y si es false es macho.

Clase Filtrador

Clase abstracta pública que extiende de Pez la cual representa un pez con un tipo de alimentación herbívora.

Métodos

Filtrador(String nombre, String nombreCientifico, boolean sexo)

protected Filtrador(String nombre, String nombreCientifico, boolean sexo)
Constructor de la clase filtrador el cual llama a los parámetros de la superclase.

Parámetros

nombre Nombre del pez.
nombreCientifico Nombre científico del pez.
sexo Sexo del pez si es true es hembra y si es false es macho.

comer()

public int comer()
Método que devuelve la cantidad de comida que consumirá el pez.

devuelve La cantidad de comida que consumirá.

Clase Omnívoro

Clase abstracta pública que extiende de Pez la cual representa un pez con un tipo de alimentación omnívora.

Métodos

Omnivoro(String nombre, String nombreCientifico, boolean sexo)

protected Omnivoro(String nombre, String nombreCientifico, boolean sexo)
Constructor de la clase omnívoro el cual llama al constructor de la superclase.

Parámetros

nombre Nombre del pez.
nombreCientifico Nombre científico del pez.
sexo Sexo del pez si es true es hembra y si es false es macho.

comer()

public int comer()
Método que devuelve la cantidad de comida que consumirá el pez.

devuelve La cantidad de comida que consumirá.

Clase Caballa

Clase pública que hereda de la clase abstracta Carnivoro, implementa la interfaz bandera Mar y representa a una caballa.

Métodos

Caballa(boolean sexo)

public Caballa(boolean sexo)
Constructor de caballas. Este llama al constructor de la superclase.

Parámetros

sexo Sexo de la caballa si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado de la caballa. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que la caballa crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si la caballa está madura.

devuelve True si la caballa está madura.

isEdadOptima()

public boolean isEdadOptima()
Indica si la caballa está en la edad óptima para ser vendida.

devuelve True si la caballa está en la edad óptima para ser vendida.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve una caballa recién nacida de sexo masculino.

devuelve Caballa recién nacida de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve una caballa recién nacida de sexo femenino.

devuelve Caballa recién nacida de sexo femenino.

Clase Pejerrey

Clase pública que hereda de la clase abstracta Carnivoro, implementa la interfaz bandera Rio y representa a un pejerrey.

Métodos

Pejerrey(boolean sexo)

public Pejerrey(boolean sexo)
Constructor de pejerreyes. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del pejerrey si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del pejerrey. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el pejerrey crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el pejerrey está maduro

devuelve True si el pejerrey está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el pejerrey está en la edad óptima para ser vendido.

devuelve True si el pejerrey está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un pejerrey recién nacido de sexo masculino.

devuelve Pejerrey recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un pejerrey recién nacido de sexo femenino.

devuelve Pejerrey recién nacido de sexo femenino.

Clase PercaEuropea

Clase pública que hereda de la clase abstracta Carnivoro, implementa la interfaz bandera Rio y representa a una perca europea.

Métodos

PercaEuropea(boolean sexo)

public PercaEuropea(boolean sexo)
Constructor de percas europeas. Este llama al constructor de la superclase.

Parámetros

sexo Sexo de la perca europea si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado de la perca europea. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que la perca europea crezca un día. Este método sobrescribe el método abstracto de la superclase.

comer()

public int comer()
Indica la cantidad de alimento que consume la perca europea en un día.

devuelve Cantidad de alimento que consume la perca europa en un día.

isMaduro()

public boolean isMaduro()
Indica si la perca europea está madura.

devuelve True si la perca europea está madura.

isEdadOptima()

public boolean isEdadOptima()
Indica si la perca europea está en la edad óptima para ser vendida.

devuelve True si la perca europea está en la edad óptima para ser vendida.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve una perca europea recién nacida de sexo masculino.

devuelve Perca europea recién nacida de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve una perca europea recién nacida de sexo femenino.

devuelve Perca europea recién nacida de sexo femenino.

Clase Robalo

Clase pública que hereda de la clase abstracta Carnivoro, implementa la interfaz bandera Mar y representa a un róbalo.

Métodos

Robalo(boolean sexo)

public Robalo(boolean sexo)
Constructor de róbalos. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del róbalo si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del róbalo. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el róbalo crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el róbalo está maduro.

devuelve True si el róbalo está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el róbalo está en la edad óptima para ser vendido.

devuelve True si el róbalo está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un róbalo recién nacido de sexo masculino.

devuelve Róbalo recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un róbalo recién nacido de sexo femenino.

devuelve Róbalo recién nacido de sexo femenino.

Clase SalmonAtlantico

Clase pública que hereda de la clase abstracta Carnivoro, implementa las interfaces bandera Mar y Rio y representa a un salmón atlántico.

Métodos

SalmonAtlantico(boolean sexo)

public SalmonAtlantico(boolean sexo)
Constructor de salmones atlánticos. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del salmón atlántico si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del salmón atlántico. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el salmón atlántico crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el salmón atlántico está maduro.

devuelve True si el salmón atlántico está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el salmón atlántico está en la edad óptima para ser vendido.

devuelve True si el salmón atlántico está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un salmón atlántico recién nacido de sexo masculino.

devuelve Salmón atlántico recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un salmón atlántico recién nacido de sexo femenino.

devuelve Salmón atlántico recién nacido de sexo femenino.

Clase SalmonChinook

Clase pública que hereda de la clase abstracta Carnivoro, implementa la interfaz bandera Rio y representa a un salmón chinook.

Métodos

SalmonChinook(boolean sexo)

public SalmonChinook(boolean sexo)
Constructor de salmones chinook. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del salmón chinook si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del salmón chinook. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el salmón chinook crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el salmón chinook está maduro.

devuelve True si el salmón chinook está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el salmón chinook está en la edad óptima para ser vendido.

devuelve True si el salmón chinook está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un salmón chinook recién nacido de sexo masculino.

devuelve Salmón chinook recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un salmón chinook recién nacido de sexo femenino.

devuelve Salmón chinook recién nacido de sexo femenino.

Clase ArenqueDelAtlantico

Clase pública que hereda de la clase abstracta Filtrador, implementa la interfaz bandera Mar y representa a un arenque del atlántico.

Métodos

ArenqueDelAtlantico(boolean sexo)

public ArenqueDelAtlantico(boolean sexo)
Constructor de arenques del atlántico. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del arenque del atlántico si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del arenque del atlántico. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el arenque del atlántico crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el arenque del atlántico está maduro.

devuelve True si el arenque del atlántico está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el arenque del atlántico está en la edad óptima para ser vendido.

devuelve True si el arenque del atlántico está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un arenque del atlántico recién nacido de sexo masculino.

devuelve Arenque del atlántico recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un arenque del atlántico recién nacido de sexo femenino.

devuelve Arenque del atlántico recién nacido de sexo femenino.

Clase TilapiaDelNilo

Clase pública que hereda de la clase abstracta Filtrador, implementa la interfaz bandera Rio y representa a una tilapia del nilo.

Métodos

TilapiaDelNilo(boolean sexo)

public TilapiaDelNilo(boolean sexo)
Constructor de tilapias del nilo. Este llama al constructor de la superclase.

Parámetros

sexo Sexo de la tilapia del nilo si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado de la tilapia del nilo. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que la tilapia del nilo crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si la tilapia del nilo está madura.

devuelve True si la tilapia del nilo está madura.

isEdadOptima()

public boolean isEdadOptima()
Indica si la tilapia del nilo está en la edad óptima para ser vendida.

devuelve True si la tilapia del nilo está en la edad óptima para ser vendida.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve una tilapia del nilo recién nacida de sexo masculino.

devuelve Tilapia del nilo recién nacida de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve una tilapia del nilo recién nacida de sexo femenino.

devuelve Tilapia del nilo recién nacida de sexo femenino.

Clase Abadejo

Clase pública que hereda de la clase abstracta Omnivoro, implementa la interfaz bandera Mar y representa a un abadejo.

Métodos

Abadejo(boolean sexo)

public Abadejo(boolean sexo)
Constructor de abadejos. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del abadejo si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del abadejo. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el abadejo crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el abadejo está maduro.

devuelve True si el abadejo está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el abadejo está en la edad óptima para ser vendido.

devuelve True si el abadejo está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un abadejo recién nacido de sexo masculino.

devuelve Abadejo recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un abadejo recién nacido de sexo femenino.

devuelve Abadejo recién nacido de sexo femenino.

Clase CarpinTresEspinas

Clase pública que hereda de la clase abstracta Omnivoro, implementa la interfaz bandera Rio y representa a un carpín de tres espinas.

Métodos

CarpinTresEspinas(boolean sexo)

public CarpinTresEspinas(boolean sexo)
Constructor de carpines de tres espinas. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del carpín de tres espinas si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del capín de tres espinas. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el carpín de tres espinas crezca un día. Este método sobrescribe el método abstracto de la superclase.

comer()

public int comer()
Indica la cantidad de comida que consume el carpín de tres espinas un determinado día.

devuelve Cantidad de comida que consume el carpín de tres espinas un determinado día.

isMaduro()

public boolean isMaduro()
Indica si el carpín de tres espinas está maduro.

devuelve True si el carpín de tres espinas está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el carpín de tres espinas está en la edad óptima para ser vendido.

devuelve True si el capín de tres espinas está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un carpín de tres espinas recién nacido de sexo masculino.

devuelve Carpín de tres espinas recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un carpín de tres espinas recién nacido de sexo femenino.

devuelve Carpín de tres espinas recién nacido de sexo femenino.

Clase Dorada

Clase pública que hereda de la clase abstracta Omnivoro, implementa las interfaces bandera Rio y Mar y representa a una dorada.

Métodos

Dorada(boolean sexo)

public Dorada(boolean sexo)
Constructor de doradas. Este llama al constructor de la superclase.

Parámetros

sexo Sexo de la dorada si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado de la dorada. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que la dorada crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si la dorada está madura.

devuelve True si la dorada está madura.

isEdadOptima()

public boolean isEdadOptima()
Indica si la dorada está en la edad óptima para ser vendida.

devuelve True si la dorada está en la edad óptima para ser vendida.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve una dorada recién nacida de sexo masculino.

devuelve Dorada recién nacida de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve una dorada recién nacida de sexo femenino.

devuelve Dorada recién nacida de sexo femenino.

Clase Sargo

Clase pública que hereda de la clase abstracta Omnivoro, implementa la interfaz bandera Mar y representa a un sargo.

Métodos

Sargo(boolean sexo)

public Sargo(boolean sexo)
Constructor de sargos. Este llama al constructor de la superclase.

Parámetros

sexo Sexo del sargo si es true es hembra y si es false es macho.

showStatus()

public void showStatus()
Muestra el estado del sargo. Este método sobrescribe el método abstracto de la superclase.

grow()

public void grow()
Hace que el sargo crezca un día. Este método sobrescribe el método abstracto de la superclase.

isMaduro()

public boolean isMaduro()
Indica si el sargo está maduro.

devuelve True si el sargo está maduro.

isEdadOptima()

public boolean isEdadOptima()
Indica si el sargo está en la edad óptima para ser vendido.

devuelve True si el sargo está en la edad óptima para ser vendido.

obtenerPezHijo()

public Pez obtenerPezHijo()
Devuelve un sargo recién nacido de sexo masculino.

devuelve Sargo recién nacido de sexo maculino.

obtenerPezHija()

public Pez obtenerPezHija()
Devuelve un sargo recién nacido de sexo femenino.

devuelve Sargo recién nacido de sexo femenino.

AlmacenCentral

Clase pública que representa a un almacén central que almacena comida de dos tipos (animal y vegetal) y la distribuye equitativamente entre las piscifactorías.

Atributos

private int capacidadComida

Capacidad máxima de comida para cada tipo del almacén central.

private int cantidadComidaAnimal

Cantidad de comida animal disponible en el almacén central.

private int cantidadComidaVegetal

Cantidad de comida vegetal disponible en el almacén central.

Métodos

AlmacenCentral()

public AlmacenCentral()
Constructor de almacenes centrales que inician con una capacidad de 200 para comida animal y vegetal.

getCapacidadComida()

public int getCapacidadComida()
Permite obtener la capacidad máxima de comida para cada tipo del almacén central.

devuelve Capacidad máxima de comida para cada tipo del almacén central.

setCapacidadComida(int capacidadComida)

public void setCapacidadComida(int capacidadComida)
Permite establecer la capacidad máxima de cada tipo de comida del almacén central.

Parámetros
capacidadComida Capacidad máxima de cada tipo de comida a establecer.

getCantidadComidaAnimal()

public int getCantidadComidaAnimal()
Permite obtener la cantidad de comida animal disponible en el almacén central.

devuelve Cantidad de comida animal disponible en el almacén central.

setCantidadComidaAnimal(int cantidadComidaAnimal)

public void setCantidadComidaAnimal(int cantidadComidaAnimal)
Permite establecer la cantidad de comida animal disponible en el almacén central.

Parámetros
cantidadComidaAnimal Cantidad de comida animal disponible a establecer.

getCantidadComidaVegetal()

public int getCantidadComidaVegetal()
Permite obtener la cantidad de comida vegetal disponible en el almacén central.

devuelve Cantidad de comida vegetal disponible en el almacén central.

setCantidadComidaVegetal(int cantidadComidaVegetal)

public void setCantidadComidaVegetal(int cantidadComidaVegetal)
Permite establecer la cantidad de comida vegetal disponible en el almacén central.

Parámetros
cantidadComidaVegetal Cantidad de comida vegetal disponible a establecer.

mejorar()

public void mejorar()
Mejora el almacén central aumentando la capacidad de ambas comidas en 50 unidades.

toString()

public String toString()
Devuelve un string con información relevante del almacén central. Este método sobrescribe al método de la superclase Object.

devuelve Información relevante del almacén central.

Clase Tanque

Clase pública que representa a un tanque que contiene un número de peces. Este hace uso de la clase Pez y de la clase pública estática interna AlmacenComida de la clase Piscifactoria.

Atributos

private final int numeroTanque

Número del tanque de la piscifactoría.

private int capacidadMaximaPeces

Capacidad máxima de peces que puede tener el tanque.

private ArrayList<Pez> peces

Peces del tanque.

Métodos

getCapacidadMaximaPeces()

public int getCapacidadMaximaPeces()
Permite obtener la capacidad máxima de peces que puede tener el tanque.

devuelve Capacidad máxima de peces que puede tener el tanque.

setCapacidadMaximaPeces(int capacidadMaximaPeces)

public void setCapacidadMaximaPeces(int capacidadMaximaPeces)
Permite establecer la capacidad máxima de peces que puede tener el tanque.

Parámetros

capacidadMaximaPeces Capacidad máxima de peces que puede tener el tanque a establecer.

getPeces()

public ArrayList<Pez> getPeces()
Permite obtener los peces del tanque.

devuelve Peces del tanque.

setPeces(ArrayList<Pez> peces)

public void setPeces(ArrayList<Pez> peces)
Permite establecer los peces del tanque.

Parámetros

peces Peces del tanque a establecer.

getNumeroTanque()

public int getNumeroTanque()
Permite obtener el número del tanque en la piscifactoría.

devuelve Número del tanque en la piscifactoría.

Tanque(int numeroTanque, int capacidadMaximaPeces)

public Tanque(int numeroTanque, int capacidadMaximaPeces)
Constructor de tanques.

Parámetros
numeroTanque Número del tanque en la piscifactoría.
capacidadMaximaPeces Capacidad máxima de peces del tanque.

showStatus()

public void showStatus()
Imprime el estado del tanque por pantalla.

pecesVivos()

public int pecesVivos()
Devuelve el número de peces vivos en el tanque.

devuelve Número de peces vivos en el tanque.

pecesAlimentados()

public int pecesAlimentados()
Devuelve el número de peces alimentados en el tanque.

devuelve Número de peces alimentados en el tanque.

pecesAdultos()

public int pecesAdultos()
Devuelve el número de peces adultos en el tanque.

devuelve Número de peces adultos en el tanque.

pecesAdultosVivos()

public int pecesAdultosVivos()
Devuelve el número de peces adultos vivos en el tanque.

devuelve Número de peces adultos vivos en el tanque.

pecesMacho()

public int pecesMacho()
Devuelve el número de peces macho en el tanque.

devuelve Número de peces macho en el tanque.

pecesHembra()

public int pecesHembra()
Devuelve el número de peces hembra en el tanque.

devuelve Número de peces hembra en el tanque.

pecesFertiles()

public int pecesFertiles()
Devuelve el número de peces fértiles en el tanque.

devuelve Número de peces fértiles en el tanque.

showFishStatus()

public void showFishStatus()
Muestra el estado de cada pez del tanque por pantalla.

showCapacity(String piscifactoria)

public void showCapacity(String piscifactoria)
Imprime el porcentaje de llenado del tanque por pantalla.

Parámetros
piscifactoria Nombre de la piscifactoría en la que se sitúa el tanque.

alimentar(Piscifactoria.AlmacenComida almacenComida)

public void alimentar(Piscifactoria.AlmacenComida almacenComida)
Gestiona la lógica para alimentar a los peces.

Parámetros
almacenComida Almacén de comida de la piscifactoría donde se sitúa el tanque.

alimentarAleatorio(ArrayList<Integer> cantidadDeComidaNecesariaPorPez, Piscifactoria.AlmacenComida almacenComida, int comidaDisponible)

private void alimentarAleatorio(ArrayList<Integer> cantidadDeComidaNecesariaPorPez, Piscifactoria.AlmacenComida almacenComida, int comidaDisponible)
Gestiona la lógica de alimentación de los peces cuando la comida es insuficiente para alimentar a todos los peces.

Parámetros

cantidadDeComidaNecesariaPorPez Cantidad de comida que necesita cada pez para alimentarse.
almacenComida Almacén de comida de la piscifactoría donde se sitúa el tanque.
comidaDisponible Comida de la que se dispone para alimentar a los peces.

hayMachoFertil()

private boolean hayMachoFertil()
Indica si hay un macho fértil en la piscifactoría.

devuelve True si hay un macho fértil en la piscifactoría.

reproducir()

private void reproducir()
Gestiona la lógica de reproducción de los peces del tanque.

venderPecesOptimos()

private void venderPecesOptimos()
Vende todos los peces que se encuentran en la edad óptima para ser vendidos.

venderPeces()

public void venderPeces()
Vende todos los peces que hayan llegado a la madurez.

eliminarPecesMuertos()

public void eliminarPecesMuertos()
Elimina los peces muertos del tanque.

nextDay()

public int nextDay()
Realiza la lógica de que ha pasado un día haciendo crecer a los peces, reproduciendolos y vendiendo los peces que se encuentren en edad óptima.

devuelve Peces vendidos.

vaciarTanque()

public void vaciarTanque()

Elimina todos los peces de un tanque, independientemente de si están vivos o muertos y de su edad.

toString()

public String toString()
Devuelve un string con información relevante del tanque. Este método sobrescribe al método de la superclase Object.

> **devuelve** String con información relevante del tanque.

Clase Piscifactoría

Clase pública abstracta que representa a una piscifactoría con múltiples tanques de peces. Esta hace uso de la clase tanque.

Clases

AlmacenComida

Clase interna de la clase Piscifactoría que representa a un almacén de comida de una piscifactoría.

Atributos

private int capacidadMaximaComida

Capacidad máxima de cada tipo de comida en el almacén.

private int cantidadComidaAnimal

Cantidad de comida animal que hay en el almacén.

private int cantidadComidaVegetal

Cantidad de comida vegetal que hay en el almacén.

Métodos

AlmacenComida(int capacidadMaximaComida, int cantidadComidaAnimal, int cantidadComidaVegetal)

public AlmacenComida(int capacidadMaximaComida, int cantidadComidaAnimal, int cantidadComidaVegetal)
Constructor parametrizado de almacenes de comida.

Parámetros

capacidadMaximaComida Capacidad máxima de comida de cada tipo.
cantidadComidaAnimal Cantidad de comida animal del almacén.
cantidadComidaVegetal Cantidad de comida vegetal del almacén.

getCapacidadMaximaComida()

public int getCapacidadMaximaComida()
Permite obtener la capacidad máxima de comida de cada tipo del almacén.

devuelve Capacidad máxima de comida de cada tipo del almacén.

setCapacidadMaximaComida(int capacidadMaximaComida)

public void setCapacidadMaximaComida(int capacidadMaximaComida)
Permite establecer la capacidad máxima de comida de cada tipo del almacén.

Parámetros

capacidadMaximaComida Capacidad máxima de comida de cada tipo de almacén a establecer.

getCantidadComidaAnimal()

public int getCantidadComidaAnimal()
Permite obtener la cantidad de comida animal del almacén.

devuelve Cantidad de comida animal del almacén.

setCantidadComidaAnimal(int cantidadComidaAnimal)

public void setCantidadComidaAnimal(int cantidadComidaAnimal)
Permite establecer la cantidad de comida animal del almacén.

Parámetros

cantidadComidaAnimal Cantidad de comida animal del almacén a establecer.

getCantidadComidaVegetal()

public int getCantidadComidaVegetal()
Permite obtener la cantidad de comida vegetal del almacén.

devuelve Cantidad de comida vegetal del almacén.

setCantidadComidaVegetal(int cantidadComidaVegetal)

public void setCantidadComidaVegetal(int cantidadComidaVegetal)
Permite establecer la cantidad de comida vegetal del almacén.

Parámetros

cantidadComidaVegetal Cantidad de comida vegetal del almacén a establecer.

mejorar(int capacidadAAumentar)

public void mejorar(int capacidadAAumentar)
Mejora el almacén aumentando la capacidad máxima de comida de cada tipo.

Parámetros

capacidadAAumentar Unidades a aumentar la capacidad máxima de comida de cada tipo.

toString()

public String toString()
Devuelve un string con información relevante del almacén.

devuelve String con información relevante del almacén.

Atributos

protected ArrayList<Tanque> tanques

Tanques de la piscifactoría.

protected Tanque tanqueInicial

El tanque con el que empieza obligatoriamente la piscifactoría.

protected AlmacenComida almacenInicial

Almacén de comida de la piscifactoría.

protected String nombre

El nombre de la piscifactoría.

Métodos

Piscifactoria(String nombre)

protected Piscifactoria(String nombre)
Constructor de la clase Piscifactoría el cual establece el nombre de la piscifactoría.

Parámetros
nombre nombre de la piscifactoría.

getTanques()

public ArrayList<Tanque> getTanques()
Devuelve los tanques.

devuelve Tanques de la piscifactoría.

setTanques(ArrayList<Tanque> tanques)

public void setTanques(ArrayList<Tanque> tanques)
Permite establecer los tanques de la piscifactoría.

Parámetros
tanques Tanques a establecer.

getTanqueInicial()

public Tanque getTanqueInicial()
Devuelve el tanque inicial de la piscifactoría.

devuelve Tanque inicial de la piscifactoría.

setTanqueInicial(Tanque tanqueInicial)

public void setTanqueInicial(Tanque tanqueInicial)
Permite establecer el tanque inicial de la piscifactoría.

Parámetros
tanqueInicial Tanque inicial a establecer.

getNombre()

public String getNombre()
Devuelve el nombre de la piscifactoría.

devuelve Nombre de la piscifactoría.

getAlmacenInicial()

public AlmacenComida getAlmacenInicial()
Devuelve el almacén de comida de la piscifactoría.

devuelve Almacén de comida de la piscifactoría.

setAlmacenInicial(AlmacenComida almacenInicial)

public void setAlmacenInicial(AlmacenComida almacenInicial)
Permite establecer el almacén de la piscifactoría.

Parámetros
almacenInicial Almacén inicial de la piscifactoría a establecer.

setNombre(String nombre)

public void setNombre(String nombre)
Permite establecer el nombre de la piscifactoría.

Parámetros
nombre Nombre de la piscifactoría a establecer.

showStatus()

public void showStatus()
Imprime el estado de la piscifactoría.

datosTanques()

private String datosTanques()
Devuelve el resto del texto, datos y cálculos de showStatus().

> **devuelve** Texto y cálculos necesarios para mostrar el estado de la piscifactoría en el método showStatus().

showTankStatus()

public void showTankStatus()
Imprime el estado de los tanques de la piscifactoría.

showFishStatus()

public void showFishStatus()
Imprime el estado de los peces de los tanques.

showCapacity()

public void showCapacity()
Imprime la capacidad de los tanques.

showFood()

public void showFood()
Imprime el estado del almacén

sellFish()

public void sellFish()
Vende todos los peces maduros y vivos de los tanques.

upgradeFood()

public abstract void upgradeFood()
Mejora el almacén e imprime en qué cantidad se ha mejorado.

nextDay()

public int nextDay()
Pasa un día en la piscifactoría realizando toda la lógica relacionada.

> **devuelve** Número de peces vendidos cuando pasa el día.

getPecesVivos()

public int getPecesVivos()
Devuelve el número de peces vivos en la piscifactoría.

> **devuelve** Número de peces vivos.

getPecesTotales()

public int getPecesTotales()
Devuelve el número de peces totales de la piscifactoría.

devuelve Número de peces totales.

getEspacioPeces()

public int getEspacioPeces()
Devuelve el espacio total de peces en la piscifactoría.

devuelve Espacio total de peces en la piscifactoría.

getIndiceTanqueVacio()

public int getIndiceTanqueVacio()
Permite obtener el índice del ArrayList de un tanque vacío de la piscifactoría.

devuelve Índice del ArrayList de un tanque vacío de la piscifactoría o -1 sino hay ningún tanque vacío.

getIndiceTanqueConEspacioParaPez(String nombrePez)

public int getIndiceTanqueConEspacioParaPez(String nombrePez)
Permite obtener el índice del ArrayList de un tanque que tiene un espacio para un pez en concreto.

Parámetros

nombrePez Nombre del pez del cual se quiere obtener el índice del tanque que tiene espacio para él.

devuelve Índice del tanque que tiene un espacio para un pez en concreto o -1 si no hay un tanque con un espacio para el pez en concreto.

isTodosLosTanqueLlenos()

public boolean isTodosLosTanqueLlenos()
Indica si todos los tanques de la piscifactoría están llenos.

devuelve True si todos los tanques de la piscifactoría están llenos.

toString()

public String toString()
Devuelve un string con información relevante de la piscifactoría. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante de la piscifactoría.

Clase PiscifactoriaRio

Clase pública que representa a una piscifactoría de río y que hereda de la clase Piscifactoria.

Métodos

PiscifactoriaRio(String nombre)

public PiscifactoriaRio(String nombre)
Constructor de piscifactorías de río el cual establece el nombre de la piscifactoría. Este hace uso del constructor de la superclase,

Parámetros

nombre Nombre de la piscifactoría de río.

upgradeFood()

public void upgradeFood()
Mejora el almacén de comida aumentando en 25 unidades la capacidad máxima e imprime en qué cantidad se ha mejorado.

toString()

public String toString()
Devuelve un string con información relevante de la piscifactoría de río. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante de la piscifactoría de río.

Clase PiscifactoriaMar

Clase pública que representa a una piscifactoría de mar y que hereda de la clase Piscifactoria.

Métodos

PiscifactoriaMar(String nombre)

public PiscifactoriaMar(String nombre)
Constructor de piscifactorías de mar el cual establece el nombre de la piscifactoría. Esta hace uso del constructor de la superclase.

Parámetros

nombre Nombre de la piscifactoría de mar.

upgradeFood()

public void upgradeFood()
Mejora el almacén de comida aumentando en 100 unidades la capacidad máxima e imprime en qué cantidad se ha mejorado.

toString()

public String toString()
Devuelve un string con información relevante de la piscifactoría de mar. Este método sobrescribe al método de la superclase Object.

devuelve String con información relevante de la piscifactoría de mar.

Clase Simulador

Clase pública que representa a un simulador de piscifactoría esta hace uso de todas las otras clases del sistema y contiene el main.

Atributos

private static int diasPasados

Indica el número de días que han pasado en la simulación.

public static ArrayList<Piscifactoria> piscifactorias

Piscifactoría de las que se dispone en la simulación.

private static String nombre

Nombre de la entidad, empresa o partida de la simulación.

public static SistemaMonedas sistemaMonedas

Sistema de monedas usado en la simulación.

public static AlmacenCentral almacenCentral

Almacén central de comida usado en la simulación.

public static Estadisticas estadisticas

Estadísticas de los peces de la simulación.

public static final File archivoLogsGeneral

Archivo de *logs* donde se registran todos los errores que hayan sucedido en cualquier partida.

public static File archivoLogPartida

Archivo de *log* de la partida.

public static File archivoTranscripcionesPartida

Archivo de transcripciones de la partida.

public static File archivoGuardadoPartida

Archivo de guardado de la partida.

Métodos

init()

private static void init()
Inicializa la simulación con unos datos solicitados al usuario.

menu()

private static void menu()
Imprime por pantalla el menú principal.

menuPisc()

private static void menuPisc()
Imprime por pantalla un menú para seleccionar una piscifactoría de la simulación.

selectPisc()

private static int selectPisc()
Permite seleccionar una piscifactoría del menú de piscifactorías.

devuelve Opción seleccionada.

selectTank(Piscifactoria piscifactoria)

private static int selectTank(Piscifactoria piscifactoria)
Muestra el menú de tanque de una piscifactoría y permite al usuario seleccionar uno.

Parámetros

piscifactoria Piscifactoría de la cuál se muestra el menú del tanque.

devuelve Índice del tanque seleccionado.

showGeneralStatus()

private static void showGeneralStatus()
Imprime por pantalla el estado general de la simulación.

showSpecificStatus()

private static void showSpecificStatus()
Muestra por pantalla el estado de todos los tanques de una piscifactoría seleccionada por el usuario.

showTankStatus(Piscifactoria piscifactoria)

private static void showTankStatus(Piscifactoria piscifactoria)
Muestra un menú para que el usuario seleccione el tanque de una piscifactoría y muestra el estado del tanque seleccionado.

Parámetros

piscifactoria Piscifactoría de la que se muestra el estado del tanque.

showStats()

private static void showStats()
Muestra un desglose de las estadísticas por cada tipo de pez.

showIctio()

private static void showIctio()
Muestra la información relativa a un pez seleccionado por el usuario.

nextDay()

private static void nextDay()
Avanza un día en todas las piscifactorías y muestra el número de peces vendidos y las monedas ganadas con ello.

addFood()

private static void addFood()
Gestiona la lógica de compra de comida para el almacén central o para una piscifactoría.

menuTipoComida()

private static int menuTipoComida()
Muestra un menú para que el usuario escoja el tipo de comida.

devuelve Opción seleccionada por el usuario.

menuCantidadComida()

private static int menuCantidadComida()
Muestra un menú para seleccionar la cantidad de comida a añadir.

devuelve Opción seleccionada por el usuario.

calcularCosto(int cantidad)

private static int calcularCosto(int cantidad)
Calcula el costo de la comida a añadir, aplicando descuentos cada 25 unidades.

Parámetros
cantidad Cantidad de comida a añadir.

devuelve Costo total en monedas.

addFish()

private static void addFish()
Añade un pez a una piscifactoría seleccionada por el usuario.

addFishMar(Piscifactoria piscifactoria)

private static void addFishMar(Piscifactoria piscifactoria)
Gestiona la lógica de añadir un pez a una piscifactoría de mar.

Parámetros
piscifactoria Piscifactoría donde se va a añadir el pez.

addFishRio(Piscifactoria piscifactoria)

private static void addFishRio(Piscifactoria piscifactoria)
Gestiona la lógica de añadir un pez a una pisicfactoría de río.

Parámetros
piscifactoria Piscifactoría donde se va a añadir el pez.

crearPezMar(int pez, boolean sexo)

private static Pez crearPezMar(int pez, boolean sexo)
Crea un pez que puede vivir en una piscifactoría de mar.

Parámetros
pez Código numérico del pez a crear.
sexo Sexo del pez a crear.

devuelve Pez que puede vivir en una piscifactoría de mar creado.

crearPezRio(int pez, boolean sexo)

private static Pez crearPezRio(int pez, boolean sexo)
Crea un pez que puede vivir en una piscifactoría de río.

Parámetros
pez Código numérico del pez a crear.
sexo Sexo del pez a crear.

devuelve Pez que puede vivir en una piscifactoría de río creado.

mostrarInformacionPez(PecesDatos datosDelPez)

private static void mostrarInformacionPez(PecesDatos datosDelPez)
Imprime por pantalla los datos relativos a un pez.

Parámetros
datosDelPez Datos relativos a un pez a mostrar.

sell()

private static void sell()
Vende todos los peces adultos que estén vivos en una piscifactoría seleccionada.

cleanTank()

private static void cleanTank()
Elimina todos los peces muertos de una piscifactoría seleccionada.

emptyTank()

private static void emptyTank()
Elimina todos los peces de un tanque, estén vivos o muertos.

upgrade()

private static void upgrade()
Muestra un menú para hacer mejoras como comprar o mejorar un edificio.

comprarEdificio()

private static void comprarEdificio()
Muestra un menú para comprar edificios.

comprarPiscifactoria()

private static void comprarPiscifactoria()
Permite al usuario comprar una nueva piscifactoría.

mejorarEdificio()

private static void mejorarEdificio()
Muestra un menú para mejorar edificios existentes.

mejorarPiscifactoria()

private static void mejorarPiscifactoria()
Permite al usuario mejorar una piscifactoría seleccionada.

mejorarAlmacenComidaPiscifactoria(Piscifactoria piscifactoria)

private static void mejorarAlmacenComidaPiscifactorio(Piscifactoria piscifactoria)
Gestiona la lógica de mejorar el almacén de comida de una piscifactoría.

Parámetros

piscifactoria Piscifactoría de la que se mejora el almacén de comida.

comprarTanque(Piscifactoria piscifactoria)

private static void comprarTanque(Piscifactoria piscifactoria)
Permite al usuario comprar un tanque para una piscifactoría.

Parámetros

piscifactoria Piscifactoría en la que se compra el tanque.

aumentarCapacidadAlmacenCentral()

private static void aumentarCapacidadAlmacenCentral()
Aumenta la capacidad del almacén central.

seleccionarTipoPiscifactoría()

private static int seleccionarTipoPiscifactoria()
Muestra un menú para seleccionar el tipo de piscifactoría

devuelve Opción seleccionada por el usuario.

calcularCostoPiscifactoría(int tipo)

private static int calcularCostoPiscifactoria(int tipo)
Calcula el costo de adquirir una piscifactoría en función de si es de mar o de río.

Parámetros

tipo Tipo de la piscifactoría si es de mar o de río.

devuelve Costo de adquirir la piscifactoría.

numeroPiscifactoriasRio()

private static int numeroPiscifactoriaRio()
Permite obtener el número de piscifactorías de río de la simulación.

devuelve Número de piscifactorías de río de la simulación.

numeroPiscifactoriasMar()

private static int numeroPiscifactoriaMar()
Permite obtener el número de piscifactorías de mar de la simulación.

devuelve Número de piscifactorías de mar de la simulación.

mostrarEstadoTanque()

private static void mostrarEstadoTanque()
Gestiona la lógica para mostrar un tanque de una piscifactoría seleccionada por el usuario.

pasarDias()

private static void pasarDias()
Método que permite al usuario pasar varios días en la simulación.

anadirPezAleatorio()

private static void anadirPezAleatorio()
Añade 4 peces aleatorios a una piscifactoría seleccionada por el usuario.

repartirComida()

private static void repartirComida()
Reparte la comida del almacén central equitativamente entre las piscifactorías.

todasLasPiscifactoriasEnLaMediaComidaAnimal(int mediaCantidadComidaAnimal)

private static boolean todasLasPiscifactoriasEnLaMediaComidaAnimal(int mediaCantidadComidaAnimal)
Indica si todas las piscifactorías que no están llenas están en la media en cuanto a capacidad de comida animal.

Parámetros

mediaCantidadComidaAnimal Media de la cantidad de comida animal.

devuelve True si todas las piscifactorías que no están llenas están en la media en cuanto a cantidad de comida animal.

todasLasPiscifactoriasLlenasDeComidaAnimal()

private static boolean todasLasPiscifactoriasLlenasDeComidaAnimal()
Indica si todas las piscifactorías están llenas de comida animal.

devuelve True si todas las piscifactorías están llenas de comida animal.

mediaComidaAnimal()

private static int mediaComidaAnimal()
Devuelve la media de comida animal de las piscifactorías que no estén llenas.

devuelve Media de comida animal de las piscifactorías que no estén llenas.

repartirComidaAnimal()

private static void repartirComidaAnimal()
Gestiona la lógica de distribución equitativa de comida animal del almacén central a las piscifactorías.

todasLasPiscifactoriasEnLaMediaComidaVegetal(int mediaCantidadComidaVegetal)

private static boolean todasLasPiscifatoriasEnLaMediaComidaVegetal(int mediaCantidadComidaVegetal)
Indica si todas las piscifactorías que no están llenas están en la media en cuanto a cantidad de comida vegetal.

Parámetros

mediaCantidadComidaVegetal Media de la cantidad de comida vegetal

devuelve True si todas las piscifactorías que no están llenas están en la media en cuanto a cantidad de comida vegetal.

todasLasPiscifactoriasLlenasDeComidaVegetal()

private static boolean todasLasPiscifactoriasLlenasDeComidaVegetal()
Indica si todas las piscifactorías están llenas de comida vegetal.

devuelve True si todas las piscifactorías están llenas de comida vegetal.

mediaComidaVegetal()

private static int mediaComidaVegetal()
Devuelve la media de comida vegetal de las piscifactorías que no estén llenas.

devuelve Media de comida vegetal de las piscifactorías que no estén llenas.

repartirComidaVegetal()

private static void repartirComidaVegetal()
Gestiona la lógica de distribución equitativa de la comida vegetal del almacén central a las piscifactorías.

anadirMonedasOculto()

private static void anadirMonedasOculto()
Añade 1000 monedas al sistema de monedas de la simulación.

toString()

public String toString()
Devuelve un string con información relevante de la clase.

devuelve String con información relevante de la clase.

main(String[ ] args)

public static void main(String[ ] args)
Método principal del programa que gestiona el uso del programa por parte del usuario. Este método sobrescribe al método de la superclase Object.

Parámetros

param Argumentos pasados por línea de comandos.

Interfaces

Interfaz Mar

Interfaz bandera de peces de mar.

Interfaz Río

Interfaz bandera de peces de río.