

Trabajo Práctico 1. Programación Orientada a Objetos

Puntuación

Puntaje Total: 100 puntos

Aprobación: 60 puntos

Fecha de entrega: 27/09/2022- 23:55 Hs.

Condiciones de entrega

1. El presente trabajo práctico deberá resolverse en grupo de hasta tres (3) alumnos.
2. Entrega: Se realizará por medio del Campus Virtual de la UTN, en la tarea correspondiente al TP 1. La extensión del archivo será .zip o .rar, de acuerdo al programa de compresión usado. El nombre del archivo se consigue concatenando un prefijo del número del TP con los apellidos de los integrantes por orden alfabético y separados por guiones (por ejemplo: para Pérez, Acevedo y González, el nombre será: TP1-Acevedo-Gonzalez-Perez.zip). Note que no hay espacios en blanco ni acentos en el nombre de archivo. Dentro del archivo de entrega, deben constar los siguientes:
 - Diagrama de clases conceptual, indicando la jerarquía de clases y relaciones entre las mismas. Debe incluir los cambios (si hubiera) con respecto al modelo que se propone en el enunciado. Puede utilizar PlantText, Violet, UMLet, Dia, etc. Entregar en formato pdf. Nombre: diagrama.pdf
 - Fuentes Pharo Smalltalk: Se debe crear un paquete para todas las clases del TP, llamado TP1. Realizar un #fileOut de este paquete y guardarlo en un archivo TP1.st
 - Los casos de prueba se entregarán en un archivo de texto, no deben ser capturas de pantalla. Deberán cubrir diferentes resultados que puedan obtenerse según las funcionalidades solicitadas. Se enfatiza que se adjunten casos de prueba que sean claros, válidos y suficientes para poder probar el trabajo. Como sugerencia puede ir acumulando los casos en un Playground y luego grabarlo a disco con el nombre casos-de-prueba.txt.

- Archivo de texto (integrantes.txt) con una línea para cada integrante en la cual figure el nombre del alumno/a y su dirección de email.

3. Penalización por entrega fuera de término: Si el trabajo práctico se entrega después de la fecha indicada, y hasta una semana tarde, tendrá una quita de 15 puntos. No serán recibidos trabajos luego de una semana de la fecha de entrega. Los trabajos que se deban rehacer/corregir fuera de la fecha de entrega tienen una quita de 30 puntos.

Descripción del problema:

Una empresa agropecuaria desea implementar un sistema de gestión y control de cultivos. Para ello debe tener un registro de parcelas sobre las cuales trabajará, datos de siembra, cosecha, gastos y rentabilidad.

Las parcelas podrán ser propias, alquiladas a un valor fijo o utilizadas mediante un acuerdo de entregar al propietario un porcentaje del rendimiento de la cosecha.

Cada parcela tendrá un tamaño y puede tener un propietario (para el caso de parcelas alquiladas o acuerdo de porcentajes sobre la cosecha). Las alquiladas tienen un costo de alquiler fijado al momento de alquilarse y las que corresponden a acuerdos tendrán definido el porcentaje de cosecha a abonar al propietario del campo una vez levantada la cosecha, mientras que las propias tendrán gastos impositivos los cuales serán considerados importes fijos.

Se debería poder registrar la siembra de un campo, dejando registros de fecha de siembra y variedad de cultivo sembrado.

Una vez realizada la cosecha, ésta arroja un rendimiento (toneladas por hectárea). Debería poder calcularse el rendimiento de una parcela, calculando el precio actual del cultivo por el rendimiento por hectárea por el nro de hectáreas, menos los gastos de la parcela.

Implementación:

Paso 1. Se solicita identificar las clases necesarias para la solución, sus atributos y métodos, y proponer un diagrama de clases para el sistema.

Clases mínimas requeridas:

- Empresa
- Propietario
- Parcela
- ParcelaPropia
- ParcelaAlquilada
- ParcelaAcuerdo
- Siembra

Paso 2. Se solicita implementar los siguientes métodos de instancia en la clase **Empresa**:

registrarDatosPropietario: dni **nombre:**n **apellido:**ap

Registra un nuevo propietario en la empresa. Devuelve una instancia de Propietario

incorporarParcelaPropia: *codigoParcela hectareas: cant costoImpuestos: importe*

Incorpora o agrega a la lista de Parcelas una parcela del tipo propia. Retorna una instancia de ParcelaPropia

incorporarParcelaAlquilada: *codigoParcela hectareas: cant costoAlquiler: importe: dniPropietario:dni*

Incorpora o agrega a la lista de Parcelas una parcela del tipo alquilada. Devuelve una instancia de ParcelaAlquilada. Si no existe ningún propietario con documento dni, deberá retornar nil y no realizar ninguna modificación.

incorporarParcelaAcuerdo: *codigoParcela hectareas: cant porcentajeAcordado: porcentaje dniPropietario:dni*

Incorpora o agrega a la lista de Parcelas una parcela del tipo acuerdo.

Si no existe ningún propietario con documento dni, deberá retornar nil y no realizar ninguna modificación. En caso contrario retorna una instancia de ParcelaAcuerdo.

Nota: Al momento de agregar una parcela (propia, alquilada o acuerdo) se deberá verificar que codigoParcela no esté asignado a una parcela existente. Si ya está asignado deberá retornar nil y no realizar ninguna modificación.

registrarSiembra: *codigoParcela cultivo:nombreCultivo*

Registra la siembra en la parcela indicada (chequear que la misma no esté sembrada en este momento). Devuelve una instancia de Siembra

registrarCosecha: *codigoParcela rendimientoPorHectarea:valor*

precioActualCultivo: *valor*

Deja la parcela disponible para ser vuelta a sembrar y conservará el registro histórico tanto de siembra como de cosecha para ser consultado en el futuro. Devuelve la instancia de Parcela para la cual se ha registrado una cosecha.

calcularGananciaUltimaCosecha:*codigoParcela*

Calcula y devuelve la ganancia obtenida por la empresa agropecuaria en la última cosecha en la parcela indicada. Esta se obtiene de la siguiente manera:

$$\text{ganancia} = \text{precio del cultivo} \times \text{cantidad de hectáreas de la parcela} \times \text{rendimiento por hectárea} - \text{gastos}$$

Los gastos serán el alquiler, los impuestos, el porcentaje a pagar al propietario (según corresponda).

calcularGananciaTotal:*codigoParcela*

Calcula y retorna la ganancia total histórica obtenida por la empresa agropecuaria en la parcela indicada (suma de las ganancias de todas las cosechas usando la misma fórmula del método anterior).

calcularPagoPropietarioUltCosecha:*codigoParcela*

Calcula y devuelve el importe a pagar al propietario a raíz de la última cosecha en la parcela indicada.

calcularPagoPropietarioUltCosecha:*dni*

Calcula y devuelve el importe a pagar al propietario a raíz de la última cosecha en todas las parcelas de su propiedad.

listarGanancias: *codigoParcela*

Devuelve un diccionario donde las claves corresponden a las fechas de cosecha y los valores a la ganancia total obtenida en la cosecha.

Para implementar las funcionalidades solicitadas la instancia de **Empresa** debe delegar responsabilidades en instancias de las otras clases, según corresponda. Para esto debe definir los atributos y métodos de dichas clases, implementando así el estado y comportamiento que necesite para resolver el problema planteado.

Nota: Si lo requiere puede definir otras clases y/o métodos además de los solicitados explícitamente.