

Recuperatorio
Paradigmas de Programación
[Práctico]

Nota: ()

Fecha: 31/10/15

Recuperatorio de Paradigmas de Programación

Objetivo

Evaluar al estudiante en la parte práctica de la unidad nro. 6 (Paradigma Orientado a Objetos) a partir de la resolución que guarde en el archivo más abajo especificado, correspondientes a las consignas solicitadas.

Condiciones de trabajo:

- Para resolver las consignas correspondientes al paradigma orientado a objetos, se deberá generar un archivo con el nombre Legajo_ApellidoNombre.pl para definir los hechos y reglas, conforme se solicite en las consignas que se detallan más abajo.
- **Es responsabilidad de cada alumno ir guardando periódicamente cada archivo solicitado, como así también del contenido de los mismos, teniendo la precaución de guardarlo en el disco D: para su posterior backup.**
- En caso de que máquina no funcione correctamente durante el transcurso de la evaluación, debe notificar de esta situación a cualquier docente de la mesa examinadora.
- En ningún caso debe reiniciar la máquina, ya que perderá la totalidad del examen.
- **Es responsabilidad de cada estudiante el contenido que quede guardado al momento de finalizar la evaluación, en todos sus archivos generados durante el examen.**
- El **tiempo previsto** para la realización de este examen es de 1 y 30 hs.

Recuperatorio
Paradigmas de Programación
[Práctico]

Nota: ()
Fecha: 31/10/15

Evaluación del Paradigma Orientado a Objetos

Caso de estudio:

Suponga que usted forma parte de un equipo de desarrollo que está programando en SmallTalk el viejo y conocido juego “PacMan”. Dentro de ese grupo, se divide el trabajo y a usted le toca la siguiente tarea:

- 1) Desarrollar la clase “Ficha” con los siguientes atributos:
 - a. “codigoInterno”: Es un número que identifica la ficha.
 - b. “estado”: Es un booleano que indica si la ficha ha sido comida o no por un pacman. (True = comida, False = no comida).
 - c. “coordenadaX”: Es un entero que indica la coordenada en el eje de las X donde aparece la ficha.
 - d. “coordenadaY”: Es un entero que indica la coordenada en el eje de las Y donde aparece la ficha.
- 2) Heredando de la clase “Ficha” defina las clases “FichaGrande” y “FichaPremio” con los siguientes atributos:
 - a. Para “FichaGrande” existe un atributo llamado “brillo” que es un número entero que va del uno al diez.
 - b. Para “FichaPremio” existe un atributo llamado “tipo” que puede valer: “frutilla”, “moneda” o “estrella”.
- 3) Desarrolle la clase “Juego” de manera que tenga en su interior una colección de fichas, de tal modo que algunas puedan ser tipo “Ficha”, otras tipo “FichaGrande” y otras tipo “FichaPremio”. Implemente en esta clase, los siguientes comportamientos:
 - a. “cargarFichas”: Recibe como colaborador externo una ficha y la agrega a la colección.
 - b. “cambiarEstado”: Recibe como colaborador externo un código de ficha y si la encuentra dentro de la colección, cambia su estado. Si estado era “True” pasa a “False” y si era “False” pasa a true.



Recuperatorio Paradigmas de Programación

Nota: ()

Fecha: 31/10/15

[Práctico]

- c. “contarFichasEstado”: Recibe como colaborador externo un estado (recuerde que puede ser solo True o False) y retorna la cantidad de fichas que están en ese estado dentro de la colección.
- d. “totalizarPuntos”: Retorna un número que indica el total de puntos del juego. Los puntos se calculan a partir de las fichas alojadas en la colección y mediante la siguiente lógica.
 - i. Si el estado de la ficha es False (NO esta comida) suma cero puntos.
 - ii. Si el estado de la ficha es True (SI esta comida) suma de la siguiente manera:
 - 1. Si es una ficha común, suma siempre un punto.
 - 2. Si es una ficha tipo “FichaGrande”, suma tantos puntos como brillo tenga, recuerde que brillo es un atributo de las fichas grandes.
 - 3. Si es una ficha tipo “FichaPremio”, suma, dos puntos si es “frutilla”, cuatro puntos si es “moneda” y seis puntos si es “estrella”.

Preste especial atención a estas aclaraciones porque inciden radicalmente en la calificación del examen. No aprobará el examen si no cumple con estas premisas.

- 1) No puede agregar atributos a las clases. Los atributos declarados en cada clase tienen que ser exactamente los que se piden y denominarse tal como se indica en el enunciado del problema.
- 2) Puede (y probablemente tenga que hacerlo) agregar comportamientos a las clases. Tiene total libertad para elegir su estrategia de trabajo, pero cualquier camino que tome tiene que ser correcto según las técnicas y normas de programación que propone el paradigma de objetos vistas en la materia.
- 3) El comportamiento “totalizarPuntos” debe estar implementado de manera polimórfica, o sea, cada ficha debe poder responder un mensaje que indique cuantos puntos vale. Cualquier solución que lo lleve a validar los tipos de los objetos de la colección mediante uso de condicionales o estructuras similares para hacer luego el cálculo del valor

Recuperatorio Paradigmas de Programación

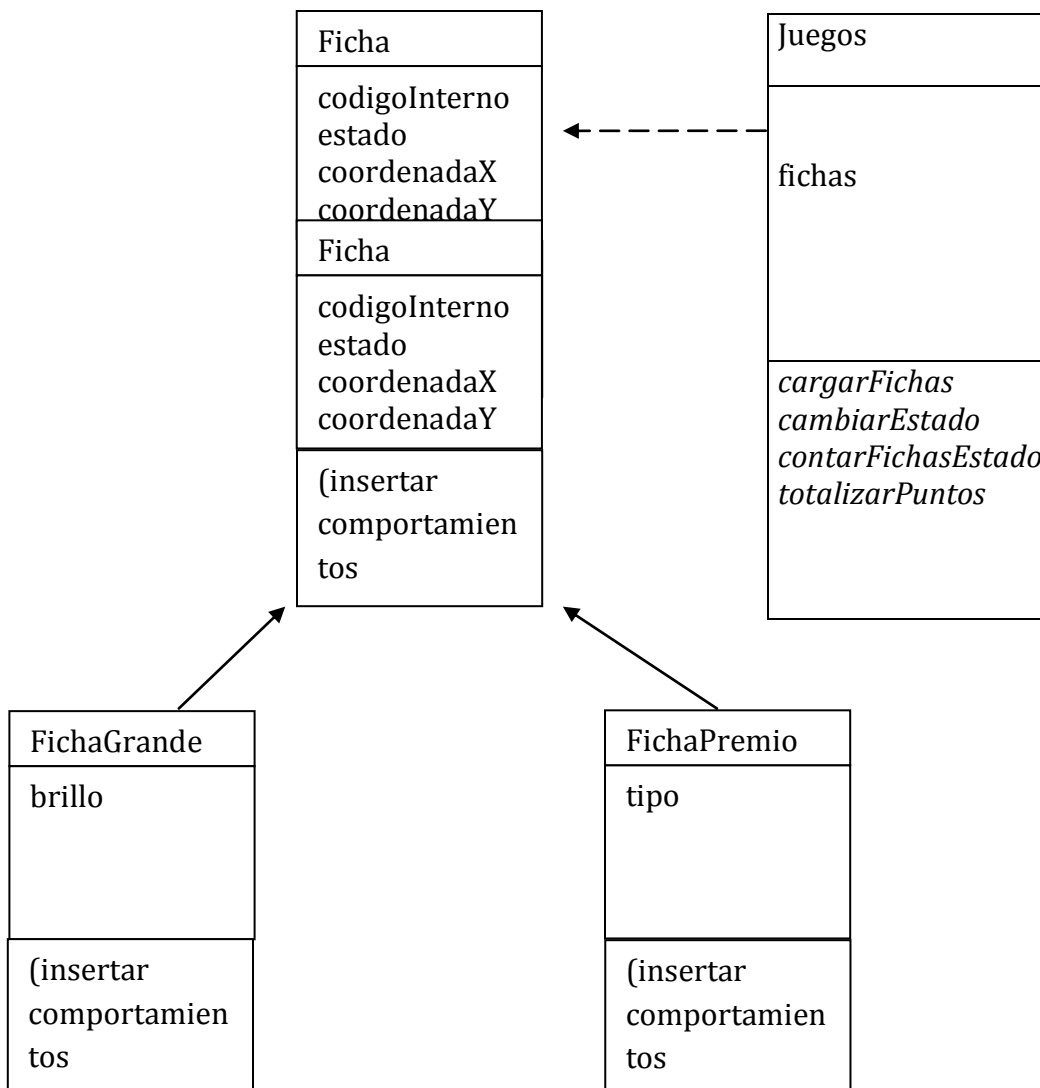
Nota: ()

Fecha: 31/10/15

[Práctico]

en puntos de cada ficha dentro del recorrido de la colección, será considerado incorrecto en un 100%

El siguiente es un diagrama de clases que representa solo los atributos y la estructura general de las clases propuestas:





Recuperatorio
Paradigmas de Programación
[Práctico]

Nota: ()

Fecha: 31/10/15

Puntaje aplicado para la corrección:

Desarrollo genera, declaración de las clases y estructura general del código: 10 puntos.

Desarrollo punto a: 15 puntos.

Desarrollo punto b: 20 puntos.

Desarrollo punto c: 25 puntos.

Desarrollo punto d: 30 puntos.

IMPORTANTE: Independientemente del puntaje obtenido, el alumno no aprobará el examen si no se desarrolla conforme a las aclaraciones realizadas en los puntos 1, 2 y 3.

← - - - - -



Universidad Tecnológica Nacional

Facultad Regional Córdoba

Ing. en Sistemas de Información

Recuperatorio

Paradigmas de Programación

Nota: ()

Fecha: 31/10/15

[Práctico]

