

Orientación Objetos 2 - Parcial 2da fecha

Patrones

Sea el desarrollo de una aplicación para registrar el desempeño académico de alumnos de la Facultad. Cada curso tiene una cantidad definida de Trabajos Prácticos. Estos se identifican por un número entero comenzando en 1. Cada curso toma un Parcial con 3 instancias (1, 2, 3). **Se toma como válida sólo la nota más alta de las tres instancias.** Tanto los parciales como los trabajos prácticos se califican con nota entera de 1 a 10. La cursada puede ser Aprobada (promedio de 6 o superior), Desaprobada (promedio menor a 6) o Ausente (no registra nota de parcial).

Para determinar la nota de la cursada se usa un algoritmo que es único por cada curso, pero que puede cambiar para diferentes cursos. Además se espera poder agregar nuevos algoritmos para computar la nota de cursada.

Actualmente existen los siguientes algoritmos para calcular la nota de cursada:

- El promedio de las notas de trabajos prácticos contribuye con un 40% a la nota final y la nota de parcial contribuye con el 60%
- Sólo se computa la nota de parcial, pero todos los TP deben tener alguna nota (aprobada o no).

Como parte de la aplicación en desarrollo, la clase Curso representa al conjunto de los inscriptos en una asignatura y ciclo lectivo y la clase Estudiante representa a un alumno y las notas en ese curso. Parte del protocolo está compuesto por los siguientes métodos

Clase Curso

variables de instancia: 'codigoMateria, cicloLectivo, estudiantes, cantidadTP'
#calcularNotas
"devuelve una colección de pares ordenados: nro legajo estudiante, nota"

Clase Estudiante

variables de instancia: 'nroLegajo, apellido, nombre, notasTP, notasParcial'
#agregarNotaTP: unaNota tpNro: numeroTP
"agrega la nota del trabajo práctico indicado"

#agregarNotaParcial: unaNota enInstancia: unaInstancia
"agrega la nota de parcial en la instancia indicada"

#notasTP
"devuelve una colección con todas las notas de los parciales"

#notaTP: numeroTP
"devuelve la nota del TP indicado"

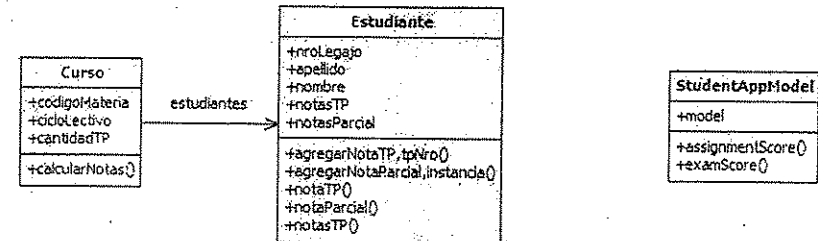
#notaParcial
"devuelve la nota del parcial"

Por otra parte, para mostrar las notas en la interfaz gráfica del sistema se dispone de la clase StudentAppModel (subclase de ApplicationModel) cuyo modelo de dominio será una instancia de Estudiante. Parte del protocolo es:

#assignmentScore: TPnumber
"retorna la nota del TP indicado por TPnumber"
self model assignmentScore: anInteger

#examScore
"retorna la nota del parcial"
self model examScore

Para realizar las tareas indicadas, tenga en cuenta no es posible modificar la implementación de la clases Estudiante y StudentAppModel, indicadas en el diagrama. Asuma que existen los accessors para todas las variables de instancia



Tareas:

1. Diseñar y modelar como diagrama de clases UML la solución para usar diferentes formas de calcular la nota de cursada. Si utiliza algún patrón, indique con estereotipos.
2. Diseñar y modelar como diagrama de clases UML la solución para mostrar las notas de un estudiante con StudentAppModel. Si utiliza algún patrón, indique con estereotipos
3. Implemente en Smalltalk
 - a. Curso#calcularNotas y todo lo necesario para resolver este método
 - b. Los dos algoritmos indicados para cálculo de nota de cursada
 - c. El mecanismo para cambiar un algoritmo de cálculo por otro.
 - d. StudentAppModel#model y todo lo necesario para resolver este método vinculando con una instancia de Estudiante de manera que los métodos #examScore y #assignmentScore devuelvan la nota del parcial y los trabajos prácticos de un estudiante