Instituto de Tecnología ORT

Carrera de Analista de Sistemas

Materia: Fundamentos de Programación

EXAMEN PARCIAL 2 - Clases y Colecciones

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta implementación de constructores.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando *setters* y *getters* sólo cuando corresponda.
- Manejo de clases, enumerados y colecciones.

Enunciado

Un grupo de amigos decide juntarse a comer un asado. Desde hace un tiempo ya definieron una mecánica que los ayuda a definir cuándo terminarán haciendo la Reunión:

- De cada Amigo se sabe su nombre (String) y su domicilio (String).
- Lo primero que hacen es anotarse a la Reunion que van a realizar todos los amigos que estén interesados en participar.
- En cualquier momento, posterior a anotarse, algún Amigo puede hacer una Propuesta de Reunión (está compuesta por un dia de la semana y un momento del dia: SABADO MEDIODIA, DOMINGO NOCHE, MARTES TARDE, SABADO NOCHE, etc.). Puede haber muchas propuestas diferentes hechas por el mismo Amigo o por diferentes amigos.
- Luego, un Amigo puede sumarse a la Propuesta de Reunión hecha por otro, “bajarse” de una Propuesta determinada o cambiar de opinión (“moverse” a otra de las Propuestas). Siempre se indica el nombre del Amigo y la descripción de la Propuesta (por ejemplo: “Juan”, SABADO, MEDIODIA). Si alguna propuesta quedara sin ningún amigo anotado la misma debe eliminarse.
- Finalmente se listan las propuestas “sobrevivientes” y quiénes están anotados para cada una de ellas.

NOTA IMPORTANTE: Se deben tener en cuenta las siguientes consideraciones.

- No pueden proponer y sumarse a las propuestas otras personas que no se hayan anotado previamente a la Reunion.
- No puede haber dos amigos que se llamen igual; si dos o más amigos tuvieran el mismo nombre, se aclara de alguna manera (por ejemplo, “Juan el Flaco”).
- No habrá propuestas repetidas (planteadas para el mismo día y momento). Si alguien propone juntarse en el mismo momento que otro ya propuso simplemente se lo agrega a la propuesta anterior.
- Si alguien se arrepiente y quiere elegir una propuesta distinta a la que ya eligió puede hacerlo, pero la nueva elección debe ser sobre una propuesta ya existente.
- Si al quitar a alguien de una propuesta de reunión ésta quedase vacía la misma debe eliminarse.
- Cada operación (el agregado de una propuesta, de un amigo, agregar o sacar a alguien de una propuesta, etc.) da por resultado uno de los siguientes valores:
  - ***OPERACION OK***,
  - ***AMIGO YA EXISTENTE***,
  - ***AMIGO NO EXISTENTE,***
  - ***PROPUESTA NO EXISTENTE***

EJEMPLOS. Estos son ejemplos que pueden darte una idea de lo requerido:

EXAMEN PARCIAL 2 - Clases y Colecciones

Acción	Datos
Crear la reunión	"Asadito"
Anotar amigo	"Amigo 1", "Domicilio 1"
Anotar amigo	"Amigo 2", "Domicilio 2"
Anotar amigo	"Amigo 3", "Domicilio 3"
Anotar amigo	"Amigo 4", "Domicilio 4"
Anotar amigo	"Amigo 5", "Domicilio 5"
Crear Propuesta	"Amigo 1", SABADO, NOCHE
Crear Propuesta	"Amigo 2", DOMINGO, MEDIODIA
Crear Propuesta	"Amigo 3", SABADO, NOCHE
Unir a Propuesta	"Amigo 3", DOMINGO, MEDIODIA
Unir a Propuesta	"Colado", SABADO, NOCHE
Bajar de Propuesta	"Amigo 3", SABADO, MEDIODIA
Bajar de Propuesta	"Amigo 3", DOMINGO, MEDIODIA
Cambiar de Propuesta	"Amigo 2", SABADO, NOCHE, DOMINGO, MEDIODIA
Listar Propuestas	Debe mostrar tanto los días que quedaron propuestos como los amigos que se sumaron a esa propuesta.

Si esto fuese un programa funcionando podría mostrar algo semejante a:

```
Agregando a Amigo 1 que vive en Domicilio 1
Agregando a Amigo 2 que vive en Domicilio 2
Agregando a Amigo 3 que vive en Domicilio 3
Agregando a Amigo 4 que vive en Domicilio 4
Agregando a Amigo 5 que vive en Domicilio 5
Agregando propuesta SABADO (NOCHE). Propuesta por Amigo 1
Agregando propuesta DOMINGO (MEDIODIA). Propuesta por Amigo 2
Agregando propuesta SABADO (NOCHE). Propuesta por Amigo 3
Amigo 3 se quiere agregar al DOMINGO (MEDIODIA)
Colado se quiere agregar al SABADO (NOCHE)
AMIGO_NO_EXISTENTE
Amigo 3 se borra del SABADO (MEDIODIA)
PROPUESTA_NO_EXISTENTE
Amigo 3 se borra del DOMINGO (MEDIODIA)
Amigo 2 se pasa del DOMINGO (MEDIODIA) al SABADO (NOCHE)



Amigos registrados para El Asadito
Amigo 1
Amigo 2
```

```
Amigo 3
Amigo 4
Amigo 5
Propuestas registradas para El Asadito
Hay 3 amigos anotados para el SABADO (NOCHE)
Amigo 1
Amigo 3
Amigo 2
```

Importante: tomar en cuenta que SOLO se debe diagramar en NS+ lo que está detallado en la sección “Se Pide” (y todos los submétodos que pudieran desprenderse de estos). Todo el resto del enunciado que describe el funcionamiento del mismo está para contextualizar el problema y puede llegar a ser pedido en otro enunciado tanto de esta materia (FP) como así también de THP.

Se pide:

- Diseñar el diagrama UML completo que resuelva el problema propuesto.
- Desarrollar en NS+ los siguientes puntos (incluyendo los métodos derivados):
    - El constructor de **Reunion**.
    - El método `crearPropuesta(...)`, que recibe el nombre de un amigo, el día y el momento del día de una Propuesta y agrega la propuesta y el amigo. En ambos casos sólo cuando es necesario.
    - El método `cambiarDePropuesta(...)`, que recibe el nombre de un Amigo, el día y el momento del día de ambas propuestas (propuesta original y propuesta actualizada) y mueve a la persona de la propuesta original a la nueva.