PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IIC2113 - Diseño Detallado de Software (2020-2)

Interrogación 2

Indicaciones

- **La prueba es individual**. Si se detecta copia será evaluado con nota mínima.
- El formato aceptado será PDF. Pueden elegir si escriben sus respuestas en un procesador de texto, un markdown, en papel y escanearlo, o lo que les acomode. Lo importante es que al final lo unan para subir un archivo PDF que sea **legible**.
- Pueden usar cualquier material (apuntes) que les acomoden para responder la prueba, siempre que si usan material externo al del curso lo citen. **Si se suben respuestas sacadas de internet sin su respectiva cita, se considerará como copia**.
- Todas las preguntas deben ser vía Issues de Github con el tag [I2 Teórico]. Solo se aceptarán dudas de enunciado. Así mismo, solo el equipo docente puede responder dudas de este tipo.
- Durante el horario de clases, desde las 14:00 hasta las 17:00, en el zoom de la clase correspondiente, se leerá el enunciado y se responderán consultas del mismo. Esta sesión será grabada y subida inmediatamente después al canal de Youtube del curso.
- Si alguien sufre un problema de fuerza mayor durante el día Viernes de la interrogación, deben escribir cuanto antes a mfsepulveda@uc.cl

Plazo de entrega: 14 de Noviembre a las 9:00 am vía buzón de tareas en Canvas.

Pregunta 1 [15 puntos totales]

Responda las siguientes preguntas justificando el por qué de su respuesta (4 puntos cada respuesta). Además explique cómo llegó a esta conclusión citando la fuente (1 punto). Considere no más de 5 líneas en cada una de sus respuestas, y una sola línea referenciando la fuente (clases, lectura, sitio, etc.-).

a) ¿Cuál es la ventaja de usar Domain Driven Design? ¿Por qué?
b) El libro Clean de Uncle Bob salió en 2008. A 12 años de su salida ¿sigue vigente? ¿Por qué?
c) ¿Por qué hay posturas que dicen que testing asegura la calidad del código? Justifique.

Pregunta 2 [15 puntos totales]

Refactoriza el siguiente código para hacerlo más mantenible y para disminuir el código repetido, mejorando también las buenas prácticas. Tu respuesta debe contener:

1. Líneas o funciones donde hiciste los cambios. Debes explicar el por qué de los cambios.
2. Debes subir tu código refactorizado funcional, en tipografía Courier new tamaño 8.

```ruby
# He knows the way
class Mando
 attr_accessor :hp
 def initialize
   @hp = 100
   @power_attack = 10
 end
 def shot(enemy)
   attack = (@power_attack * rand).to_i
   enemy.hp -= attack
   p "bang (-#{attack})"
 end
 def walk
   p 'walking'
 end
end
# A cute baby
class Baby
 attr_accessor :hp
 def initialize
   @hp = 50
   @power_attack = 100
 end
 def the_force(enemy)
   attack = @power_attack * rand
   enemy.hp -= attack
   @hp -= 2
   p "*magic sounds* (-#{attack})"
 end
 def walk
   p 'walking'
 end
end
# Someone evil
class Enemy
 attr_accessor :hp
 def initialize
   @hp = 20
   @power_attack = 1
 end
 def attack(someone)
   attack = (@power_attack * rand).to_i
   someone.hp -= attack
   p "bang (-#{attack})"
 end
```

```ruby
 def walk
   p 'walking'
 end
end

# Runs the simulation
class Main
 def self.run
   mando = Mando.new
   baby = Baby.new
   enemies = [Enemy.new, Enemy.new]

   while enemies.map(&:hp).reduce(0, :+) > 0 || (mando.hp <= 0 || baby.hp <= 0)
     mando.walk if mando.hp > 0
     baby.walk if baby.hp > 0
     enemies[0].walk if enemies[0].hp > 0
     enemies[1].walk if enemies[1].hp > 0
     mando.shot(enemies[0])
     mando.shot(enemies[1])
     if mando.hp < 10
       baby.the_force(enemies[0])
       baby.the_force(enemies[1])
     end
   end
 end
end

Main.run
```

Pregunta 3 [15 puntos totales]

Una clase `Notificator` necesita enviar mensajes de un evento (`Event`) cada una hora a distintos tipos de usuario. Esta información base es la misma, pero varía lo que se decide mostrar o lo que se calcula dependiendo del tipo de usuario. Hay 3 tipos de usuarios de momento: `manager`, `officer` y `supervisor`. Los 3 usuarios reciben el cuerpo del evento que contiene un indicador decimal que va del -10 al 10. Las diferencias en la notificación son que el supervisor recibe la lista de id's de usuarios `officer` que recibirán la notificación y el `manager` recibe el valor máximo y mínimo de los indicadores de eventos enviados de los últimos 100 eventos.

Utilizando algún patrón de diseño, implemente **solamente** la lógica de la notificación a los usuarios en base al patrón escogido. Puede usar un diagrama de clases o escribir el código. Si decide usar un diagrama, cualquier información faltante le restará puntaje. Si escribe código, puede abstraerse de algunos cálculos a través de funciones no implementadas. Finalmente, justifique el por qué del patrón utilizado.

Pregunta 4 [15 puntos totales]

Identifique el o los code smells presentes en el siguiente código. **Con un solo code smell basta**. Debe señalar, **explicar el por qué** de este code smell **y las implicancias** que podrían significar.

Contexto: Google es un buscador web que además ofrece servicios de correo como Gmail. Facebook es una red social. La aplicación actualmente permite login con Facebook y Google, además de compartir en redes sociales algunos avances de la aplicación. También realiza búsquedas a través de google con el buscador público de la aplicación.

```
public class Client
{
    protected void Login() { ... }
    protected void NewPost() { ... }
    protected void Search() { ... }
    protected void AddFriend() { ... }
}

public class Google : Client
{
    ...
}

public class Facebook : Client
{
    ...
}
```

[Éxito y buena suerte]