

Primera Práctica

Lenguajes de Programación Orientada a Objetos:

Java y Ruby

Tercera sesión (Ruby)

A) Asegúrate que has incluido en la cabecera del método *initialize* de *BadConsequence* todas las variables necesarias:

1. La cabecera de tu método *initialize* debería ser:

```
initialize(aText, someLevels, someVisibleTreasures,
someHiddenTreasures, someSpecificVisibleTreasures,
someSpecificHiddenTreasures, death)
```

2. Define los siguientes métodos de instancia de clase (en cada uno de ellos se usará el *new* de *BadConsequence* correspondiente):

```
BadConsequence.newLevelNumberOfTreasures (aText, someLevels,
someVisibleTreasures, someHiddenTreasures)
```

```
BadConsequence.newLevelSpecificTreasures (aText, someLevels,
someSpecificVisibleTreasures, someSpecificHiddenTreasures)
```

```
BadConsequence.newDeath (aText)
```

3. Indica que el método de instancia de clase de *new* tiene visibilidad privado con:
`private_class_method :new`

B) Asegúrate de que has incluido en la cabecera del método *initialize* de la clase *Monster* todas las variables necesarias (incluidas *Prize* y *BadConsequence*).

C) Define el método *to_s* (similar a *toString* de Java) en las tres clases definidas (*Monster*, *BadConsequence* y *Prize*), este método devuelve un *String* con el estado del objeto correspondiente, por ejemplo en la clase *Prize*, el método *to_s* tendrá la siguiente implementación:

```
def to_s
  "Tesoros ganados: #{@treasure} \n Niveles ganados: #{@levels}"
end
```

D) En *PruebaNapakalaki* prueba las modificaciones realizadas creando y consultado objetos de las clases modificadas.

E) Construye y rellena el mazo de monstruos, para ello:

1. En *PruebaNapakalaki* define un array en el que se van a incluir todos los monstruos, al igual que se hizo en la sesión de Java.

```
monsters = Array.new
```

2. Construye todos los monstruos que están en el pdf e inclúyelos en el array definido. Por ejemplo, para el primer monstruo de ejemplo de la sesión anterior sería:

```
price = Prize.new(4,2)
```

```
badConsequence = BadConsequence.newLevelNumberOfTreasures('Pierdes
5 niveles y 3 tesoros visibles',5 , 3, 0)
```

```
monsters << Monster.new('El rey de rosa',13,price,badConsequence)
```

Nota: Modifica esta carta en la versión de Java, ya que no habíamos tenido en cuenta la pérdida de 3 tesoros visibles.

Para el segundo ejemplo de la sesión anterior sería:

```
prize = Prize.new(4,1)
badConsequence = BadConsequence.newLevelSpecificTreasures ('Te
    atrapan para llevarte de fiesta y te dejan caer en mitad del
    vuelo. Descarta 1 mano visible y 1 mano oculta', 0,
    [TrearureKind::ONEHAND],[TrearureKind::ONEHAND])
monsters<< Monster.new('Ángeles de la noche ibicenca', 14, prize,
    badConsequence)
```

NOTA: para no tener problemas con los acentos debes incluir en la primera línea de código de todos los archivo .rb la siguiente línea: `#encoding: utf-8`

F) Desarrolla el código necesario para llevar a cabo las siguientes funcionalidades:

- Mostrar todos los monstruos que tengan un nivel de combate superior a 10.
- Mostrar todos los monstruos que tengan un mal rollo que implique sólo pérdida de niveles.
- Mostrar todos los monstruos que tengan un buen rollo que indique una ganancia de niveles superior a 1.
- Mostrar todos los monstruos que tengan un mal rollo que suponga la pérdida de un determinado tipo de tesoros ya sea visibles y/o ocultos.

Debe mostrarse el nombre, nivel de combate, buen rollo y mal rollo de cada monstruo.

NOTA: Antes de terminar la práctica comprueba que no te falte ningún monstruo, las cartas de monstruos son los que están en `cartasMonstruos.pdf` de swad.

Entrega y examen de la primera práctica en la semana del 27 al 31 de octubre.

- **Lugar:** Aula en la que se desarrolla la correspondiente sesión de prácticas.
- **Día:** El correspondiente a la sesión de cada uno de la semana indicada.
- **Duración:** 20 minutos al comienzo de la sesión.
- **Tipo examen:** Realizar pequeñas modificaciones sobre el propio código.
- Se pedirá entregar el código de la primera práctica con las modificaciones realizadas.