

Examen Final 18/6/2025

A un joc tenim una classe character on un personatge pot ser del tipus Knight, Wizard o Archer, podràs veure el seu codi a la següent imatge. Segons el tipus que sigui farà una defensa o un atac en concret:

- **Knight:** Attacks with a sword, and can defense with a shield (default defense), dodge, or magic barrier.
- **Wizard:** Casts spells to attack, and uses magic barrier to defend.
- **Archer:** Shoots arrows to attack, and uses dodge to to defend

```

1 public class Character {
2     private final String name;
3     private final int type; //0= Archer, 1=Wizard, 2=Knight
4     private int typeDefense; //0=shield, 1=dodge, 2=magic barrier
5
6     public Character(String name, int type){
7         this.name = name;
8         this.type = type;
9     }
10
11     public void setDefense( int typeDefense ){ this.typeDefense = typeDefense;}
12
13     public String getName() { return this.name; }
14
15     public int getType() { return this.type; }
16
17     public String attacking(){
18         String aux = null;
19         if( this.type == 0) aux = "Archer shoots an arrow";
20         else if( this.type == 1) aux = "Wizard casts a spell!";
21         else if( this.type == 2) aux = "Knight attacks with a sword!";
22         return aux;
23     }
24
25     public String defending(){
26         String aux = null;
27         switch (this.type){
28             case 0:
29                 aux = "Archer is using a shield to defend!"; break;
30             case 1:
31                 aux = "Wizard is creating a magic barrier for defense!"; break;
32             case 2:
33                 if(typeDefense == 0)
34                     aux = "Knight is using a shield to defend!";
35                 else if (typeDefense == 1)
36                     aux = "Knight is dodging to avoid attack!";
37                 else if (typeDefense == 2)
38                     aux = "Knight is creating a magic barrier for defense!";
39                 break;
40             default: return null;
41         }
42         return aux;
43     }
44 }

```

Exercici 1 [30 punts]:

A partir del codi del mètode `defending` de la classe `Character`:

- a) Dibuixa el graf de flux de control [10 punts].
- b) Quina és la complexitat ciclomàtica, i per què la volem calcular? [5 punts]
- c) Selecciona un conjunt bàsic [15 punts].

Exercici 2 [50 punts]:

Com has pogut observar al exercici 1, la complexitat ciclomàtica és elevada, a més ens agradaria protegir-nos de possibles variacions (crear nous personatges, o crear nous atacs o defenses pels personatges actuals). Com pots veure al codi, el personatge `Knight` és l'únic que pot canviar la seva defensa, i cada personatge pot fer un tipus d'atac o defensa. Explica quins patrons cal aplicar i per què, representa un diagrama de classe UML on detallis tots els membres necessaris per aplicar el patró.

Exercici 3 [20 punts]:

A partir de les classes indicades al exercici 2, utilitza un patró creador per instanciar personatges del joc.