

Metodologías de Programación I

Héroes de Ciudad

Patrón Decorator

Propósito

Implementar la particularidad de agregar “dificultades” al apagado de incendios de plazas por parte de los bomberos.

Plazas

Un bombero que asista a sofocar un incendio en una plaza se puede encontrar con diferentes dificultades para apagar el fuego:

- La plaza puede tener pasto seco que hace que se necesite el doble de caudal de agua para apagar el incendio.
- La plaza puede tener árboles grandes que “frenen” el chorro de agua, causando una pérdida del 25% del caudal de agua.
- Si en la plaza había mucha gente, ésta al salir corriendo pisa la manguera al bombero causando que por cada persona la manguera pierda $\frac{3}{4}$ de su caudal.
- El incendio puede ocurrir en un día de mucho calor, avivando las llamas y haciendo que se necesite más caudal de agua (depende de la temperatura del día).
- El incendio puede ocurrir en un día con mucho viento haciendo que sea más difícil de combatir las llamas, para lo cual se necesitará más de agua (dependerá de la velocidad del viento).

El bombero solo tiene suerte si:

- El incendio ocurre en un día lluvioso lo que causa que el agua de lluvia le ayude a apagar el incendio (dependerá de cuánto llueva).

Implementación

- 1) Implemente una clase *Sector* que represente un sector de la plaza. Ésta clase deberá ser capaz de almacenar un porcentaje de afectación del fuego (un variable de tipo *double*). Además deberá implementar dos métodos:
 - a. *void mojar(double agua)*
resta al porcentaje de afectación almacenado el valor recibido por parámetro.
 - b. *bool estaApagado()*
devuelve *true* si el porcentaje de afectación llegó a cero o *false* en caso contrario.
- 2) Modifique la interface *ILugar* para que el método *getSectores* devuelva una matriz de *NxN* pero de *Sectores*. Haga los cambios necesarios en todas las clases afectadas.
 - a. *Sector[][] getSectores()*

- 3) Modifique el método *apagarIncendio* de *Bombero* invocando a los métodos *mojar* y *estaApagado* de *Sector*.
- 4) Implemente con el patrón Decorator los cinco decorados descritos para un *Sector*, donde cada decorado modifica el valor del agua recibida en el método *mojar*:
 - a. El decorado “pasto seco” divide por dos el valor del caudal de agua.
 - b. El decorado “árboles grandes” resta un 25% al valor del caudal de agua.
 - c. El decorado “día de mucho calor” tendrá un parámetro *temperatura* (entre 30 y 45) y le resta al valor de agua recibido el resultado de $(0.1 * temperatura)$.
 - d. El decorado “día ventoso” tendrá un parámetro *velocidad del viento* (entre 80 y 250) y le resta al valor de agua recibido el resultado de $(e^{(velocidad / 100)})$. Use `Math.Exp(valor)` para el cálculo del exponencial.
 - e. El decorado “gente asustada” tendrá un parámetro *cantidad de personas* (entre 0 y 5) que restará el 75% al valor de agua recibido. Esta resta solo lo hará como máximo *cantidad de personas* veces.
 - f. El decorado “día lluvioso” tendrá un parámetro intensidad de lluvia (entre 1 y 500) que se sumará al caudal recibido.

Testing

Utilice y adapte el siguiente código para crear sectores decorados al azar.

```
Sector decorarSector(Sector sector, int caudalLluvia, int temperatura, int velocidadViento){
    double probabilidad_de_decorar = 0.2;
    if(random.NextDouble() < probabilidad_de_decorar)
        sector = CREAR DECORADO “PASTO SECO”;
    if(random.NextDouble() < probabilidad_de_decorar)
        sector = CREAR DECORADO “ARBOLES GRANDES”;
    if(random.NextDouble() < probabilidad_de_decorar)
        sector = CREAR DECORADO “GENTE ASUSTADA”;
    if(temperatura > 30)
        sector = CREAR DECORADO “MUCHO CALOR”;
    if(velocidadViento > 80)
        sector = CREAR DECORADO “MUCHO VIENTO”;
    if(caudalLluvia > 0)
        sector = CREAR DECORADO “DIA LLUVIOSO”;
    return sector;
}

Sector crearSector(int caudalLluvia, int temperatura, int velocidadViento){
    int caudal = random.Next(100);
    Sector sector = new Sector(caudal);
    return decorarSector(sector, caudalLluvia, temperatura, velocidadViento);
}
```

Use la función *main* de la clase *HeroesDeCiudad* implementada en la actividad 2 para comprobar el correcto funcionamiento del sistema.