

Metodologías de Programación I

Héroes de Ciudad

Patrón Abstract factory

Propósito

Implementar la correcta creación de héroes, sus vehículos, sus herramientas y sus cuarteles de operación.

Héroes y sus vehículos, herramientas y cuarteles

Cada uno de nuestros cuatro tipos de héroes necesitan vehículos especiales, herramientas y un cuartel adecuado para trabajar.

- Bombero
 - Vehículo: Autobomba
 - Herramienta: Manguera
 - Cuartel: Cuartel de bomberos
- Médico
 - Vehículo: Ambulancia
 - Herramienta: Desfibrilador
 - Cuartel: Hospital
- Policía
 - Vehículo: Patrullero
 - Herramienta: Pistola
 - Cuartel: Comisaría
- Electricista
 - Vehículo: Camioneta con plataforma elevadora
 - Herramienta: Buscapolo
 - Cuartel: Central eléctrica

Implementación

- 1) Implemente la interface *IVehiculo* con los siguientes métodos
 - a. `void encenderSirena();`
 - b. `void conducir();`
- 2) Implemente las clases *Autobomba*, *Ambulancia*, *Patrullero* y *Camioneta* haciendo que todas implementen la interface *IVehiculo*. Los métodos *encenderSirena* y *conducir* deberían imprimir textos distintos por consola.
- 3) Implemente la interface *IHerramienta* con los siguientes métodos
 - a. `void usar();`
 - b. `void guardar();`

- 4) Implemente las clases *Manguera*, *Desfibrilador*, *Pistola* y *Buscapolo* haciendo que todas implementen la interface *IHerramienta*. Los métodos *usar* y *guardar* deberían imprimir textos distintos por consola.
- 5) Modifique las clases *Bombero*, *Medico*, *Policia* y *Electricista* para que tengan una variable de instancia *IHerramienta* y otra *IVehiculo*, junto con sus getters y setters correspondientes.
- 6) Implemente la interface *ICuartel* con los siguientes métodos
 - a. void agregarVehiculo(IVehiculo);
 - b. void agregarPersonal(IResponsable);
 - c. void agregarHerramienta(IHerramienta);
 - d. IResponsable getPersonal();
- 7) Implemente las clases *CuartelDeBomberos*, *Hospital*, *Comisaria* y *CentralElectrica* haciendo que todas implementen la interface *ICuartel*. Los métodos *agregarVehiculo*, *agregarHerramienta* y *agregarPersonal* almacenan el objeto recibido como parámetro en distintas listas internas (*vehículos*, *responsables* y *herramientas*).
El método *getPersonal*, saca un personal de la lista correspondiente, saca un vehículo y una herramienta de sus listas correspondientes y se los *setea* al *IResponsable*, luego lo devuelve listo para la acción.
- 8) Implemente usando Abstract Factory cuatro fábricas para crear a nuestros héroes con sus respectivos cuarteles, herramientas y vehículos. Use para esto la interface *IFabricaDeHeroes*:
 - a. IResponsable crearHeroe();
 - b. IVehiculo crearVehiculo();
 - c. IHerramienta crearHerramienta();
 - d. ICuartel crearCuartel();
- 9) Modifique los métodos de “acción” de cada héroe (*apagarIncendio*, *atenderInfarto*, *patrullarCalles* y *revisar*) para que se invoque a los métodos *sirena* y *conducir* del *IVehiculo* asociado y al método *usar* de la *IHerramienta* correspondiente, antes de realizar la acción. Al finalizar la acción invoque al método *guardar* de la *IHerramienta* utilizada.

Testing

Implemente en la clase *HeroesDeCiudad* una función estática *crearHeroe* que reciba una *IFabricaDeHeroes* por parámetro y que devuelva un *ICuartel*. Usando la fábrica recibida esta función debe:

1. Pedir la creación de un cuartel
2. Pedir la creación de un héroe y guardarlo en el cuartel creado en 1.
3. Pedir la creación de un vehículo y guardarlo en el cuartel creado en 1.
4. Pedir la creación de una herramienta y guardarla en el cuartel creado en 1.
5. Devolver el cuartel creado en 1.

Implemente en un método *main* de la clase *HeroesDeCiudad* la instanciación de cada una de las cuatro fábricas de héroes.

Por cada fábrica invoque a la función *crearHroe* y al cuartel recibido envíele el mensaje *getPersonal* y al personal recibido envíelo a la acción (*apagarIncendio*, *atenderInfarto*, *patrullarCalles* o *revisar* según sea el caso). Cuando el héroe finalice su tarea devuélvalo al cuartel correspondiente.

Compruebe el correcto funcionamiento del sistema.