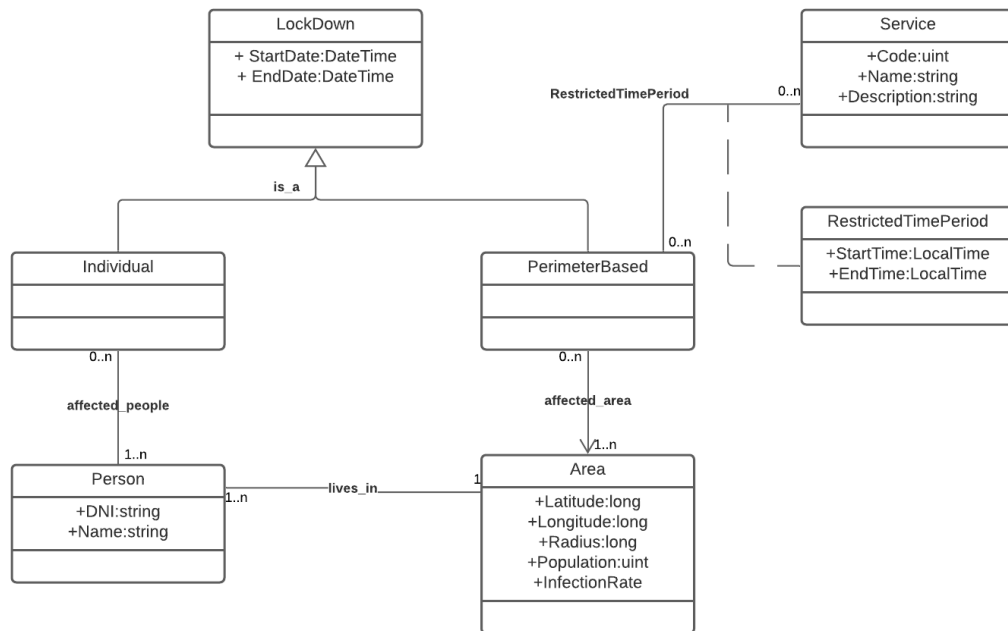


Ingeniería del Software (ISW)



Nota 1. Existe una restricción de navegabilidad entre **PerimeterBased** y **Area**

Nota 2. Solo se piden los atributos, no los métodos.

Nota 3. **LockDown** es una clase abstracta.

- (1.5 puntos) Realiza el diseño en C# atendiendo a los patrones vistos en clase.
- (1 punto) Proporciona la cabecera de los métodos constructores necesarios (sin la implementación del cuerpo del método)
- (1 punto) Implementa el código necesario para crear un *LockDown* de tipo *PerimeterBased* con un servicio restringido y que afecta a un área. El sistema ha de tener también otro *LockDown* de tipo individual que afecta solo a una persona. Inventa el valor de todos los parámetros necesarios (**solo los imprescindibles**) para dejar el sistema en un estado consistente tras dicha creación.

Solución:

Area

```
public class Area
{
    //Propiedades directas
    public long Latitude { get; set; }
    public long Longitude { get; set; }
    public long Radius { get; set; }
    public uint Population { get; set; }
    public long InfectionRate { get; set; }

    //Relationships
```

Ingeniería del Software (ISW)

```
public virtual ICollection<Person> LivesInPeople { get; set; }
```

```
//Constructor header
```

```
public Area(long Latitude, long Longitude, long Radius, uint Population,  
            long InfectionRate, Person LivesInPerson) { }
```

```
}
```

LockDown

```
public abstract class LockDown
```

```
{
```

```
//Propiedades directas
```

```
public DateTime StartDate { get; set; }
```

```
public DateTime EndDate { get; set; }
```

```
//Constructor header
```

```
public LockDown(DateTime StartDate, DateTime EndDate) { }
```

```
}
```

Individual

```
public class Individual:LockDown
```

```
{
```

```
//Relationships
```

```
public virtual ICollection<Person> AffectedPeople { get; set; }
```

```
//Constructor header
```

```
public Individual(DateTime StartDate, DateTime EndDate, Person  
                  AffectedPerson):base(StartDate, EndDate) { }
```

```
}
```

PerimeterBased

```
public class PerimeterBased:LockDown
```

```
{
```

```
//Relationships
```

```
public virtual ICollection<Area> AffectedAreas { get; set; }
```

```
public virtual ICollection<RestrictedTimePeriod> RestrictedTimePeriods { get; set; }
```

```
//Constructor header
```

```
public PerimeterBased(DateTime StartDate, DateTime EndDate, Area AffectedArea) :  
                      base(StartDate, EndDate) { }
```

```
}
```

Person

```
public class Person
```

```
{
```

```
//Propiedades directas
```

```
public string DNI { get; set; }
```

```
public string Name { get; set; }
```

```
// Relationships
```

```
public virtual Area LivesInArea { get; set; }
```

Ingeniería del Software (ISW)

```
public virtual ICollection<Individual> AffectingIndividualLockDowns { get; set; }
```

```
//relaxed constructor 1-1..n association  
//Constructor header  
public Person(string DNI, string Name) { }  
}
```

Service

```
public class Service  
{  
    //Propiedades directas  
    public uint Code { get; set; }  
    public string Name { get; set; }  
    public string Description { get; set; }  
  
    //Relationships  
    public virtual ICollection<RestrictedTimePeriod> RestrictedTimePeriods { get; set; }  
  
    //Constructor header  
    public Service(uint Code, string Name, string Description) { }  
}
```

RestrictedTimePeriod

```
public class RestrictedTimePeriod  
{  
    //Propiedades directas  
    public LocalTime StartTime { get; set; }  
    public LocalTime EndTime { get; set; }  
  
    //Relationships  
    public virtual Service RestrictedService { get; set; }  
    public virtual PerimeterBased RestrictedPerimeterBasedLockDown { get; set; }  
  
    //Constructor header  
    public RestrictedTimePeriod(LocalTime StartTime, LocalTime EndTime, Service  
        RestrictedService, PerimeterBased RestrictedPerimeterBasedLockDown) { }  
}
```

C)

Main

```
static void Main(string[] args)  
{  
    Person affectedPerson = new Person("11111111A", "Javier Jaen");  
    Individual individualLockDown = new Individual(new DateTime(2020, 12, 12), new  
        DateTime(2020, 12, 24), affectedPerson);  
    Area area = new Area(2021, 1221, 23, 532123, 600, affectedPerson);  
    affectedPerson.LivesInArea = area;  
    affectedPerson.AffectingIndividualLockDowns.Add(individualLockDown);  
}
```

Ingeniería del Software (ISW)

```
PerimeterBased perimeterBasedLockDown = new PerimeterBased(new
    DateTime(2020, 11, 12), new DateTime(2020, 11, 21), area);
Service affectedService = new Service(23, "Hotel Cristina", "Hotel Cerrado por foco
    Covid");
RestrictedTimePeriod period = new RestrictedTimePeriod(new LocalTime(22, 00),
    new LocalTime(23, 30), affectedService, perimeterBasedLockDown);
perimeterBasedLockDown.RestrictedTimePeriods.Add(period);
affectedService.RestrictedTimePeriods.Add(period);

}
```