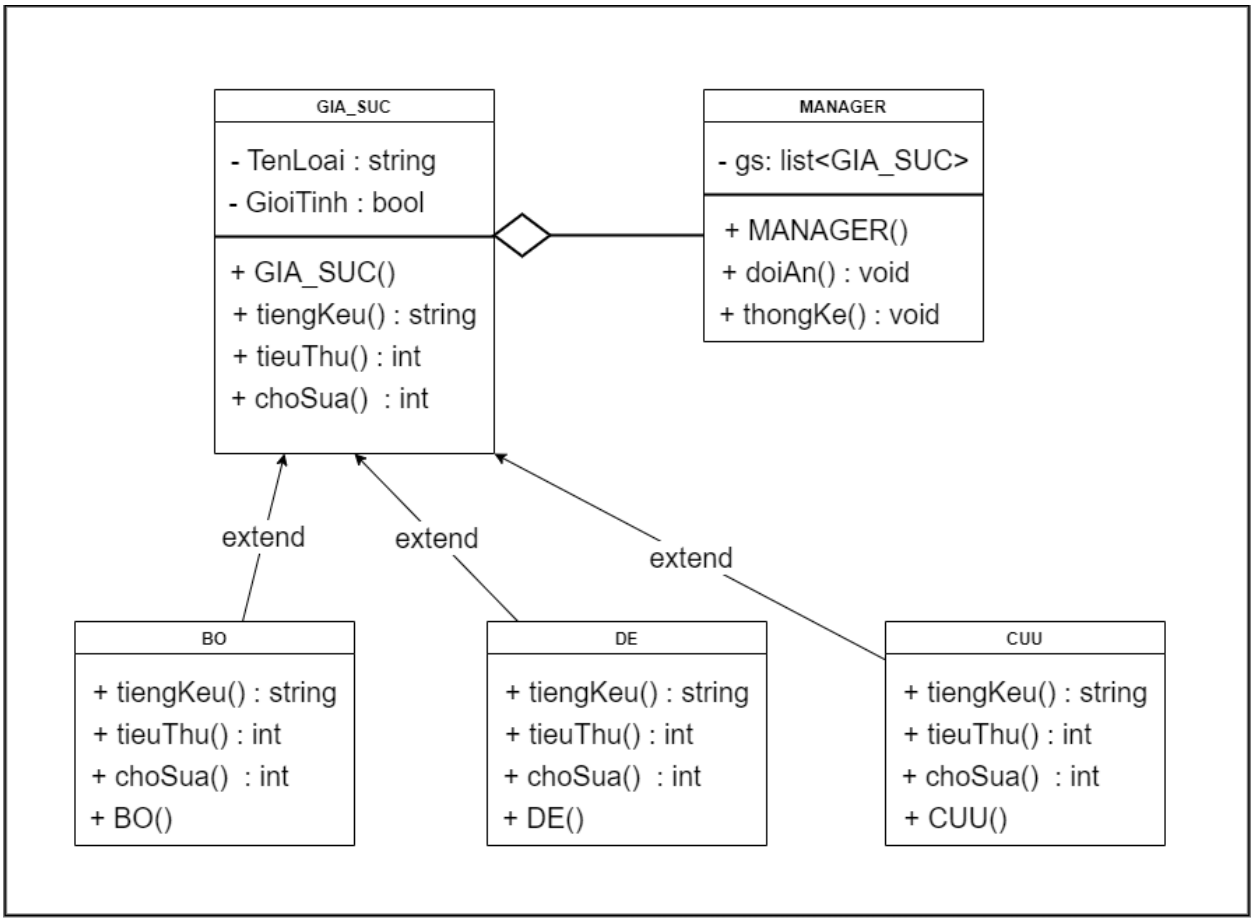
UML :


GIA_SUC
- TenLoai : string
- GioiTinh : bool
+ GIA_SUC()
+ tiengKeu() : string
+ tieuThu() : int
+ choSua() : int
MANAGER
- gs: list<GIA_SUC>
+ MANAGER()
+ doiAn() : void
+ thongKe() : void
extend
extend
extend
BO
+ tiengKeu() : string
+ tieuThu() : int
+ choSua() : int
+ BO()
DE
+ tiengKeu() : string
+ tieuThu() : int
+ choSua() : int
+ DE()
CUU
+ tiengKeu() : string
+ tieuThu() : int
+ choSua() : int
+ CUU()

SRC:

```
using System;
using System.Collections.Generic;

namespace BTVN_5
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Console.OutputEncoding = System.Text.Encoding.UTF8;
            Manager manager = new Manager();

            manager.themGiaSuc(new BO(true));  // đực
            manager.themGiaSuc(new BO(false)); // cái
            manager.themGiaSuc(new DE(true));
            manager.themGiaSuc(new DE(false));
            manager.themGiaSuc(new CUU(true));
            manager.themGiaSuc(new CUU(false));

            manager.doiAn();
            manager.thongKe();

            Console.ReadLine();
        }
```

```
        7 references
        public abstract class GIA_SUC
        {
            4 references
            public bool GioiTinh { get; protected set; }
            5 references
            public string tenLoai { get; protected set; }

            0 references
            public GIA_SUC() { }

            5 references
            public abstract string tiengKeu();
            4 references
            public abstract int tieuThu();
            4 references
            public abstract int choSua();
            1 reference
            public int deCon()
            {
                Random random = new Random();
                return random.Next(1, 4);
            }
        }
```

```
3 references
public class DE : GIA_SUC
{
    2 references
    public DE(bool GioiTinh)
    {
        this.tenLoai = "Dê";
        this.GioiTinh = GioiTinh;
    }

    3 references
    public override string tiengKeu() => "Be Be";

    2 references
    public override int tieuThu() => new Random().Next(1, 3);

    2 references
    public override int choSua() => new Random().Next(2, 5);
}

3 references
public class BO : GIA_SUC
{
    2 references
    public BO(bool GioiTinh)
    {
        this.tenLoai = "Bò";
        this.GioiTinh = GioiTinh;
    }

    3 references
    public override string tiengKeu() => "Uhm bò";

    2 references
    public override int tieuThu() => new Random().Next(1, 6);

    2 references
    public override int choSua() => new Random().Next(5, 11);
}
```

```
3 references
public class CUU : GIA_SUC
{
    2 references
    public CUU(bool GioiTinh)
    {
        this.tenLoai = "Cừu";
        this.GioiTinh = GioiTinh;
    }

    3 references
    public override string tiengKeu() => "Cừu kêu";

    2 references
    public override int tieuThu() => new Random().Next(1, 4);

    2 references
    public override int choSua() => new Random().Next(3, 7);
}
```

```
        3 references
        public class Manager
        {
            List<GIA_SUC> giaSuc = new List<GIA_SUC>();

            1 reference
            public Manager() { }

            6 references
            public void themGiaSuc(GIA_SUC giaSuc)
            {
                this.giaSuc.Add(giaSuc);
            }

            1 reference
            public void doiAn()
            {
                for (int i = 0; i < this.giaSuc.Count; i++)
                {
                    Console.WriteLine($"{this.giaSuc[i].tenLoai}: {this.giaSuc[i].tiengKeu()}");
                }
            }
            1 reference
            public void thongKe()
            {
                foreach (var animal in giaSuc)
                {
                    Console.WriteLine($"{animal.tenLoai}:");
                    Console.WriteLine($"- Tiếng kêu: {animal.tiengKeu()}");
                    Console.WriteLine($"- Số lượng thức ăn tiêu thụ: {animal.tieuThu()} kg");

                    if (!animal.GioiTinh) // Chỉ gia súc cái mới đẻ con và cho sữa
                    {
                        Console.WriteLine($"- Số con đẻ ra: {animal.deCon()}");
                        Console.WriteLine($"- Số lượng sữa cho: {animal.choSua()} lít");
                    }

                    Console.WriteLine(); // Dòng trống ngăn cách giữa các con vật
                }
            }
        }
    }
}
```