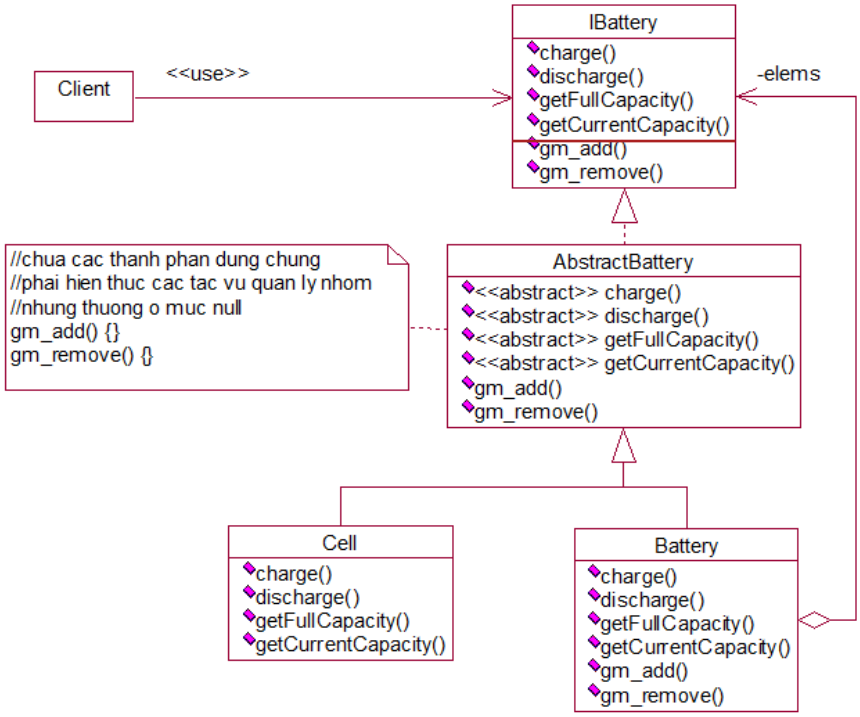
Ta sẽ dùng mẫu thiết kế Composite với lược đồ class như sau để tổ chức các class liên quan đến hệ thống cung cấp dịch vụ dùng pin sạc :



Vai trò, vị trí của từng phần tử trong lược đồ trên như sau :

- IBattery : cung cấp interface hợp nhất cho mọi đối tượng pin sạc, interface hợp nhất này sẽ cung cấp 2 nhóm tác vụ : nhóm các tác vụ chức năng của mọi pin sạc, nhóm tác vụ quản lý các thành phần trong đối tượng pin tích hợp.
- AbstractBattery : class đặc tả những thông tin được sử dụng chung cho mọi loại pin. Nó cần hiện thực các tác vụ quản lý thành phần trong đối tượng phức hợp (nhưng thường chỉ ở mức tối thiểu) để các đối tượng pin cơ bản không phải hiện thực lại chúng (vì không liên quan đến pin cơ bản).
- Cell : class đặc tả pin cơ bản.
- Battery : class đặc tả pin tích hợp.

Sau đây là mã nguồn VC# để đặc tả các thành phần trên :

```
//đặc tả interface của mọi đối tượng pin
interface IBattery  {
    //1. Nhóm các tác vụ chức năng của đối tượng (tích hợp hay thành phần)
    //tác vụ tham khảo công suất max của battery
    int getFullCapacity();
    //tác vụ tham khảo công suất hiện hành của battery
    int getCurrentCapacity();
    //tác vụ thải 1u (đơn vị điện năng) khỏi battery
    bool discharge();
    //tác vụ nạp 1u vào battery
```

```
        bool charge();
        //2. Nhóm các tác vụ quản lý các thành phần con của đối tượng tích hợp
        //tác vụ thêm thành phần b vào battery hiện hành
        void gm_add(IBattery b);
        //tác vụ bớt thành phần b ra khỏi battery hiện hành
        void gm_remove(IBattery b);
}
//đặc tả class chứa các thông tin dùng chung của mọi đối tượng pin
abstract class AbstractBattery : IBattery  {
        //tác vụ tham khảo công suất max của battery
        public abstract int getFullCapacity();
        //tác vụ tham khảo công suất hiện hành của battery
        public abstract int getCurrentCapacity();
        //tác vụ thải 1u khỏi battery
        public abstract bool discharge();
        //tác vụ nạp 1u vào battery
        public abstract bool charge();
        //tác vụ thêm battery b vào battery hiện hành
        public virtual void gm_add(IBattery b) { }
        //tác vụ bớt battery b ra khỏi battery hiện hành
        public virtual void gm_remove(IBattery b) { }
}
//đặc tả pin cơ bản
class Cell : AbstractBattery {
        private int capacity;
        private int fullCapacity;
        //tác vụ khởi tạo battery
        public Cell(int capamax) {
            fullCapacity = capamax;
            capacity = capamax;
        }
        //tác vụ nạp 1u vào battery
        public override bool charge() {
            //version chưa xử lý nạp quá công suất
            capacity++;
            return true;
        }
        //tác vụ thải 1u khỏi battery
        public override bool discharge() {
            //version chưa xử lý cạn công suất
            capacity--;
            return true;
        }
        //tác vụ tham khảo công suất hiện hành của battery
        public override int getCurrentCapacity()  {
            return capacity;
```

```
        }
        //tác vụ tham khảo công suất max của battery
        public override int getFullCapacity() {
            return fullCapacity;
        }
        //tác vụ giải mã trạng thái battery thành chuỗi văn bản
        public override String ToString() { return capacity.ToString(); }
}
//đặc tả pin tích hợp
class Battery : AbstractBattery  {
        //danh sách các battery thành phần
        private List<IBattery> elems = new List<IBattery>();
        int aCharger = -1;  //thiết lập chỉ số battery đã nạp lần cuối
        int aDecharger = -1;  //thiết lập chỉ số battery đã thải lần cuối

        //tác vụ khởi tạo battery tích hợp có nCells, mỗi cell có công suất capCells
        //dùng chiến lược nạp/thải được qui định bởi bycycles
        public Battery(int nCells, int capCells, bool bycycles) {
            for (int i = 0; i < nCells; i++) {
                elems.Add(new Cell(capCells));
            }
            //và các việc khởi động khác nếu cần
        }
        //tác vụ thêm battery b vào battery hiện hành
        public override void gm_add(IBattery b)  {
            elems.Add(b);
        }
        //tác vụ bớt battery b ra khỏi battery hiện hành
        public override void gm_remove(IBattery b)  {
            elems.Remove(b);
        }
        //tác vụ nạp 1u vào battery dùng chiến lược Round-Robin
        public override bool charge() {
            //hiệu chỉnh chỉ số battery cần nạp
            aCharger = (aCharger + 1) % elems.Count;
            return elems[aCharger].charge();
        }
        //tác vụ thải 1u khỏi battery dùng chiến lược Round-Robin
        public override bool discharge() {
            //hiệu chỉnh chỉ số battery cần thải
            aDecharger = (aDecharger + 1) % elems.Count;
            return elems[aDecharger].charge();
        }
        //tác vụ tham khảo công suất hiện hành của battery
        public override int getCurrentCapacity()  {
            int sum = 0;
```

```
            foreach (IBattery c in elems) {
                sum += c.getCurrentCapacity();
            }
            return sum;
        }
        //tác vụ tham khảo công suất max của battery
        public override int getFullCapacity() {
            int sum = 0;
            foreach (IBattery c in elems) {
                sum += c.getFullCapacity();
            }
            return sum;
        }
        //tác vụ giải mã trạng thái battery thành chuỗi
        public override String ToString()  {
            String buf = "[";
            foreach (IBattery c in elems) {
                buf += c + ", ";
            }
            return buf.Substring(0, buf.Length - 2) + "]";
        }
    }
```

Code của chương trình nhỏ để dùng và kiểm tra các pin Cell, Battery có hoạt động đúng theo đặc tả hay không như sau :

```
    static void Main(string[] args) {
        //thử tạo 1 Cell có 2u
        IBattery b1 = new Cell(2);
        //hiển thị trình trạng pin hiện hành
        Console.WriteLine("DL max của b1 = " + b1.getFullCapacity()
                + ", DL hiện hành = " + b1.ToString());
        //thử nạp thêm 1u và hiển thị kết quả để kiểm tra
        b1.charge();
        Console.WriteLine("Dung lượng của b1 sau khi nạp thêm 1u = "
                + b1.ToString());
        //thử xả 4u và hiển thị kết quả để kiểm tra
        b1.discharge(); b1.discharge(); b1.discharge(); b1.discharge();
        Console.WriteLine("Dung lượng của b1 sau khi xả 4u = " + b1.ToString());
        //thử tạo pin tích hợp
        b1 = new Cell(10);
        IBattery b2 = new Battery(3, 5, false);         //dùng chiến lược nạp/xả MinMax
        IBattery b3 = new Battery(2, 10, false);        //dùng chiến lược nạp/xả MinMax
        b3.gm_add(b1); b3.gm_add(b2);
        Console.WriteLine("Trạng thái của b3 = " + b3.ToString());
```

```
        //thử xả 2u và hiển thị kết quả để kiểm tra
        b3.discharge(); b3.discharge();
        Console.WriteLine("b3 sau khi xả 2u = " + b3.ToString());
        //thử nạp 2u và hiển thị kết quả để kiểm tra
        b3.charge(); b3.charge();
        Console.WriteLine("b3 sau khi nạp 2u = " + b3.ToString());
        Console.Read();
    }
```