

Ví dụ 3.22. Để quản lý các học phần của một trường đại học, ta dùng một danh sách liên kết đơn, mỗi nút của danh sách biểu diễn một đối tượng học phần và có các thông tin: mã học phần, tên học phần, số tín chỉ. Kiểu lớp dùng để mô tả đối tượng học phần được cho như sau:

```
class Hocphan
{
    private:
        char *mahp    // mã học phần
        char *tenhp;  // tên học phần
        int sotc;      // số tín chỉ
    public:
        Hocphan *next;
        // Các phương thức để khởi tạo giá trị
        // cho mahp, tenhp, sotc
        // Các phương thức để trả về giá trị
        // của mahp, tenhp, sotc
};
```

Cho khai báo kiểu lớp `List` sau đây, trong đó mỗi đối tượng của lớp `List` là một danh sách liên kết chứa các phần tử kiểu `Hocphan`:

```
class List
{
    private:
        Hocphan * first;
    public:
        //các phương thức thể hiện yêu cầu
};
```

Chương trình sau đây sẽ thực hiện các yêu cầu:

1. Định nghĩa các phương thức để khởi tạo giá trị cho `mahp`, `tenhp`, `sotc`; Các phương thức lấy giá trị của `mahp`, `tenhp`, `sotc` của lớp `Hocphan`.

2. Định nghĩa các phương thức của lớp `List`:

- Nhập dữ liệu cho đối tượng của lớp `List`, quá trình nhập dữ liệu kết thúc khi mã học phần nhập vào là rỗng.
- Sắp xếp các học phần theo thứ tự tăng của mã học phần.
- Tìm kiếm theo mã học phần.
- Xóa học phần theo mã học phần.
- Hiển thị thông tin về các học phần.

3. Hàm `main()` minh họa.

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
using namespace std;
class Hocphan
{
    private:
        char *mahp, *tenhp;
        int sotc;
    public:
        Hocphan *next;
        Hocphan(char *mhp, char *thp, int tc)
        {
            mahp = new char[strlen(mhp)+1];
            tenhp = new char[strlen(thp)+1];
            strcpy(mahp, mhp);
            strcpy(tenhp, thp);
            sotc = tc;
        }
}
```

```

char *GetMaHP()
{
    return mahp;
}

char *GetTenHP()
{
    return tenhp;
}

int GetSoTC()
{
    return sotc;
}

void Get(char *mhp, char *thp, int tc)
{
    mahp = new char[strlen(mhp)+1];
    tenhp = new char[strlen(thp)+1];
    strcpy(mahp, mhp);
    strcpy(tenhp, thp);
    sotc = tc;
}

};

```

```

class List
{
    private:
        Hocphan *first;
    public:
        List()

```

```

        {
            first = NULL;
        }

        void Nhap();
        void HienThi();
        void SapXep();
        Hocphan *TimKiem(char *hp);
        void Xoa(char *hp);
};

void List::Nhap()
{
    char mhp[10],thp[30];
    int tc;
    do
    {
        cout<<"\n Nhap ma hoc phan: ";
        fflush(stdin); gets(mhp);
        if(strcmpi(mhp,"") != 0)
        {
            cout<<"\n  Nhap ten hoc phan: ";
            fflush(stdin); gets(thp);

            cout<<"\n  Nhap so tin chi: "; cin>>tc;
            Hocphan *tam = new Hocphan(mhp,thp,tc);
            tam->next = first;
            first = tam;
        }
    }while(strcmpi(mhp,"") != 0) ;
}

```

```

void List::HienThi()
{
    Hocphan * p = first;
    while(p != NULL)
    {
        cout<<"\n\n-----";
        cout<<"\n    Ma hoc phan: "<<p->GetMaHP();
        cout<<"\n    Ten hoc phan: "<<p->GetTenHP();
        cout<<"\n    So tin chi: "<<p->GetSoTC();
        cout<<"\n\n-----";
        p = p->next;
    }
}

void List::SapXep()
{
    Hocphan *p,*q;
    char temmahp[30],temthp[30];
    int temtc;
    if(first != NULL)
        for(p = first; p->next != NULL; p= p->next)
            for(q = p->next; q != NULL; q = q->next)
                if(strcmpi(p->GetMaHP(),q->GetMaHP()) > 0)
                {
                    strcpy(temmahp,p->GetMaHP());
                    strcpy(temthp,p->GetTenHP());
                    temtc = p->GetSoTC();
                    p->Get(q->GetMaHP(),q->GetTenHP(),
                        q->GetSoTC());
                }
            }
}

```

```

        q->Get(temmahp, temthp, temtc);
    }

}

Hocphan * List::TimKiem(char *mhp)
{
    Hocphan *p = first;
    while (p!=NULL &&
           strcmpi(p->GetMaHP(), mhp) !=0)
        p = p->next;
    return p;
}

void List::Xoa(char *mhp)
{
    while (first!=NULL &&
           strcmpi(first->GetMaHP(), mhp) == 0)
    {
        Hocphan *tam = first;
        first = first->next;
        delete tam;
    }
    if(first!=NULL)
    {
        Hocphan *p = first;
        while(p->next!=NULL)
            if(strcmpi(p->next->GetMaHP(), mhp) == 0)
            {
                Hocphan *tam = p->next;
                p->next = p->next->next;
            }
    }
}

```

```

        delete tam;
    }
    else
        p = p->next;
    }
}

int main()
{
    List ds;
    ds.Nhap();
    cout<<"\n Hien thi danh sach hoc phan: ";
    ds.HienThi();
    ds.SapXep();
    cout<<"\n Hien thi sau khi sap xep: ";
    ds.HienThi();
    char mhp[10],thp[30];
    cout<<"\n Nhap vao ma hp muon xoa: ";
    fflush(stdin); gets(mhp);
    ds.Xoa(mhp);
    cout<<"\n Hien thi sau khi xoa: ";
    ds.HienThi();
    cout<<"\n Nhap vao ma hp muon tim kiem ";
    fflush(stdin); gets(mhp);
    Hocphan *p = ds.TimKiem(mhp);
    char c;
    if(p!= NULL)
    {
        cout<<"\n Da tim thay hp cua ban! ";
    }
}

```

```
        cout<<"\n Ma hp : "<<p->GetMaHP();
        cout<<"\n Ten hp: "<<p->GetTenHP();
        cout<<"\n So tc: "<<p->GetSoTC();
    }
    else
        cout<<"\n Khong tim thay ma hoc phan nay";
    return 0;
}
```