

# Bài tập về nhà: Đa năng hóa toán tử

- Môn: Lập trình hướng đối tượng
- Họ và tên: Lương Văn Võ
- Lớp: 23T\_DT1
- MSSV: 102230224
- Nhóm: 23.Nh10

Bài 1: Cài đặt lớp tập hợp các số nguyên **SET** gồm  $n$  là số phần tử có trong tập hợp, *data* là mảng động chứa các phần tử của tập hợp, *size* là kích thước tối đa của tập hợp.

Định nghĩa các hàm: default constructor (với 01 tham số khởi tạo kích thước của tập hợp), constructor (khởi tạo tập hợp từ 01 mảng các số nguyên), constructor (khởi tạo tập hợp từ 01 tập hợp khác), destructor và các toán tử: + (giữa 01 tập hợp với số nguyên để thêm phần tử vào tập hợp); - (giữa 01 tập hợp với số nguyên để xóa phần tử khỏi tập hợp); +, -, \* để tìm hợp, hiệu, giao của 02 tập hợp; () để kiểm tra 01 phần tử có thuộc tập hợp; toán tử [ ] để truy xuất 01 phần tử của tập hợp; toán tử gán =; toán tử << để in nội dung tập hợp ra màn hình.

1. Các thành viên của lớp:

```

1  const int max_size = 100; // Kích thước tối đa của SET
2
3  class set {
4      private:
5          int n;
6          int *data;
7          int size;
8      public:
9          set(int n = 10);
10         ~set();
11         set(const set &);
12         set(int soNguyen[], int size);
13         set operator + (int x);
14         set operator - (int x);
15         set operator + (set &other);
16         set operator - (set &other);
17         set operator * (set &other);
18         bool operator () (int x);
19         int operator [] (unsigned int x);
20         set &operator = (const set &s);
21         friend ostream &operator << (ostream &out, const set &s);
22     };
23

```

- n: số phần tử có trong tập hợp.
- data: mảng động chứa các phần tử của tập hợp.
- size là kích thước tối đa của tập hợp.
- set(int n = 10): hàm dựng với 1 tham số khởi tạo kích thước của tập hợp
- ~set(): hàm hủy
- set(const set &): hàm khởi tạo tập hợp từ 1 tập hợp khác
- set(int soNguyen[], int size): hàm khởi tạo tập hợp từ 1 mảng các số nguyên
- set operator + (int x): hàm đa năng hóa toán tử + để thêm phần tử vào tập hợp
- set operator - (int x): hàm đa năng hóa toán tử - để xóa phần tử khỏi tập hợp.
- set operator + (set &other): hàm đa năng hóa toán tử + để tìm hợp của hai tập hợp

- set operator – (set &other): hàm đa năng hóa toán tử - để tìm hiệu của hai tập hợp
- set operator \* (set &other): hàm đa năng hóa toán tử \* để tìm giao của hai tập hợp
- bool operator () (int x): hàm đa năng hóa toán tử () kiểm tra 1 phần tử có thuộc tập hợp
- int operator [] (unsigned int x): hàm đa năng hóa toán tử [] để truy xuất 1 phần tử của tập hợp
- set &operator = (const set &s): hàm đa năng hóa toán tử gán
- friend ostream &operator << (ostream &out, const set &s): hàm đa năng hóa toán tử << để in nội dung tập hợp ra màn hình.

## 2. Kết quả chạy thử nghiệm chương trình

- Hàm main:

```

1  #include "SET.h"
2
3  int main()
4  {
5      cout << endl;
6      // Tạo mảng s1 và thêm các phân tử
7      int a[] = {2, 3, 5, 1, 8, 10};
8      set s1(a, 6);
9
10     set ss(6);
11     ss = s1;
12     cout << "SET sau khi duoc gan cho SET khac: ";
13     cout << ss;
14
15     s1 = s1 + 14;
16     s1 = s1 + 15;
17     s1 = s1 + 16;
18     cout << "SET sau khi them phan tu vao: ";
19     cout << s1;
20
21     // Xóa các phân tử khỏi s1
22     s1 = s1 - 10;
23     s1 = s1 - 14;
24     s1 = s1 - 15;
25     cout << "SET sau khi xoa di phan tu: ";
26     cout << s1;
27
28     cout << endl;
29
30     cout << "Phep hop cua 2 tap s2 va s3: \n";
31     int b[] = {1, 2, 3};
32     int c[] = {3, 4, 5};
33     set s2(b, 3), s3(c, 3);
34     cout << s2 << s3;
35     cout << "s2 + s3: " << s2 + s3 << endl;
36
37     cout << "Phep hieu cua 2 tap s4 va s5: \n";
38     int d[] = {1, 2, 3, 4, 5};
39     int e[] = {3, 4, 6};
40     set s4(d, 5), s5(e, 3);
41     cout << s4 << s5;
42     cout << "s4 - s5: " << s4 - s5 << endl;
43
44     cout << "Phep giao cua 2 tap s6 va s7: \n";
45     int f[] = {1, 2, 3, 4};
46     int g[] = {3, 4, 5, 6};
47     set s6(f, 4), s7(g, 4);
48     cout << s6 << s7;
49     cout << "s6 * s7: " << s6 * s7 << endl;
50
51
52     // Sử dụng toán tử []
53     cout << "Phan tu nam o vi tri thu 2 trong set s1 la: " << s1[1] << endl;
54     cout << "Phan tu nam o vi tri thu 4 trong set s1 la: " << s1[3] << endl;
55     cout << endl;
56
57     // Sử dụng toán tử ()
58     bool check = s1(3);
59     bool check2 = s1(10);
60
61     return 0;
62 }
63
64

```

- Kết quả chạy thử nghiệm chương trình:

```
SET sau khi duoc gan cho SET khac: 2 3 5 1 8 10
SET sau khi them phan tu vao: 2 3 5 1 8 10 14 15 16
SET sau khi xoa di phan tu: 2 3 5 1 8 16
```

Phép hợp của 2 tập s2 và s3:

1 2 3

3 4 5

s2 + s3: 1 2 3 4 5

Phép hiệu của 2 tập s4 và s5:

1 2 3 4 5

3 4 6

s4 - s5: 1 2 5

Phép giao của 2 tập s6 và s7:

1 2 3 4

3 4 5 6

s6 \* s7: 3 4

Phan tu nam o vi tri thu 2 trong set s1 la: 3

Phan tu nam o vi tri thu 4 trong set s1 la: 1

3 nam o vi tri thu 2 trong mang.

10 khong nam trong mang.

Bài 2: Cài đặt lớp **LinkList** là 01 danh sách liên kết (DSLK) gồm con trỏ đầu và cuối danh sách liên kết các số nguyên. Định nghĩa hàm dựng tạo DSLK rỗng; hàm hủy; hàm dựng sao chép; hàm thêm 01 phần tử x vào đầu/cuối/sau 01 phần tử; hàm xóa 01 phần tử đầu/cuối/sau 01 phần tử; hàm thêm/xóa 01 phần tử có giá trị x trong DSLK; hàm toán tử << để hiển thị DSLK; hàm sắp xếp các phần tử trong DSLK theo thứ tự tăng/giảm dần.

### 1. Các thành viên của lớp

```

1  class node {
2      public:
3          int value;
4          node *next;
5  };
6
7  class LinkList {
8      private:
9          node *begin;
10         node *end;
11     public:
12         LinkList();
13         ~LinkList();
14         LinkList(const LinkList &);
15
16         void addEnd(int x); // Thêm phần tử x vào cuối danh sách
17         void addBegin(int x); // Thêm phần tử x vào đầu danh sách
18         void addLaterOneElement(int element, int x); // Thêm phần tử x vào sau 1 phần tử
19
20         void deleteEnd(); // Xóa 1 phần tử ở cuối danh sách
21         void deleteBegin(); // Xóa phần tử ở đầu danh sách
22         void deleteLaterOneElement(int element); // Xóa phần tử sau 1 phần tử
23
24         void addElementX(int element); // Thêm 1 phần tử có giá trị x trong DSLK
25         void deleteElementX(int element); // Xóa 1 phần tử có giá trị x trong DSLK
26
27         void sortAscending(); // sắp xếp danh sách tăng dần
28         void sortDescending(); // sắp xếp danh sách giảm dần
29
30         friend ostream &operator << (ostream &out, const LinkList &);
31     };
32

```

- class node: dùng để xây dựng các node trong cấu trúc danh sách liên kết
  - int value: lưu giá trị của mỗi node.
  - node \*next: con trỏ đến node tiếp theo trong danh sách liên kết.
- class LinkList:
  - node \*begin: con trỏ đến node đầu tiên của danh sách liên kết.
  - node \*end: con trỏ đến node cuối cùng của danh sách liên kết.
  - LinkList(): Hàm dựng tạo DSLK rỗng
  - ~LinkList(): Hàm hủy
  - LinkList(const LinkList &): Hàm dựng sao chép
  - void addEnd(int x): Hàm thêm 1 phần tử x vào cuối DSLK
  - void addBegin(int x): Hàm thêm 1 phần tử vào đầu DSLK
  - void addLaterOneElement(int element, int x): Thêm phần tử x vào sau 1 phần tử trong DSLK
  - void deleteEnd(): Xóa 1 phần tử ở cuối danh sách

- void deleteBegin(): xóa 1 phần tử ở đầu danh sách
  - void deleteLaterOneElement(int element): Xóa 1 phần tử sau 1 phần tử có trong DSLK
  - void addElementX(int element): Thêm 1 phần tử có giá trị x trong DSLK
  - void deleteElementX(int element): Xóa 1 phần tử có giá trị x trong DSLK
  - void sortAscending(): sắp xếp danh sách liên kết tăng dần
  - void sortDescending(): sắp xếp danh sách liên kết giảm dần
  - friend ostream &operator << (ostream &out, const LinkList &): Hàm toán tử << để hiển thị danh sách liên kết.
2. Kết quả chạy thử nghiệm chương trình
- Hàm main:

```

1  #include "LinkList.h"
2  int main()
3  {
4      LinkList p1;
5      // Thêm phần tử vào cuối danh sách
6      p1.addEnd(1);
7      p1.addEnd(2);
8      p1.addEnd(3);
9      p1.addEnd(17);
10     p1.addEnd(12);
11     p1.addEnd(21);
12
13     cout << "Danh sach sau khi them cac phan tu vao cuoi danh sach:: ";
14     cout << p1;
15
16     // Thêm phần tử vào đầu danh sách
17     p1.addBegin(10);
18     p1.addBegin(14);
19     p1.addBegin(32);
20     p1.addBegin(29);
21     cout << "Danh sach sau khi them phan tu vao dau danh sach: ";
22     cout << p1;
23
24     // Phân tử sau vào sau 1 phân tử
25     p1.addLaterOneElement(1, 9);
26     p1.addLaterOneElement(3, 7);
27     cout << "Danh sach sau khi them phan tu 9 va 7 lan luot vao sau gia tri 1 va 3: ";
28     cout << p1;
29
30     // Xóa 1 phần tử ở cuối danh sách
31     p1.deleteEnd();
32     cout << "Danh sach sau khi xoa phan tu cuoi: ";
33     cout << p1;
34     // Xóa 1 phần tử ở đầu danh sách
35     p1.deleteBegin();
36     cout << "Danh sach sau khi xoa phan tu dau: ";
37     cout << p1;
38     // Xóa 1 phần tử sau 1 phân tử
39     p1.deleteLaterOneElement(9);
40     cout << "Danh sach sau khi xoa phan tu sau phan tu co gia tri 9: ";
41     cout << p1;
42
43     // Sắp xếp danh sách theo chiều tăng dần
44     p1.sortAscending();
45     cout << "Danh sach sau khi sap xep tang dan: ";
46     cout << p1;
47     // Sắp xếp danh sách theo chiều giảm dần
48     p1.sortDescending();
49     cout << "Danh sach sau khi sap xep giam dan: ";
50     cout << p1;
51
52     // Thêm 1 phần tử có giá trị x trong DSLK
53     p1.addElementX(14);
54     cout << "Danh sach sau khi them phan tu co gia tri x trong DSLK: ";
55     cout << p1;
56
57     // Thêm 1 phần tử có giá trị x trong DSLK
58     p1.deleteElementX(9);
59     cout << "Danh sach sau khi xoa phan tu co gia tri x trong DSLK: ";
60     cout << p1;
61     return 0;
62 }

```



- Kết quả chạy thử nghiệm chương trình:

```
Danh sach sau khi them cac phan tu vao cuoi danh sach: 1 2 3 17 12 21
Danh sach sau khi them phan tu vao dau danh sach: 29 32 14 10 1 2 3 17 12 21
Danh sach sau khi them phan tu 9 va 7 lan luot vao sau gia tri 1 va 3: 29 32 14 10 1 9 2 3 7 17 12 21
Danh sach sau khi xoa phan tu cuoi: 29 32 14 10 1 9 2 3 7 17 12
Danh sach sau khi xoa phan tu dau: 32 14 10 1 9 2 3 7 17 12
Danh sach sau khi xoa phan tu sau phan tu co gia tri 9: 32 14 10 1 9 3 7 17 12
Danh sach sau khi sap xep tang dan: 1 3 7 9 10 12 14 17 32
Danh sach sau khi sap xep giam dan: 32 17 14 12 10 9 7 3 1
Danh sach sau khi them phan tu co gia tri 14 trong DLSK: 32 17 14 12 10 9 7 3 1 14
Danh sach sau khi xoa phan tu co gia tri 9 trong DSLK: 32 17 14 12 10 7 3 1 14
```